NUCLEAR INSTRUMENTS AND METHODS 140 (1977) 577–579;

SOFTWARE TO DRIVE A GEC ELLIOT CAMAC INTERFACE ON A PDP-11 COMPUTER RUNNING THE RSX-11M EXECUTIVE

CHRISTIAN LENGAUER

Hahn-Meitner-Institut für Kernforschung, Berlin GmbH, W. Germany

and

ALAN SILVERMAN

CERN Geneva, Switzerland

Received 16 August 1976

A software driver module has been written and implemented to allow access to CAMAC via the GEC Elliot System crate SI-11 interface from any application programme running on a PDP-11 using the RSX-11M operating system. On the one hand, the driver conforms to the normal RSX-11M executive rules while on the other, a user can access the CAMAC via a set of simple FORTRAN subroutines.

1. Introduction

1.1. Environment

The project for which this work was initially begun is the Omega Spectrometer at CERN. This is a large superconducting magnet used in the analysis of particles produced by the CERN accelerators (until last year by the CERN PS; from October 1976 by the SPS). The actual particle detectors are such things as spark chambers read out by plumbicons, multiwire proportional chambers and so on. All data is gathered into the computer via CAMAC and stored on magnetic tape for off-line analysis. There is also a certain amount of on-line checking of the data.

It is now planned to use the same driver in the European Muon Collaboration, a large experiment being set up to run in 1978 in the North Area of the SPS.

1.2. Hardware

In order to cope with the higher data rates and more demanding requirements of the SPS, Omega has been re-equipped with new computers and new CAMAC.

The computers are three Digital Equipment PDP-11/40's with the following peripherals – two moving head discs, console, storage scope terminal/display, line printer, high speed data link and a GEC Elliot System crate interface. One PDP-11 also has two fast magnetic tape transports. Two computers have 48K words and are used one for reading the user's own CAMAC data and the other to run checking programmes and to allow the engineers to debug their equipment. The third PDP has 64K words and reads all the "common" CAMAC data (e.g. plumbicon, MWPC, central detectors, etc.), adds on the user's data and writes the whole event data to tape. A fourth PDP-11 is being added in order to allow multi-user operation of Omega.

The operating system for these PDPs is RSX-11M, a real-time system provided by Digital Equipment[1]). It allows multi-users, multi-programming, interrupt driven tasks and so on. Programming languages available are FORTRAN and MACRO-11 assembler, plus the CERN-developed PL-11 language[2]).

The CAMAC interface is the GEC Elliot System Crate. This allows the computer to control up to four branches, each of which may have up to seven crates. LAMs 1-20 are handled by a special module in the System Crate and are priority-vectored into the PDP-11. A DMA unit is now being tested.

2. The driver

The driver is accessed via a normal RSX I/O request, called a QIO. The request directive is of the form (in MACRO-11):

```
QIO$  FNC,LUN,FLAG,PRIORITY,ISB,AST,
      <BUF,CNT,MODE,CUR,CSR>
```

where the parameters are:

FNC = I/O function, one of the following

IO.ATT	attach CAMAC to this task;
IO.DET	detach CAMAC;
IO.BKL	book a CAMAC LAM for this task; BUF contains LAM number;
IO.RLL	release a LAM;

IO.INC initialise a crate (*Z*, remove *I*, enable demand); BUF contains B, C of crate;
IO.RLC release a crate (allow another task to initialise it);
IO.RLB read a CAMAC module (single word); BUF contains data;
IO.WLB write a CAMAC module (single word); BUF contains data;
IO.OPR perform a dataless operation on a CAMAC module;
IO.DMA perform a CAMAC DMA transfer;
IO.LST perform a list of CAMAC commands; BUF contains address of the list;
LUN = logical unit number assigned to the CAMAC,
FLAG = event flag to set on I/O completion,
PRIORITY = not used,
ISB = I/O completion status (e.g. Q and X from CAMAC),
AST = not used in general,
BUF = data buffer address,
CNT = size in bytes of data buffer,
MODE = 0 for 16-bit CAMAC I/O,
1 for 24-bit CAMAC I/O,
CUR = CAMAC unibus address (= NA),
CSR = control and status register contents (= BCF).

The execution of a QIO takes 1.5–2 ms due to system overheads and queuing – hence the reason for the IO.LST command to perform several (indeed, many) CAMAC commands with one QIO.

The use of QIO allows several tasks 'simultaneous' access to the CAMAC and RSX ensures that users do not interfere with each other's data.

3. Privileged I/O

Since there may be cases where a task is time-critical and cannot afford 2 ms overhead, we have provided five privileged commands to allow fast I/O. These are (in FORTRAN-type calls):

CAMOP (CUR,CSR,STATUS)
dataless command,
CAMRS (CUR,CSR,DATA,STATUS)
16-bit read,
CAMRD (CUR,CSR,DATA,STATUS)
24-bit read,
CAMWS (CUR,CSR,DATA,STATUS)
16-bit write,
CAMWD (CUR,CSR,DATA,STATUS)
24-bit write.

These calls can only be used by 'privileged' tasks in RSX; tasks which can directly access hardware devices and hence are totally unprotected by (and to) the executive. However, they are callable from FORTRAN and PL-11. The execution times on a PDP-11/40 range from 33 to 56 μs.

4. LAMs

In the first version of the driver, the LAM number was forced to correspond to the number of the station which issued the LAM. There were two states of LAMs:

1) Booked LAMs were those reserved by tasks. Tasks were able to book stations and then get information (branch and crate numbers) when a LAM occurred in a station booked by them. The occurrence of a booked LAM caused an attempt to be made to resume the owner task, so that it could (but need not) wait for the LAM in a suspended state. If the owner task was not active, it lost its reservation on the occurrence of the LAM.
2) Unbooked LAMs were understood as not important and were cleared immediately on occurrence – i.e. transparent to current user tasks.

An interface task informed the owner of a booked LAM on its occurrence, or printed an error message at the operator console on the occurrence of an unbooked LAM.

(In the current version of the driver, the LAM number equals station number requirement has been removed, unbooked LAMs are masked out, LAM grading is permitted and LAMs are freed automatically when a task aborts or exits.)

5. Service subroutines

A full set of FORTRAN-callable subroutines (which themselves are written in PL-11) have been developped which allow an easy use of the driver. even to users who are not familiar with the features of the operating system. They accept BCNAF-data parameter strings and issue the command or they will encode BCNAF strings into the CUR, CSR values needed for the privileged commands. We give a few examples in FORTRAN (all the routines are also callable from PL-11).

CALL CAMCOM (B,C,N,A,F,DATA,SIZE, MODE,Q,X):
– perform command BCNAF, data buffer is DATA,

buffer size SIZE, 16/24 bit given by MODE, return Q and X);

CALL CAMLAM (LAM,MODE):

– reserve LAM number LAM if MODE = 1, free LAM number LAM if MODE = 0;

CALL CAMIN (B,C):

– initialise crate C in branch B;

CALL CDECOD (B,C,N,A,F,CUR,CSR):

– encode command BCNAF into CUR and CSR for later use (e.g. in a call to CAMRS, a privileged command).

We would like to thank Messrs. Bourgeois, Bizeau and Lauper for their help in hardware and software questions.

References

[1]) RSX-11M software product description, Digital Equipment Corporation, DEC-11-XPDBG-C-D (Sept. 1975).

[2]) R. D. Russell, PL-11: a programming language for the DEC PDP-11 computer, (ed. T. C. Streater) CERN Yellow Report, 74-24 (Dec. 1974).