MPS/CCI Note 74-15
16 April, 1974

SOFTWARE TO DRIVE CAMAC

H. Davies, G. Jennings, A. Silverman

Introduction

We describe below our current ideas on the standard software package which CCI will provide to drive the separate CAMAC subsystems for the new PS control system. We define a subsystem as the CAMAC equipment attached via a CAMAC highway and driver, either branch and/or serial, to the unibus of a front-end PDP 11 computer (expected to be an 11/10), which in turn is connected to BIDUL and hence the central 11/45's.

Assumptions

i) Initially we shall use a branch highway with a driver of the CA 11 type or similar. At a later stage a serial highway may be added (see the memo of 26th March, 1974).

ii) There need only be one read sequence after, and one write sequence before, each machine cycle. Any control or acquisition during a burst would complicate the software and slow down the data transfer. Any data which accumulate during a burst could be stored in a local buffer within CAMAC and read at the end of the cycle.

iii) There will be no variable length data block transfers; all blocks will have a fixed size, or at least a fixed maximum size which will be read.

iv) All data are 16 bit words.

v) The occurrence of a LAM does not alter the order of either the read or the write sequence.

Description of method

All the data transfers are table-driven: in other words there will be in the 11/10 memory, tables of read and write data and at fixed

/je

times in the machine cycle the 11/10 will use these tables to select the CNAF for the CAMAC operations requested. Also some surveillance tasks and communication requests from the central computers will be driven by tables. The various table formats are shown in figures 1 - 3 and described below.

1. Acquisition

In order to simplify implementation, speed up the transfer and have a complete picture of the current state of the process, all acquisition parameters will be read each pulse. The driving table (fig. 1) is in a block structure where each block uses the same crate number and read function - that is the same CF (usually F(0)). Thus a block consists of the CF word, in the form required by the CAMAC command and status register, and a list of NA words, in the form of CAMAC address registers, and data words which are refreshed on each read sequence. A blank word signifies end of block and the following word in a new CF word.

N.B. If a read function other than F(0) is used, the transfer will be slowed down.

After each read operation the Q response will be checked and if an error occurs an error flag will be set along with the bit in an error bit map corresponding to the position of that CNA in the table. At the end of the acquisition phase if the error flag has been set, the error bit map will be interrogated to find the address of the error and a message sent to the operator. In addition, a second bit map could be compared and used to initiate special action such as immediate retry or process shutdown.

2. Surveillance

Some percentage of the acquisition parameters should be regularly checked for drift, either every cycle or every N cycles, or by checking some fraction on one cycle and a different fraction on the next. The table for this operation is shown in figure 2.

For each parameter there are 4 entries - a code number to identify it (on display to the operator this would be translated into a name), a pointer to its position in the acquisition table and the maximum and minimum permissible values. The checking would be done between bursts and the operator informed of any fault condition.

3. Request read

When a program in a central 11/45 requires the value of a parameter or a set of parameters, it will send a request to the 11/10 which will be translated - in which computer it is not yet finally decided - into a pointer or set of pointers in a table (fig. 3). Each pointer refers to the address of the required parameter data value in the acquisition table and at the end of the acquisition phase the pointers are replaced by the data values and the table is sent back via BIDUL to the requesting 11/45.

The implication of overwriting the table of pointers with the required data is that a request is only valid for one pulse and a new set of requests must be made for the next pulse, although some provision could be made for repeated request of some parameters.

4. Control

We neglect here the initial start-up of a process where all the parameters must be set simultaneously. Under the following model this would make extreme demands in core space for relatively infrequent use and so should be handled separately.

The table format for control is similar to that for acquisition (see fig. 1) - a series of blocks of one CF word followed by several NA and data words and a final blank word to signify end of block and a new CF word following.

N.B. As a general rule F(16) will be used to write control parameters.

However, there may be two types of control table - fixed and variable. The fixed control tables will be used for switching between fixed parameter sets from pulse to pulse, in particular for SPS supercycles. There will be as many of these tables as there are forms of PS, or Linac, pulse per supercycle.

The variable tables will be used to set parameters to new values before the next machine pulse as requested by a program in the control computer. This variable table will be effectively zeroed after each write sequence and so must be set up afresh for the next pulse with the requested control parameters for that pulse.

At a later stage, we could envisage some form of closed loop control within the 11/10 where a user-written program may reset control values as a result of the analysis of some acquisition values.

11/10 core space and transfer rates

Acquisition table	requires approximately	2 words per parameter
Surveillance table	requires	4 words per parameter
Request read table	requires	1 word per parameter
Control table	requires approximately	2 words per parameter

Thus for a typical subsystem having about 1000 acquisition parameters, 500 control parameters and some SPS supercycle switching, we estimate that the tables will require 5 - 6 K words of core in the 11/10.

In calculating the transfer rate we refer again to our initial assumptions and we estimate that, with bookkeeping overheads such as checking Q response, updating the error bit map pointer and so on, a read operation should take around 40 μsec, a write operation slightly less. The implementation of a serial highway will slow down these speeds.

Parameter names and CAMAC addresses

At the present time we have not yet decided on the means of correlation between a parameter name and its CAMAC address and the best way of using the data base. However we note that the software in the 11/10 should be as simple as possible and that, at least in the production mode, the tables should be made up of data words sent via BIDUL from the 11/45's, both for the fixed tables which are set up at the beginning of a run (the acquisition, surveillance and fixed control tables) and for the variable tables which vary from pulse to pulse (the request read and variable control tables).

Also for the moment we leave aside the question of the interpreter in the 11/10 although we note that in stand-alone mode there should be some facilities to allow direct access to CAMAC from the 11/10 console.

Distribution

CCI Systems Software section
CUC Members
MPS Group Leaders
PS Programmers
E. Asséo
G. Baribaud
S. Battisti
J. Boucheron
A. Cheretakis
B. Frammery
G.L. Munday

Figure 1 - acquisition table

Figure 2 - surveillance table

Figure 3 - request read table