EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN-EP/79-52
DD/79/3
30 May 1979

AN ON-LINE DATA ACQUISITION SYSTEM FOR PARTICLE PHYSICS
EXPERIMENTS USING PDP-11 COMPUTERS

D. Botterill and B. Charles

Rutherford Laboratory, Chilton, Didcot, England

V. White

CERN, Geneva, Switzerland

ABSTRACT

An on-line computer system has been developed for the large European Muon Collaboration experiment which recently commenced data-taking at CERN. The system consists of four inter-connected PDP-11/70 computers which handle data acquisition, control, and monitoring for the experiment. Much of the software is written in a general manner and is now in use on several other experiments at CERN.

(Submitted to Nuclear Instruments and Methods)

1. INTRODUCTION

The European Muon Collaboration consists of a group of approximately 70 physicists from 12 institutes located in four countries in Europe who have jointly constructed a large experimental set-up in a muon beam at the CERN Super Proton Synchrotron (SPS) accelerator¹⁾. The apparatus for the first phase of the experiment consists of approximately 12 separate large detectors which went through final commissioning early in 1978, and data-taking commenced in August 1978. An on-line computing system has been developed, based on four PDP-11/70 computers, to handle the on-line computational needs of the experiment²⁾.

One of the main requirements was a large degree of flexibility in the software, since the apparatus and hence data acquisition and monitoring requirements were expected to evolve with time. The software was therefore written in a rather general manner and can be configured to meet the needs of most particle physics experiments.

The system is also multi-user. The different detectors were developed and tested in parallel and there had to be a reasonable amount of independence and protection for the different groups working on them. This led to a requirement for a "multi-user" data acquisition system allowing simultaneous operations by different groups. The data acquisition also had to be multi-trigger. The apparatus was planned to allow several separate physics experiments to be performed concurrently, and hence the data acquisition had to be able to handle several separate "runs" from different triggers simultaneously.

The on-line system had to be easy to use as it is controlled by the physicists themselves and not by experienced operators. The data rate handled is approximately 200,000 bytes, which arrives in the one-second beam burst with a repetition rate of about once every nine seconds.

As the on-line system was distributed over four computers, machine independence in the software was aimed for so as to allow easy reconfiguring in case of machine failure or changed requirements. In fact, the system can be run on a single machine with suitably reduced monitoring capability.

2. SYSTEM OVERVIEW

The layout of the computer system is shown in Fig. 1. The layout of computers and their interconnection is similar to that used at the CERN Omega facility³⁾. Each of the PDP-11/70s runs the DEC-provided RSX11M operating system.

The computer labelled "DAC" is the data acquisition computer which reads out data from the apparatus (via GEC-Elliott interfaced CAMAC) into a 160 Kbyte buffer, and subsequently writes the events out to one or more of the four magnetic tape drives. The U0, U1, and U2 computers may be used for monitoring the data by asking the DAC for sample events to be passed across the links⁴⁾. The U0 also shares access to the CAMAC with the DAC and can do direct checks on the apparatus and read-out electronics between bursts⁵⁾.

The U1 or U2 may also be used to commission new equipment in stand-alone mode.

The software residing in the machines can be divided roughly into two parts: that for the data acquisition, and that for monitoring the state of the data and the experiment in general. The functions of data acquisition are handled by a relatively small number of tasks which were programmed by the authors. The monitoring functions, however, are divided up amongst some 20 tasks, which were written largely in FORTRAN by other members of the collaboration using a number of software aids. The data acquisition software and these aids are written predominately in MACRO-11 and PL11, a CERN-produced high-level assembler language⁶⁾.

3. OPERATOR COMMUNICATION PHILOSOPHY

3.1 Operator-initiated communication

Operator communication is by a combination of menus (i.e. a list of options to choose between) and dialogue. Although menus can be slower than other methods of handling operator commands, they have the distinct advantage of being easy for the unfamiliar to use.

A top-level menu is provided by a program known as the Scheduler, which may be displayed on any terminal and gives a list of tasks available in the machine

together with a brief description of each. Selection of a task causes that task to be executed (or prompted if it is already executing) so that it displays its first-level menu on the terminal. Selection of options at each level allows the operator to perform the action he requires. Where parameters must be entered, a question and answer dialogue may be initiated. Each parameter of a program has a default value which the operator need only change if he wants to. A general facility is provided to make this feature easy to program by use of a named COMMON block which is automatically saved on disk between task executions.

Two special buttons TOP and BACK (characters 1 and 2 on the VDUs) permit the operator to move between menu levels. A control character (control/2) allows the operator to exit from a dialogue phase at any time. It is also possible to initiate tasks from the top level menu so that they perform the same operation as last time without fresh instructions from the operator.

The terminal used for the menus may be any of a range of devices including a touch screen and six types of VDU and graphic terminal. Each of these devices is assigned a type number, and the menu package routines configure the output and input accordingly. Passing through menu levels is speeded up for experienced users by virtue of the fact that the desired option can be input before the output of the menu is complete. There is an optional provision for an external interrupt button to display the top-level menu on the terminal.

3.2 Alarm and status output

Tasks may initiate alarm or status messages at any time independently of any operator prompting. No restriction of one task per terminal has been imposed as there is a necessity to have more monitoring tasks active than there are terminals available. Thus tasks cannot be guaranteed access to any particular terminal at any moment, and hence unsolicited output must be routed to one (or more) dedicated places.

The output of short error or status messages is centralized through one task. The advantages of such a system are that it allows greater operator control (e.g.

to suppress floods of messages), imposes a uniform interface to the operator, allows the accumulation of error statistics, and permits the creation of message logs on disk for later printing.

The method employed allows tasks to enter messages into a queue (by a subroutine call) for handling by a central task which has access to a file of messages. The method is comparable to a FORTRAN write of a single line where the values of the I/O list elements are put in the message queue together with the format statement (message) number, by the originating task. The central task uses the message number to access the format statement from the message file and encode and then output the message. This method is consistent with both short queue lengths and FORTRAN I/O flexibility.

Each message may be output on one or more "devices". There is a hard-copy terminal plus a "status" display which has several extension TV monitors. This display is divided into two sections: a fixed part for holding the most important "status" messages, and a variable part where all messages may appear successively on a wrap-around basis. Messages may also cause the sounding of an audible alarm or in serious cases cause the data-taking to stop.

Messages may be suppressed either under manual control or automatically if the message occurrence rate exceeds a certain threshold. The message handling system also encompasses the reporting of messages originating in one machine on another machine via transfers over the links between message systems. This ensures that machine independence is upheld and that all messages may be recorded on the same physical device independently of which machine the issuing task was running on.

3.3 Monitoring task results

Almost all results produced by monitoring tasks are written to disk in a standard format, and displays are produced from these files using a single task known as the Information Presenter. These results may be in the form of one- or two-dimensional histograms, graphs, pictures, or pages of text. The presenter guides the operator to the display he wants via a series of menus.

The advantages of centralizing the presentation of this information in one task lie in imposing a uniform interface to the operator and in allowing him to see easily all the information available. It also reduces the amount of programming necessary in each of the information-generating tasks. Furthermore, all information is collected together in one place thereby facilitating the archiving of the information on tape or elsewhere.

The Presenter allows up to four displays on a screen simultaneously, and a range of different terminal types plus line printer hard-copy are supported. Options are available to change the scales on displays and to automatically refresh the display with the latest data at selected intervals.

As part of the multi-machine capability, information from files on machines other than the one the terminal is on can be displayed by transfer over the computer links.

4. DATA ACQUISITION

The section of the system dealing with data acquisition consists of about six tasks which handle the function of reading data from CAMAC, writing data to tape, passing sample events to monitoring tasks, and providing control of data acquisition operation. There are facilities for data reduction and the "off-line" reading back of runs from tape.

The operations necessary to perform the read-out are defined by a set of CAMAC lists. These define the CAMAC operations which must be performed at start of burst, on occurrence of an event interrupt and at end of burst. A simple language is used to input to the system the necessary commands in an easily readable form. The language allows comments, and the addresses of the CAMAC modules may be defined symbolically.

To set up a data acquisition system for an experiment a source file is produced, according to the language rules, which is then passed through a compiler. Another task is used to set up parameters such as event buffer size, maximum event size, the interrupt numbers, mode of data acquisition, and so on. The result of

these two operations is a file which is used by the data acquisition task to create a dynamic shared region in memory containing the event buffer, the CAMAC lists, and all the defined parameter values.

At each start of burst, the data acquisition task issues a QIO (Executive I/O request) to pass the address of the shared region to a special module (the "events module") in the CAMAC driver. This module performs the read-out operations during the burst, and runs at the highest priority level so as to achieve the required response times. The start-of-burst lists are executed by the events module, which then returns control to the executive until the first event interrupt arrives. There may be up to 15 different experimental triggers in the system and these are "ORed" together to produce a single event interrupt.

The events module responds to an event interrupt by reading a CAMAC register giving the trigger(s) which have occurred. It uses this to decide which of the event read-out lists (which may be designated as specific to one or more triggers) to execute. On occurrence of the end-of-burst interrupt, the end-of-burst lists are executed and the QIO is terminated.

There may be many "users" of the data acquisition system, and each user may have a "run" in operation. Each run designates which triggers are to be activated and whether the data is to be written to tape or not. The data may arrive "on-line" (from CAMAC) or "off-line" (from tape).

The data acquisition is controlled by a run control task and a tape control task. The former may reside in any machine and is used to set up and control each user's run. The latter exists only on the DAC and is used to signal the mounting of fresh tapes or to recover tapes after a power failure or system crash. Tapes may be pre-mounted for later automatic use as other tapes become full.

A CERN tape label standard⁷⁾ is employed which is a subset of the IBM and CDC standards. There is a tape book-keeping scheme which automatically records on file and tape the details of each run. Users may also specify that certain data files are to be written to tape at the start of every run.

Several other settable options exist for use by data acquisition. For example: records on tape and data samples may be single events or blocked; event buffer "wrap-around" may be enabled (this allows the space used by events already written to tape to be freed for later events); there is a "continuous" mode for use with cosmic-ray test data or non-pulsed accelerators, and one may force all events to be sampled. Provision is made in the design for a data reduction task to be brought into the system to reduce the amount of data written to tape.

5. MONITORING

A major function of the on-line system is to monitor the state of the apparatus and the integrity of the data. This is carried out by about 20 tasks which normally reside in two or three of the computers U0, U1, and U2. Some of the tasks perform direct checks on the apparatus, but the majority of the monitoring is done using samples of the events read. There is, in general, one monitoring task of this type for each of the twelve separate detectors.

The data acquisition task accepts requests for sample events from monitoring tasks either residing in the same or on another computer. Such events are copied either between memory locations or across the links to a buffer in the monitoring task where it can have undisputed and if necessary lengthy access to the data. An algorithm is provided which ensures that the sample of events received is even over the spill. The hardware and software has been constructed so as to divide the event data into up to 32 "pieces of equipment". A routine is provided to locate the data for a piece of equipment in an event, independent of whatever other data is present. This ensures that no changes are necessary to tasks, other than those directly concerned, when a detector or the read-out electronics is changed.

Besides the event-sampling tasks, there are monitoring tasks to make direct checks on the CAMAC system, on the voltage and current levels on the electronics and the apparatus, on the integrity of the links, and on various items of electronics and some detectors which have built-in checking facilities. There is also a task which sends single events via the CERN network to a job running in the

central IBM 370/168, which performs complete event reconstruction. The results are displayed back on the PDP in the form of fitted tracks on a plan of the apparatus. A sample of events is also sent to a disk file on the IBM every hour for subsequent further analysis by a batch job.

The results produced by monitoring tasks are displayed using the information presenter as described above, and those results requiring to be saved are written to tape by requests made on the data acquisition task by the monitoring tasks.

There are generally more tasks active in each machine than there is memory space available, and competition for this and CPU resources is controlled by the Scheduler task (which also handles the top-level menu -- see above). The Scheduler does this by changing the executive software priority of tasks according to time-slices or event sample allocations. It can also be used to schedule tasks to run at preset intervals, for example to allow a task to perform a check once per hour.

Each monitoring task is controlled by the exchange of messages between it and the Scheduler. To ease implementation of this, each task has the same main routine, which calls a set of user routines to perform the required monitoring functions. Below these, a set of library routines³⁾ are available to simplify the construction of menus, to sample an event, to perform histogramming and so on.

The Scheduler is one example of the attempt made to minimize possible interference between monitoring tasks and the system. Another example is the problem of conflicting use which could arise between those tasks which access CAMAC directly. This has been resolved by the use of reservation techniques in the driver and by blocking access to the CAMAC during the burst.

6. CONCLUSIONS

An on-line data acquisition system has been developed which meets the needs of most particle physics experiments. The system has been demonstrated to be portable to other PDP computer-based experiments. Work is continuing to enhance this capability and to increase the facilities available.

Acknowledgements

The authors would like to thank the other members of the collaboration, the CERN DD Division, the Rutherford Laboratory Data Handling Group and the computing groups of the associated institutes for all their ideas and help. Individual contributions are too numerous to mention, but we would like in particular to express our thanks to R. Dobinson for his help and guidance on all matters and especially in programming and operating the read-out, to R. Clifft for the help provided in the initial planning of the system, and to DD Division for the provision of hardware (C. Parkman and group) and software (Minicomputer and CERNET groups) support.

REFERENCES

- 1) The European Muon Collaboration, Proposed experiments and equipment for a program of muon physics at the SPS, CERN/SPSC/74-78 (1974).
The European Muon Collaboration, A large magnetic spectrometer system for high-energy muon physics, Paper to be submitted to Nuclear Instrum. and Methods.
- 2) B. Charles (ed.), The EMC on-line computing manual, Rutherford Lab. report RL/76/145C.
- 3) R.D. Russell, The Omega Project: A case history in the design and implementation of an on-line data acquisition system, CERN 72-21 (1972).
C. Bizeau et al., The data acquisition system for SPS Omega, internal note OM/SPS/85 (1974), unpublished.
- 4) C. Parkman (ed.), The OMNET manual, CERN DD Division document.
- 5) D. Botterill and R. Dobinson, A shared CAMAC system, in preparation.
- 6) R.D. Russell and T.C. Streater (ed.), PL11 - A programming language for the DEC PDP-11 Computer with addendum, CERN 74-24 REV. (1978).
- 7) E.M. Rimmer and J. Ogilvie, On-line support for labelled magnetic tapes, CERN DD/US/21 (1977).
- 8) A. Silverman (ed.), The CERNLIB manual, CERN DD Division document.

* * *

Figure caption

Fig. 1 : EMC on-line computer system.

Connected terminals: 5 Digital LA36 hard-copy terminals
 7 Tektronix 4006 graphics terminals
 5 Minibee VDUs
 plus: 4 Versatec printers for computer-generated output
 and graphics hard-copy

Fig. 1