Title
SOME CHARACTERISTICS OF INTERFACES BETWEEN CAMAC AND SMALL COMPUTERS

Permalink
https://escholarship.org/uc/item/6pb7d93r

Author
Kirsten, Frederick A.

Publication Date
1970-12-01
SOME CHARACTERISTICS OF INTERFACES BETWEEN CAMAC AND SMALL COMPUTERS

Frederick A. Kirsten

December 1970

AEC Contract No. W-7405-eng-48

TWO-WEEK LOAN COPY

This is a Library Circulating Copy which may be borrowed for two weeks. For a personal retention copy, call Tech. Info. Division, Ext. 5545


DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.
SOME CHARACTERISTICS OF INTERFACES BETWEEN CAMAC AND SMALL COMPUTERS

Frederick A. Kirsten
Lawrence Radiation Laboratory
Berkeley, California 94720

Summary

The typical CAMAC system has a small computer attached to it. In the usual case, the computer acts as a repository for data generated by CAMAC, and also controls and directs the CAMAC operations. This paper discusses some of the aspects of the hardware 'interface, and also some of the interaction between the computer and the CAMAC system.

Introduction

CAMAC systems can be assembled in many shapes, sizes, and configurations. The flexibility of CAMAC, which permits the accommodation of a wide range of problems, is one of its greatest assets. There exist, in actuality or on paper, systems of all sizes, from those completely self-controlled and contained in a single crate, to multibranch, multi-computer systems involving a very high level of data interchange and communication.

This paper concentrates on a specific, relatively simple type of CAMAC system—namely, one whose crates are interconnected by a single Branch Highway, and controlled by a single source of system control—a computer. The discussion centers around the problems of interaction and intercommunication between computer and the CAMAC system, and the hardware necessary to effect the interfacing.

It is assumed that the reader is familiar with both the CAMAC Crate Dataway, the CAMAC Branch Highway, and the Crate Controller Type A. These are described in the official specifications and in other papers of this series. In addition, familiarity with the basic facilities of a typical small computer is assumed.

The "Typical" Small Computer

The computer interacts with CAMAC via what is often called its "10 structure." This term refers to the capabilities of the computer to communicate with device external to itself. (IO is the abbreviation for Input-Output.) For the purposes of discussion, the "typical" small computer is assumed to have the following characteristics in its IO structure.

(a) A programmed IO facility. This refers to the ability to transfer data between computer and external device under the complete step-by-step control of the computer program. In general, the transfer of each word, or the emission of each command, is the result of one or more instructions in the program.

(b) A block transfer facility: This permits transfers of blocks (large numbers of data words) in such a way that the computer program is required to issue only the initial instructions for the block. The remainder of the operation proceeds under hardware control.

(c) An interrupt facility, which permits the external device (CAMAC) to gain the attention of the computer program.

(d) An IO command repertoire, usually relatively limited, by which specific IO operations can be commanded.

Comparison of Crate Controller and Branch Driver Interfacing

The interface between a computer and a CAMAC system is usually placed either at the Crate Controllers of the individual crates, or at the Branch Driver of a Branch Highway system. Figure 1 shows an example of each configuration. (Systems have also been built with the interface in a module designed for use at a normal crate station. These usually require an auxiliary "control highway." This configuration is not considered here.) The prime purpose of either of the two configurations under discussion is to provide a vehicle for the communications between computer and modules. The main difference between the two is in the organization of the conversation. The Branch system shown in Fig. 1a provides a single port for the computer to "talk" to the entire system. The other system, shown in Fig. 1b, has an interface at each crate. Let us call it a "radial" system, since the communication radiates from the single computer to the several crate controllers.

A given collection of CAMAC modules can often be interfaced either way. Both ways are applicable to single-crate and multiclete systems. Which method

![Fig. 1 Two examples of how a three-crate CAMAC system may be configured: (a) using a CAMAC Branch Highway, and (b) using a computer IO bus to interconnect the crates.](attachment://image.png)
block transfers of data involving modules from several crates can probably be handled more easily with the Branch Highway—the crate boundaries can be made transparent to the computer interface. On the other hand, simultaneous block transfers to two or more separate modules, using, e.g., interleaved cycles on two separate data channels, can possibly be done more easily in the radial system, particularly if the modules are in separate crates.

**CAMAC-Computer Communication**

Certain common problems arise whenever two data structures are interfaced. This is no less true if one of the data structures is CAMAC. In the following, we discuss how some of these common problems relate to the CAMAC-computer interface.

Table I compares some characteristics of CAMAC and the IO structures of typical small computers. The table substantiates that translations are necessary—from the IO "language" of the computer to the CAMAC "language," or vice versa. The entities to be translated include the size of data words, commands (i.e., the significance of commands), and timing sequences.

Timing problems are often solved by 'staticizing' the data and commands. This is done by providing registers accessible from either side of the interface. (A register is a one-data-word or one-command-word storage device, probably composed of one flip-flop for each bit involved.) The registers behave as "mailboxes." One entity, CAMAC or computer, deposits information in the registers, using its own cycle timing and sequencing. The other entity, using its own cycle timing quirks, can then pick up the information. One entity may completely finish an operation before the other begins, thus removing the necessity for interlocking and synchronizing two different sets of cycle timing characteristics.

In addition to the registers, an appropriately designed control sequencer and organizer coordinates the interlocking of the complete operations of CAMAC and of the computer.

**A Typical Interface**

Figures 2 and 3 are simplified block diagrams showing the basic parts of a typical CAMAC-computer interface. Figure 2 shows the parts primarily concerned with the transmission of data; Fig. 3 shows portions concerned with the transmission of CAMAC commands and control features.

**Data-Oriented Parts**

Figure 2 illustrates the flow of data to and from the Branch Highway bidirectional Read/Write (BRW) bus, and to and from the computer. Some data are transferred between computer and interface via programmed IO (accumulator) transfers under complete software control. This means every step of the process is specifically controlled by the computer program; the interface has a relatively simple task. Other data are transferred in block transfer mode [shown here as Data Channel (DCH) transfers]. In this mode, the computer program initializes the system, but each individual data word in the block is transferred under autonomous (hardware) control. Depending on the computer, more or less of this hardware must be in the interface.

**Data Registers.** Figure 3 shows two data registers, one for Programmed Input-Output (Prog IO) transfers and
TABLE I

Comparison of characteristics of CAMAC and typical small computers

<table>
<thead>
<tr>
<th>Topic</th>
<th>CAMAC characteristics</th>
<th>Characteristics of the IO structure of a typical small computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Types of operations that may be carried out</td>
<td>Bidirectional data transfer</td>
<td>Bidirectional data transfer [two modes: Programmed IO (accumulator transfers), autonomous transfers (e.g., data channel)]</td>
</tr>
<tr>
<td></td>
<td>Command transfers to modules [A much larger command repertoire than the average computer IO structure]</td>
<td>Command transfers to peripheral devices [different command structure from CAMAC]</td>
</tr>
<tr>
<td></td>
<td>Look-at-Me service requests</td>
<td>Interrupts or Data Channel cycle requests</td>
</tr>
<tr>
<td>Basic cycle time</td>
<td>Cycle time &gt; 1 usec (depends on physical length of Branch Highway line)</td>
<td>Cycle time varies from one computer to next</td>
</tr>
<tr>
<td>Timing and logical control of operations</td>
<td>Specified by CAMAC specifications</td>
<td>Different for each computer (and different from CAMAC)</td>
</tr>
<tr>
<td>Data word size</td>
<td>Maximum word size = 24 bits. Actual word size (no. of bits that are actually significant) depends on the module involved</td>
<td>Word size is generally less than 24 bits</td>
</tr>
<tr>
<td>Service requests/interrupts</td>
<td>At the Crate Controller, there are available up to 23 bits of service-request information at the Branch Driver, there is 1 bit (the BD signal) available at all times; after a GL cycle, there are 24 bits available. More can be obtained by addressed commands</td>
<td>Service requests include: Interrupts: single-level or multiple levels; Data Channel cycle requests: possibly on more than one channel; generally, transfer can be made to or from computer; Direct memory access cycle request. The first can be spontaneous; the latter two must usually be &quot;set up&quot; by a preceding routine</td>
</tr>
</tbody>
</table>

one for Data Channel (DCH) transfers. Two registers are necessary because a DCH cycle is often interleaved with steps of a Prog IO transfer. The number of bits in each register is the larger of the CAMAC or computer word sizes. Since the typical small computer has word sizes of 12, 16, or 18 bits, CAMAC usually wins, and the registers therefore have 24 bits.

**GL register.** A third register, involved only in data flow from CAMAC to computer, is the GL register. During the Graded-L cycle, a 24-bit word carrying information on the status of L requests in the Branch is generated on the BRW bus. The GL register saves this information and makes it available either to the computer via Prog IO transfer or to internal portions of the interface for control of block transfers.

**Command-Oriented Parts**

Figure 3 shows the parts of the typical interface that are concerned with issuing addressed and unaddressed CAMAC commands, and with the BD and DQ signals.

**Command registers.** As with the data, separate command registers are provided for Prog IO and DCH transfers to permit interleaved Prog IO and DCH cycles. In most cases, the CAMAC commands are generated by the computer software, and are transmitted by the computer to the command register in a form the computer considers as data. Thus, the loading by the computer of the command register is similar to loading the data register.

To put it another way, the command repertoire of the average computer IO structure is limited, and is much smaller than the CAMAC command repertoire. Thus, there is usually no way a direct IO command to CAMAC command translation can be accomplished.

A complete Branch Highway command is 17 bits—3 bits for Crate address, 5 for Station Number, 4 for Subaddress, and 5 for Function Code. Thus, double word transfers to load the Command Register may be required for computer word sizes of 12 or 16 bits. Commands addressed to more than one crate or more than one station require special handling.

**Q responses.** Certain CAMAC commands [e.g., F(0), F(8), F(16), and F(27)] require that the module give a
In this section, the sequences of events in several types of operations are briefly explained, with emphasis placed on the interplay between the computer and the CAMAC system.

Programmed IO Data Transfers

In these operations the object is to move data between (the accumulator of) the computer and a specific CAMAC address (e.g., a register in a module). The computer program is in control in that it specifies the direction of data flow, the CAMAC address, and when the operation is to take place.

Two examples are given in Table II, one for a Write operation in which data move from computer to module, and one for a Read operation.

### Table II

**Sequences for programmed IO Data Transfers**

<table>
<thead>
<tr>
<th>Computer operation</th>
<th>CAMAC operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. WRITE (Data flow from computer to CAMAC address)</td>
<td>IDLE</td>
</tr>
<tr>
<td>A. Load data into the &quot;Prog 10 data reg&quot; in interface. (This may be a double transfer, if data word size &gt; computer word size.)</td>
<td>IDLE</td>
</tr>
<tr>
<td>B. Load CAMAC command into &quot;Prog 10 command reg&quot; in interface. (This may also be a double word transfer.)</td>
<td>C. Execute Branch Highway cycle, transferring data loaded in A using CAMAC address loaded in B.</td>
</tr>
<tr>
<td>† Wait for signal that Branch Highway cycle is done (e.g., Program loop using &quot;Skip if BH cycle done&quot;).</td>
<td></td>
</tr>
<tr>
<td>† D. Do next sequence — — —</td>
<td></td>
</tr>
<tr>
<td>II. READ (Data flows from CAMAC address to computer.)</td>
<td>IDLE</td>
</tr>
<tr>
<td>A. Load CAMAC command into &quot;Prog 10 command reg.&quot;</td>
<td>B. Execute Branch Highway cycle, transferring data from CAMAC address given in A to &quot;Prog 10 data reg.&quot;</td>
</tr>
<tr>
<td>† Wait for signal that Branch Highway cycle is done.</td>
<td>† IDLE</td>
</tr>
<tr>
<td>† C. Examine BQ register. If BQ (contents of BQ register) = 1, data are ok; if BQ = 0, data are invalid. Program may use &quot;SKIP if BQ = 1&quot;</td>
<td></td>
</tr>
<tr>
<td>D. Assuming BQ = 1, transfer data from &quot;Prog 10 data reg.&quot; to computer</td>
<td>E. Do next sequence.</td>
</tr>
</tbody>
</table>

**Q response.** On the Branch Highway, all Q responses from individual crates are gathered onto the BQ line. The state of the BQ line may be strobed into a one-bit BQ register during each Prog IO-initiated CAMAC cycle and made available to the computer program.

The program may test this register and may conditionally branch, depending on its state. A second one-bit BQ register may be required for block transfers that use the sequential addressing mode.

### Operational Sequences

In this section, the sequences of events in several types of operations are briefly explained, with emphasis placed on the interplay between the computer and the CAMAC system.

**Programmed IO Data Transfers**

In these operations the object is to move data between (the accumulator of) the computer and a specific CAMAC address (e.g., a register in a module). The computer program is in control in that it specifies the direction of data flow, the CAMAC address, and when the operation is to take place.

Two examples are given in Table II, one for a Write operation in which data move from computer to module, and one for a Read operation.

**Block Transfer Operations**

In block transfer operations, the object is to move blocks (many words) of data with the minimum intervention on the part of the computer program. Whereas the initiation of the block is under program control, the transfer of individual data words is controlled by relatively simple hardware logic.

Generally, the same CAMAC command—e.g., F(0)—is used during the entire block. However, two modes of CAMAC addressing are available—stationary and sequential. In stationary, the entire block transfer involves only one subaddress in one CAMAC module—the CNA (in CNAF) remains constant. An example would be the transmission of a logical record of data to a CAMAC module that interfaces to a magnetic tape unit.

In sequential addressing, successive data words come from or go to sequential locations in the set of CAMAC modules. Relatively simple hardware can be used to increment the CAMAC address (CNA) after each Branch Highway cycle according to an algorithm described elsewhere.4

Table III shows sequences of cycles that occur for two examples of block transfers.
TABLE III

Some sequences for Block Transfers of data via Data Channel

<table>
<thead>
<tr>
<th>Computer operation</th>
<th>CAMAC operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. Set up operation with Prog IO transfers:</td>
<td>______________________________</td>
</tr>
<tr>
<td>A. Load appropriate CAMAC commands into &quot;DCH command reg.&quot;</td>
<td>Command is interpreted by interface to learn direction of data transfer—e.g., F(0), Read Group 1 Register, is transfer from module to computer.</td>
</tr>
<tr>
<td>B. Load &quot;mode of transfer&quot; into control register of interface either separately or in same step as A. Mode determines whether block transfer is to or from a single module (stationary), or a block of modules (sequential address).</td>
<td></td>
</tr>
<tr>
<td>II. Cycle by cycle: Once the setup is done, data words are transferred, one per cycle, until the block transfer is done.</td>
<td>______________________________</td>
</tr>
</tbody>
</table>

STATIONARY ADDRESSING MODE: "Read" Block Transfer

A. Branch Highway cycle to fetch data word from the module to interface. This cycle may be triggered by an L request from the module.

B. Data Channel cycle sends data to computer memory.

Repeat A. and B. until entire block has been read.

SEQUENTIAL ADDRESSING MODE: "Read" Block Transfer

A. Branch Highway cycle.

B. Examine the BQ register (this is done in the interface).

C. Transfer data to computer memory if BQ = 1. If BQ = 0, do not transfer data.

D. Advance CAMAC address to get next data word from next register in the block of CAMAC modules (this is done in the hardware of the interface).

Repeat A, B, C, and D until entire block has been read.

TABLE IV

Examples of L requests and appropriate responses

<table>
<thead>
<tr>
<th>Type of module</th>
<th>Meaning of request</th>
<th>Proper service (besides resetting the L-request flag)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Analog-to-Digital Converter</td>
<td>A data word is ready for transfer</td>
<td>Transfer the data</td>
</tr>
<tr>
<td>(b) Scaler</td>
<td>Scaler has overflowed</td>
<td>Increment a computer memory location that is used to count overflows.</td>
</tr>
<tr>
<td>(c) Typewriter control</td>
<td>Typing of previously received character is completed</td>
<td>Send another character if there is one</td>
</tr>
<tr>
<td>(d) Magnetic tape Controller</td>
<td>Previously received word has been recorded on tape</td>
<td>Initiate data-channel cycle to send another word</td>
</tr>
<tr>
<td>(e) Arbitrary module</td>
<td>No significance in this particular system</td>
<td>Ignore it</td>
</tr>
</tbody>
</table>
Modules have different reasons for requesting service; they request different types of service; and the requests from different modules carry differing degrees of urgency. The computer and (or) computer interface have the duty of interpreting the requests and of initiating the proper action.

To make this more explicit, let us examine some cases. Table IV lists some of the reasons why a module would generate an L request.

The proper servicing of the requests may require different actions from the computer and its interface, depending on what it has to offer. These actions may include the following:

1. interrupt the computer on one or more levels of priority;
2. execute a data-channel cycle to transfer data into the computer memory;
3. execute a data-channel cycle to transfer data from the computer memory (there may be more than one data channel available for (ii) and (iii));
4. execute an "increment contents of memory" cycle at a specified memory address; and
5. take no action—cf. (iv) above.

The problem involved in matching the action to the request exists in the Crate Controller to computer interface and in the Branch Driver-to-computer interface. Let us concentrate on the latter, which is somewhat more complex.

The Branch Demand (BD) signal of the Branch Highway is the summation of all service requests from all crates. By itself, it carries so little information that it almost inevitably results in further action. Figure 4 shows graphically the tree-like structure that can be imagined as representing the L-source identification process.

The BD signal carries one bit of information. If there is more than one source of L requests, a GL cycle is executed. This immediately makes 24 bits of information available. (On the basis of the GL word, the computer interface must decide what computer action is required either to service the request, or, if insufficient data are in the GL word, to further identify the source of the L request.)

If there are 24 or fewer sources of L request, the search stops here; the requesting module can be identified at this point. If there are more than 24 sources, then the GL word can indicate only the direction the software search should take in order to reduce the searching time.

The allocation of searching chores to hardware and software is done simply on the basis of convenience. It is quite easy to automatically execute a GL cycle whenever a BD appears. It is much more complicated to do the next higher level of searching with hardware, hence we "resort to" software.

**Sequence.** The sequence of events following the initiation of a typical L request follows:

(a) Module X raises its L request (L+1).
(b) This results in the Branch Demand going to the true state (BD+1).
(c) The Branch Driver recognizes that bit Z is in the L state. Having been prewired to do so, it initiates the proper computer action. Let us imagine that this action is to interrupt the computer, setting into motion a service program.
(d) The nature of the specific L request now being fully recognized, the service program does the appropriate servicing that the L request had originally asked for.
(e) During the service program, the L flag in module X is reset, perhaps with an F(0)—Clear Look-at-Me command. (If the module is so designed, the L flag may have automatically been reset at the instant the servicing was done.)
(f) After L from module X is reset, BD returns to 0 unless another L from another module has appeared in the meantime.

**Graded-L facility**

A Branch Highway could conceivably have as many as 161 stations (7 crates x 23 stations/crate = 161 stations), each with one source of Look-at-Me (L).

Since the GL word has only 24 bits, some means of condensing or summarizing the Look-at-Me information must be provided. For this purpose, each Crate Controller type A (CCA) has a facility for performing logical operations on the 23 L signals from the Data-way, and uses the outputs of the logical operations as its contribution to the GL word. Figure 5 shows the situation schematically. The logical operations in each CCA are in the block called LAM grader. The outputs of each LAM grader, appropriately wired by the BG signal, are wired-OR'd onto the BRM bus. The GL word is therefore a composite of contributions by each LAM grader. The logic in each LAM grader can be chosen, by the person who designs the particular system, in any way that facilitates the identification of the L source. In other words, the LAM grader provides flexibility and adaptability in matching the
Additional searching by computer software

The 24-bit GL word

The 1-bit BD signal

Fig. 4 This tree-like structure illustrates the possible steps in identifying the source of an L request.

Pattern of L sources to the services available from the computer.

References


2. "CAMAC, Organization of Multi-Crate Systems, Specification of the Branch Highway and CAMAC Crate Controller Type A," to be published as EUR-4600e. Requests for information should be directed as shown under reference 1.

3. F. A. Kirsten, "Operational Characteristics of the CAMAC Dataway," UCRL 20214. (See Note)

4. F. A. Kirsten, "A Short Description of the CAMAC Branch Highway," UCRL 20217. (See Note)

5. R. S. Larsen, "CAMAC Dataway and Branch Highway Signal Standards." (See Note)

NOTE: References 4-7 and this paper were presented in the CAMAC tutorial session of the 1970 Nuclear Science Symposium, New York, November 4-6, 1970. They are scheduled to be published in the IEEE Transactions on Nuclear Science, April 1971.
LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.