Title

OPERATIONAL CHARACTERISTICS OF THE CAMAC DATAWAY

Author
Kirsten, Frederick A.

Publication Date
1973-02-01

To be published in
IEEE Trans. on Nuclear Science

LBL-1575
Preprint c.1

OPERATIONAL CHARACTERISTICS OF THE CAMAC DATAWAY

Frederick A. Kirsten

February 1973

Prepared for the U.S. Atomic Energy Commission
under Contract W-7405-ENG-48

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 thc 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 ncccssarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

OPERATIONAL CHARACTERISTICS OF THE CAMAC DATAWAY*

Frederick A. Kirsten
Lawrence Berkeley Laboratory
Berkeley, California

Summary

The basic purpose of CAMAC is to provide a standardized method for transferring data and control information between instrumentation modules and a digital controller. CAMAC encompasses both a hardware standard for housing the modular components of a system and an electrical and logical standard for the control "language" used to effect the information transfer. The CAMAC specification[1] contains the formally stated rules for both of these aspects of CAMAC. This paper concentrates on the control language and describes some of its features and uses in a less formal way than in the specification, and therefore may be useful as an introduction to the subject.

Other papers in this series[2-6] consider other aspects of CAMAC, including its place in the context of instrumentation systems, hardware aspects, signal standards, the Branch Highway, and coupling to computers and control systems. Here, the scope is limited to the process of information interchange within the CAMAC crate.

This is a revised version of a paper previously published.[7] Since the first version was published, a revised version of the CAMAC specification has been issued. This present paper is based only on the revised specification.[1] Information on the differences between the original and revised CAMAC specifications is available.[8-10]

Introduction

Levels of Standardization

The value of any standard depends strongly on how faithfully it is followed. The interchangeability of CAMAC components--either as pieces of hardware or as designs recorded on paper--is one of the most important goals of this standardization effort. In fact, the attainment of this objective is probably more important than the exact nature of the CAMAC standard itself. Anyone who designs to a standard is often irritated by features that he feels are not optimized for his immediate application--no standard can be optimum for all purposes. However, before giving in to the temptation to modify the standard, the designer must consider the long-range benefits of standardization--interchangeability, reusability, and compatibility with components acquired from diverse sources. In addition, the time required to design a new module can be reduced if a consistent and constant set of design rules, such as CAMAC is used.

One must also be aware that, within CAMAC, the effects of standardization become apparent at various levels, such as the Dataway level, the systems level, and the software level. The basic CAMAC Dataway specification[1] often provides several ways in which a given operation may be accomplished. Often, because of systems or software considerations, one of these ways is preferable. One might use the analogy of language and conversation. The Dataway specification defines the language of communication. It completely documents the rules that insure compatibility of plug-ins with the crate in a hardware sense and in a language sense. It carefully defines the words of the language and the rules by which the words are transmitted and received. The system uses this language to carry on a conversation. The exact nature of the conversation depends in part on the tasks to be accomplished. It also depends on which set of words controllers and modules understand. It may also depend on a mutual agreement on certain strings of words (e.g., block transfers). Perhaps there are even some semantic problems--for example, when a Read command is executed, is the module's L-flag automatically lowered, or does this require a separate operation?

One problem that the user faces is the determination of how much the system does for him more or less automatically, and how much planning and organizing he must do. He must learn what to look for in reading the data sheets that describe the components of his system. The point is: CAMAC is intended to service a wide range of applications. Therefore, flexibility must exist in the way the system can be employed. And therefore, the system cannot be defined in a rigid and inflexible way. Standardization can exist only to a certain level, beyond which the user or system designer supplies the rules.

Some CAMAC Terms

It may be helpful to review some of the terms that have specific meanings and usages in the CAMAC specifications. Other terms will be introduced in the appropriate place in the text. In the following paragraphs of this section, the appearance of a term having a specific CAMAC significance is noted by underlining.

Crate. The CAMAC crate has up to 25 stations (slots) for accommodating CAMAC plug-ins. Each station has an 86-pin card-edge connector by which the plug-ins gain access to the wiring of the crate (Dataway).

The stations are numbered from left to right as the crate is viewed from the front. All references in this paper are to a crate of maximum capacity--i.e., with 25 stations.

Dataway. The Dataway is the wiring that interconnects the 25 card-edge connectors. It includes signal wires and power wires.

Most of the signal wiring accessible at normal stations 1 through 24 is bussed. The same signal is available at the same pin on all those connectors, as demonstrated in Fig. 1. The wiring for the Station Number (N) and Look-at-Me (L) signals is point-to-point, fanning out from the connector at the control station 25. For data transfer, the Dataway contains two 24-bit parallel unidirectional data buses (highways). One is for data transfers from crate controller to module. The other is for module to crate controller transfers. The Dataway also has all the wiring necessary for carrying the specified signals between modules and controllers. The signals carried on the Dataway are sufficient to carry out any Dataway operation described in the specification.

*Work performed under the auspices of the U.S. Atomic Energy Commission.

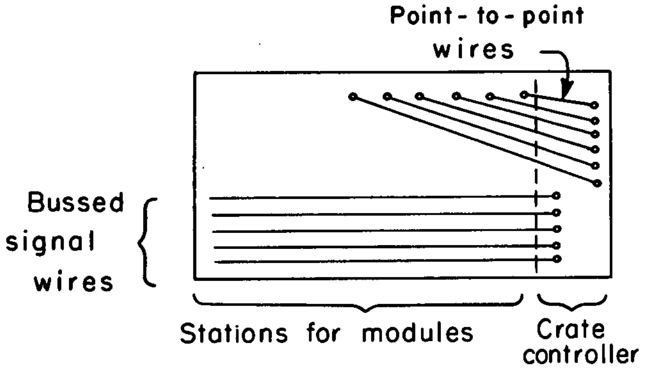


Fig. 1. A sketch illustrating a small part of the Dataway wiring. Both point-to-point and bussed signal wires are shown. Bussed signals are available at all stations; point-to-point signals travel only between specific stations and the crate controller.

The Dataway also has buses for supplying power to the plug-ins. The voltages carried on these buses are specified, and the allowable maximum currents drained by each plug-in are fixed.

Plug-in. The generic term for the unit that is accommodated by the stations of the crate is "plug-in". (There is a great temptation to use the term "module" in this context. However, in CAMAC, "module" is used in a different way.) There are two types of plug-ins--modules and controllers.

Crate Controller. The crate controller (often abbreviated as CC) is the master plug-in that controls the flow of signals in the crate. In order to do this, it must have (a) access to the bussed wires available at any of stations 1 through 24, and (b) access to certain connections (N and L) available only at station 25. Thus, it always resides at the extreme right end of the crate, and uses at least two Dataway stations (e.g., stations 24 and 25 in the 25-station crate assumed throughout). Every crate must have a crate controller.

The crate controller functions as an interface, and hence we can think of its two parts (faces). One part acts to control and coordinate the crate and its modules. This action is well defined by the CAMAC specification (which also specifies the hardware design of the CC).

The other part "talks to" (or listen to) the seat of system control, which could be a computer, hardware digital control device, or--in a very simple system--a simple control logic built into the CC itself. With one exception, the CAMAC specification is not concerned with this; the system designer therefore builds or buys a crate controller designed to interface with his particular control system. The exception is that there is a CAMAC specification[11] for the CC that interfaces the crate to the CAMAC Branch Highway. This CC is known as the Crate Controller Type A.

Module. A module is a plug-in that can be mounted in one or more normal stations. In usual circumstances modules are slaves, answering (with one exception, the Look-at-Me signal) only when spoken to by the crate controller.

Modules have at least two parts. One (part A) is the part that performs a task useful to the overall instrumentation system--the raison d'être of the module. Useful tasks include accepting and storing data, performing digitizing operations, controlling external devices, etc. The other (part B) is the part that "speaks CAMAC". Its job is to facilitate the conversation between part A and the "brains" of the system, be it computer or other forms of digital controller. The conversation is via the medium of the Dataway and the CC. In this paper, we concentrate on part B, but let us not forget the purpose of part B (and CAMAC) is to facilitate the conversation.

System control. CAMAC modules do not control themselves. Every system must have a "system control." The nature, location, and capabilities of the system control vary from one system to another. But it always has the tasks of issuing the necessary CAMAC commands to make the system fulfill its duties and of interpreting the service requests (Look-at-Me's) from the modules. In most cases, the primary system control is the program (software) of a computer. The program decides which are the necessary commands, and orders some piece of hardware to carry them out. In other systems, the control is entirely in hardware form, with the responses being whatever "reflex" actions are wired into the controller. In some cases the control is distributed throughout the system. Others use a combination of hardware and software control.

Elements of Communication on the Dataway

Table I lists the signals available at the 86-pin connectors of all stations except the control station (25). It also gives the purpose of each signal.

Figure 2 depicts the communication and control paths that exist in the crate. The provisions for data flow from CC to module or vice versa are nearly symmetric. The two 24-bit data buses--the R and W buses--provide equal facilities in the two directions. The scale is tipped slightly by the Q and X buses, which provides for the flow of two bits of status information from module to CC.

However, the control provisions are highly asymmetric. The only control-like signal originating in the module is the L (Look-at-Me). All other facilities are normally used for commands from the CC to the modules.

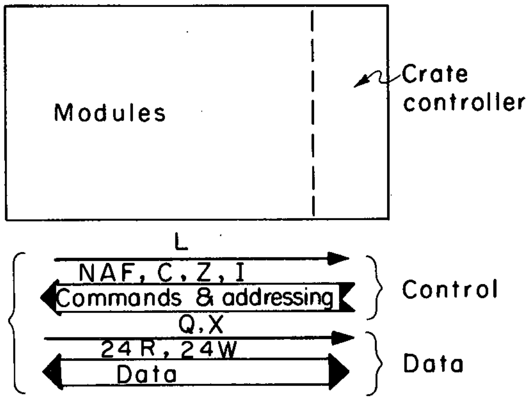


Fig. 2. This sketch summarizes the capability of the Dataway in transmitting data and controlling signals from Crate Controller to module and vice versa.

TABLE I. A list of Dataway signals available at each of the normal stations 1 through 24 of a 25-station CAMAC crate

Title	Designation	Use in module
Commands, unaddressed		
Initialize	Z	Sets registers or control functions in a module to an initial state, particularly when power turned on.
Inhibit	I	Disables features for duration of signal.
Clear	C	Clears registers, or resets flip-flops
Commands, addressed		
Function codes	F1,2,4,8,16	Carried on Dataway in binary code. Defines the function to be performed in a module during command operations.
Addressing		
Station number	N	Selects the module. There is an individual line from crate controller to each station.
Subaddress	A1,2,4,8	Also binary coded. Selects a location, within the module, to which the command is directed. There are 16 possible subaddresses.
Data		
Read bus	R1-R24	Transmits digital information from module to Crate controller. Format is bit-parallel words, 24 bits maximum.
Write bus	W1-W24	Transmits digital information from crate controller to module. Format is same as for Read bus.
Timing		
Strobe 1 and Strobe 2	S1,S2	These strobes are generated by CC during every Dataway operation. Used by modules for timing acceptance of data or execution of features of an operation.
Status		
Look-at-Me	L	A signal from module to crate controller indicating request for service or attention. There is an individual line from each module to control status.
Q-Response	Q	A one-bit reply by module to certain commands issued by Crate Controller.
Command Accepted	X	Indicates the ability of a module to execute the current command operation.
Busy	B	Indicates a Dataway operation is in progress.

Notes:
1. All signals except N and L are bussed.
2. Reserved bus, patch wiring points, and power lines not listed.

The crate controller is, practically speaking, the only device in the crate that can issue commands. The modules are slaves to the controller.*

The R and W buses provide capacity for bit-parallel transfer of words up to 24 bits in length. There is no standard word size in CAMAC. Only the maximum size is specified. Modules may use words of from 1 to 24 bits. Within these limits the designer has freedom to choose the size best fitted for his task. (He will do well, however, to consider the effect of his choice on system operation. For example, a 17-bit word is a poor choice in view of the popularity of the 16-bit word size in small computers).

As mentioned earlier a separate N and a separate L line are wired from each of the normal stations 1-24 to the control station 25. This configuration of N and L wires has two important consequences:

*Of course, all command signals except N are bussed. Therefore, in theory any module can issue all elements of a command except N. However, the use of this possibility is not often seen. To simplify the reading (and writing) it is ignored in this paper.

(a) A module is addressed by the number of the station it is plugged into. This means the system control must be supplied a list of the physical locations of all modules. If the location is changed, the list is changed. This is in contrast with other systems in which a module is assigned a permanent "device code."

(b) The crate controller is the only device that has access to the N and L signals. It is therefore the only device that can issue a complete Dataway command--i.e., the only "master" in the crate.

Dataway Operations

In a specification, certain terms must be accurately defined and used carefully in order that the exact intent of the specification be conveyed to the reader. Insofar as possible, terms used in the CAMAC specification will be used in the same way in this paper. For example, Dataway operations refer to orders issued by the CC to the module(s). They can be of two types: command operations are orders addressed to a specific module or modules; unaddressed operations are orders which are sent in parallel to all modules. All Dataway operations are accompanied by the Busy signal. According to strict usage, a module's request for service via its Look-at-Me signal is not a Dataway operation, and is therefore discussed under a separate heading. Likewise the common control signal, Inhibit, is not accompanied by Busy, and is therefore not classed as a Dataway operation.

All Dataway operations are carried on in cycles having defined timing characteristics.

Commands

A command is a set of Dataway signals that tell a particular module to perform a particular operation. The set of signals consists of the codes for each of the numbers N, A, and F. N and A together denote the address of a specific register or piece of logic within a specific module--N is a station number in the crate, and A is a subaddress within the module at the station number. The N signal is received only by the module in the addressed station. The A signals are bussed and are therefore available to every module. In a given operation, they are significant only to the module in the addressed station. The binary coded value of the subaddress is carried on four Dataway wires designated A1, A2, A4, A8. There are 16 possible subaddresses. The subaddresses can therefore be used to steer the command to one of sixteen possible locations within a module.

F denotes the function to be executed during the cycle. The value of F is carried in a binary code on the five Dataway buses F1, F2, F4, F8, and F16. There are 32 possible function codes.

Let us refer to the canonical form of the addressed command on the Dataway as NAF, and to a specific command as

$$\text{Command} = N(i)\ A(j)\ F(k),$$

where:

N(i) = the station to which the command is addressed, $1 \leq i \leq 23$;

A(j) = the subaddress of the module in station N(i), $0 \leq j \leq 15$; and

F(k) = the operation (function) to be executed by the addressed device; $0 \leq k \leq 31$.

The command N(i) A(j) F(k) implies that the logic designated as subaddress A(j) within the module plugged into station N(i) will perform the operation commanded by the function code F(k).

A confusion in symbology may arise because of the use of both ordinal and binary codes. The form F(k) will be used to represent the ordinal value of a code, whereas symbols like F1, F2, etc., represent the elements of the binary code representing F(k). In speaking of a specific command N(i) A(j) F(k), we are using the ordinal values for N, A, and F. Thus, if we wish to do a Selective Set Group 2 Register operation, we ask the CC to execute the function code expressed in ordinal form as F(19). To do so, the CC sends out the binary code for F(19) which has F1 = F2 = F16 = logic '1', and F4 = F8 = logic '0'. Both A and F are carried on the Dataway in a binary form, although in writing the values or consulting the function-code table ordinal representations are more convenient.

The station number value, N, is carried on the Dataway in the form of an ordinal code, with a separate wire for each value of N--one wire for each station. As a consequence, it is possible to address more than one station during a given command. This can be useful for operations such as Clear or Execute, where it may be desired to have the operations carried out simultaneously in a number of modules.

It may be worth pointing out that the designations for a command are slightly different from the module's point of view than for CC's or for the computer program that operates a system. On data sheets describing modules, the commands are of the form N A(j) F(k). If the module is plugged into station 10, then, and only then, $N \equiv N(10)$.

Function codes

A complete list of the 32 function codes is shown in Table II, reproduced from the specification.[1] Eighteen of the codes have been assigned specific meanings. The others are either reserved (may not be used until assigned specific meaning in the CAMAC specification), or are non-standard. Non-standard function codes may be used by the designer to implement operations which are not provided by any of the standard function codes. Additional details of the standard functions are given in the specification.

Note that the codes may be categorized into three groups:

(1) One group is used in the transfer of data from modules to CC via the R lines. All codes in this group are characterized by F16 = F8 = 0.

(2) Another group is used with the transfer of data from CC to module via the W lines. These codes all have F16 = 1, F8 = 0.

(3) The third group is not involved with data transfers. These codes have F8 = 1.

Notes on function codes. The distinction between Group 1 and Group 2 registers [cf. function codes F(0) and F(1)] is somewhat nebulous. The groups are usually thought of as representing data and module control registers, respectively. The specification states (Section 6.) "Information concerning status or system organization, or requiring restricted access, should

TABLE II. A list of the 32 CAMAC function codes. Columns: F1, 2, 4, 8, and 16 list the logic states of the signal wires used to transmit the binary coded form of the function codes.

NO.	Function name		F 16	F 8	F 4	F 2	F 1
0	Read Group 1 Register	Functions that use the R lines	0	0	0	0	0
1	Read Group 2 Register		0	0	0	0	1
2	Read and Clear Group 1 Register		0	0	0	1	0
3	Read Complement of Group 1 Register		0	0	0	1	1
4	Nonstandard		0	0	1	0	0
5	Reserved		0	0	1	0	1
6	Nonstandard		0	0	1	1	0
7	Reserved		0	0	1	1	1
8	Test Look-at-Me	These functions do not use the R or W lines	0	1	0	0	0
9	Clear Group 1 Register		0	1	0	0	1
10	Clear Look at Me		0	1	0	1	0
11	Clear Group 2 Register		0	1	0	1	1
12	Nonstandard		0	1	1	0	0
13	Reserved		0	1	1	0	1
14	Nonstandard		0	1	1	1	0
15	Reserved		0	1	1	1	1
16	Overwrite Group 1 Register	Functions that use the W lines	1	0	0	0	0
17	Overwrite Group 2 Register		1	0	0	0	1
18	Selective Set Group 1 Register		1	0	0	1	0
19	Selective Set Group 2 Register		1	0	0	1	1
20	Nonstandard		1	0	1	0	0
21	Selective Clear Group 1 Register		1	0	1	0	1
22	Nonstandard		1	0	1	1	0
23	Selective Clear Group 2 Register		1	0	1	1	1
24	Disable	These functions do not use the R or W lines	1	1	0	0	0
25	Execute		1	1	0	0	1
26	Enable		1	1	0	1	0
27	Test Status		1	1	0	1	1
28	Nonstandard		1	1	1	0	0
29	Reserved		1	1	1	0	1
30	Nonstandard		1	1	1	1	0
31	Reserved		1	1	1	1	1

be held in Group 2 registers." It also recommends that certain Group 2 subaddresses be used for registers intended for certain Look-at-Me operations or for the Module Characteristic. It is clear that to be compatible with common usage, designers should use Group 1 for data registers.

"Overwrite" might also be defined "write into." It means that both logical ones and zeros in the data word are transferred into the addressed register. This is sometimes called the "jam transfer" operation. The Selective Set and Selective Clear operations are discussed later.

Note that there is no intention to require module designers to implement function codes that they don't need. For example, a module design that utilizes F(16) (Overwrite Group 1 register) need not include F(9) (Clear Group 1 Register) unless the designer feels he needs it. The multiplicity of codes should be thought of as options which the designer may or may not use. However, if he uses a given code, its meaning must not be changed. By the same token, the user must consult the data sheets or descriptions of the modules he intends to use in order to know which function codes are implemented in the modules.

Transmission of commands on Dataway

Figure 3 depicts the flow of an NAF command from CC to module. The command used in this example is N(1)A(4)F(0). The individual components of the command are held in the N, A, and F registers of the CC. Presumably, these registers were previously loaded by the system controller. The means for loading the registers is not shown. In this example, the registers are binary and therefore contain the binary codes for 1, 4 and 0.

The binary codes for A and F will be transmitted on the Dataway "as is." However, the number for N must be converted from binary to one-out-of 24 code.

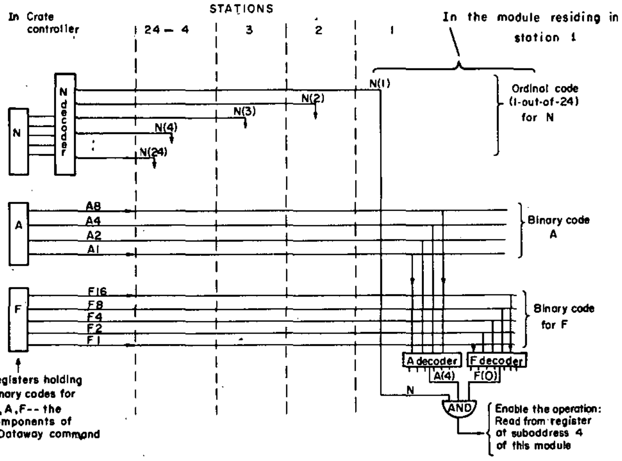

STATIONS
In Crate controller
24 — 4
3
2
1
In the module residing in station 1
N
N decoder
N(1)
N(2)
N(3)
N(4)
N(24)
Ordinal code (1-out-of-24) for N
A
A8
A4
A2
A1
Binary code A
F
F16
F8
F4
F2
F1
Binary code for F
A decoder
F decoder
A(4)
F(O)
N
AND
Enable the operation: Read from register at subaddress 4 of this module
Registers holding binary codes for N,A,F-- the components of a Dataway command

Fig. 3. Showing how an addressed command is transmitted over the Dataway, and how it is interpreted in the module.

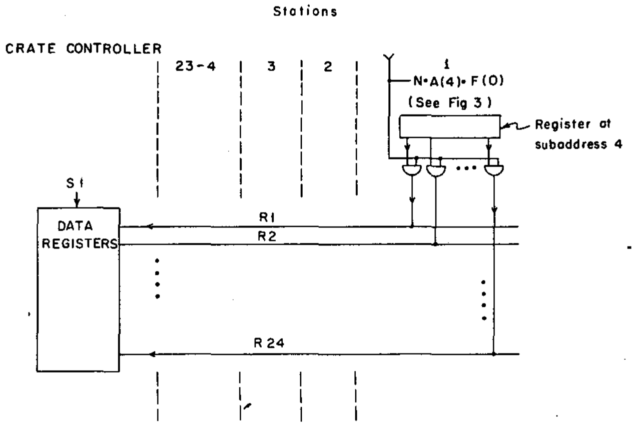


Fig. 4. This shows schematically the transmission of data from module to controller in response to a Read function code.

This is done by the N decoder. While executing this command, only the line marked N(1) will be in the logic 1 state. This is the means by which the module in station 1 is ordered to respond to the command. The binary-coded values of A and F are impressed on their respective buses, and become available at all Dataway connectors at stations 1 through 24. However, only the module in station 1 will act on these codes because, of all the N wires, only N(1) is in the logic 1 state.

Decoding of Commands in the Module

The module must contain the necessary logic for recognizing commands directed at itself, and for deciding what actions are called for. Figure 3 shows, as an example, the logic that might be used to recognize the command N A(4) F(0)--Read the register at subaddress 4 of this module. Note that the controller sends N(1) A(4) F(0); the module responds to N A(4) F(0).

It should be obvious that the signal N is an essential part of the command. It is not necessary that a subaddress be used if omission of the subaddress does not result in ambiguities. This is the case if the module responds to a given F code in one and only one way.

Transmission of Data on Dataway--Read and Write Operations

The previous section and Fig. 3 described the flow of signals on the Dataway that transmitted a Read command from CC to a module. This command directs the module to send some data back to the CC. Thus, to complete the picture of this Dataway operation, we need to show the flow of the data. This is done in Fig. 4. It shows a 24-bit data register which has been assigned the (sub) address 4.* The decoded command N·A(4) F(0) is passed along from Fig. 3. It is used to operate AND gates that connect the output of the register onto the 24 lines R1 through R24. These lines deliver the data to the DATA register in the CC. At its leisure, the system controller can access the data in the CC's DATA register for whatever purposes it had in mind. This means for this latter transfer is not shown.

Similar facilities are used for Write operations, except that data travels from a register in the CC, via the W lines, to a register in the addressed module.

Timing of Command Operations

Figures 3 and 4 painted a still-life picture of the signal flow in a command operation. In practice, time is of the essence, and the interval between the start and end of an operation must be reasonably short.

Figure 5 shows the timing of a command operation. If the minima are added, the sum is one microsecond. Thus, one command operation--the execution of one N.A.F command--requires at least one microsecond. The actual length of the cycle is determined solely by the CC--the module has no influence.** Let us assume that the CC used for our examples is adjusted for the one microsecond minimum cycle time.

The top line of Fig. 5 shows the timing of the command signals plus the Busy, B. The two levels of this line do not necessarily represent '0' and '1'. The lower level (1) indicates interval during which the signals must be established--i.e., by the time this interval begins, all signals assume the appropriate '0' or '1' states, and remain in these states until the interval ends. The upper level (2) indicates the intervals during which the signals need not be established, at least as far as the example cycle is concerned. The shaded area (3) shows an interval during which the signals may be changing. Because of rise and fall times, and propagation delays, it takes time to establish these signals.

*At least for the Read function code F(0). To direct other operations to this same register--such as writing into with F(16) or clearing with F(9)--the designer could choose other subaddresses. However, it is good practice to use the same subaddress for all operations associated with the same register. Users and computer programmers appreciate small favors like this.

**Insofar as the specification provides. The HOLD feature used by some laboratories[12] is not part of the specification.

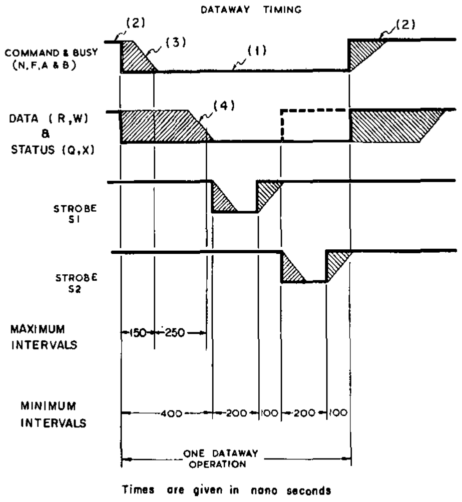


Fig. 5. Timing of a Dataway operation.

The second line shows the timing of data signals on the R and W lines (and on the Q and X lines, to be discussed later). The shaded area (4) is more generous than for the NAFB signals. The reason should be evident from an examination of Figures 3 and 4. The extra time (250 nsec) in (4) allows for propagation delays in the module in decoding the command, in the AND gates, and rise and fall times of the R-line signals.

The third and fourth lines of Fig. 5 show the timing of the Strobes S1 and S2. These are generated by the CC and are used for synchronizing certain parts of the operation. Note, for example, that S1 starts after the instant that the R and W lines are required to be stable. In Fig. 4, therefore, the CC will use S1 to time the acceptance of the data on the R lines into the DATA register.

In Write operations, the module uses S1 to time the acceptance of data on the W lines by the addressed register in the module.

Strobe S2 can be used to time other segments of the operation. For example, in a Read and Clear operation, the data from the addressed register is read by the CC at S1; the register (in the module) is then cleared (reset) at S2. The dashed portion of the second line in Fig. 5 represents the changing of the R or W data in response to an operation timed by S2.

A Note on the Importance of the Strobes. The use of the strobes S1 and S2 in timing various parts of the cycle was described above. However, this discussion may not have fully emphasized the importance of these signals. There is a rule that irreversible actions, such as clearing registers, etc., must not be taken except at the time of the strobes. The reason is as follows.

Whenever a group of signals, such as the NAF group, is suddenly impressed on the Dataway, a certain amount of transient behavior will be observed before the signals settle down. The transients may be due to self-impedance effects--ringing and rise-time--or due to mutual impedances--cross-coupling between lines carrying different signals. The amount and duration of the transients depends in part on the mechanical construction of the Dataway (and modules). Because of these effects, during the transient period the value of the signals received by the module may be different from the values sent by the controller. The value of the NAF code, for instance, may be momentarily misinterpreted. This misinterpretation will have serious consequences of irreversible actions taken on the basis of the NAF code alone. However, if the system behaves according to the specified timing diagram in Fig. 5, no misinterpretation is possible during the times of the strobes. In this example, the NAF may be thought of as a key which is inserted in the lock (decoding logic) of the module. However, the key is turned only by a strobe and only after the key is stably defined.

Unaddressed Operations (Clear and Initialize) and Inhibit

In CAMAC, unaddressed operations refer specifically to those having a specified cycle timing: namely, Initialize and Clear. Although the Inhibit signal is not addressed, it is non-synchronous--its logic state can be changed at any instant--and is therefore not classified as an unaddressed operation.

Clear (C), Initialize (Z) and Inhibit (I) are all bussed signals, available to all plug-ins.

The commands C and Z may cause irreversible operations in modules, such as erasure of data or resetting of control flip-flops. Because of the ever-present possibility of false interpretation of a signal--because of noise or cross-coupling from other signals--it is poor practice to allow irreversible actions on the basis of one signal. Therefore modules are required to gate these commands with S2. The chance of false interpretation of C·S2 or Z·S2 is much less than of C or Z alone.

Initialize

Initialize is usually used to force the system into a passive and well-behaved condition. The specification requires module designers to insure that Z places all registers and bistables into a "defined state". This may cause erasure of data and service requests and changes in the settings of control registers. This is rather strong medicine, and intentionally so. It makes Z especially useful when power is first turned on or whenever a complete restart is required. It is similar to a Power Clear signal available on many computer I/O structures.

Clear

Clear tends to be a more benign operation than Initialize. It is often used to clear data registers in a group of modules (for instance following the transfer of a block of data) without affecting other features of the modules. There is no rule that requires data registers to be connected to Clear (C).

Note the difference between Clear (C), and the commands F(9), F(10), F(11), F(21) and F(23). The latter are all addressed to specific registers. The former is available to all modules. If you issue a Clear (C) operation, will it clear all the registers of all modules in a crate? Not necessarily, since some modules are designed not to utilize C. If you want to clear a specific register can you use F(9)? Not necessarily, since the module may not be designed that way. This is an example of a "level of standardization" consideration referred to earlier. In this case, the user must read the data sheets of the modules to determine the proper method of clearing the registers. The method by which the word "Clear (C)" is transmitted to the modules is clearly spelled out. The meaning of the one-word conversation, "Clear (C)" is subject to interpretation.

Timing of Unaddressed Operations

The timing of an unaddressed operation cycle is also specified. The minimum cycle time is 750 nanoseconds. The only signals involved are C or Z, B, and S2. The inclusion of S1 is optional. Note that the B, S1 and S2 signals are common to command and unaddressed operations. This means, of course, that only one or the other operation can be going on at any instant. Note also the absence of addressing--i.e., the N and A signals are not involved.

Inhibit

Inhibit (I) is used to control activities such as data taking within the modules that use it. Here is another case where the user must consult the data sheets of his modules in order to assess the effects of I.

The timing of Inhibit is completely independent of other operations. The state of the signal may be changed at any time. In practice Inhibit is generated by the CC, but because its timing is independent, it could conveniently be generated by appropriately designed modules.

The Status Signals--Q,L,X and B

Of these four, only Busy, B, is generated by the crate controller and is used merely to indicate that a Dataway operation is in progress. The other three are generated by the modules. **Response, Q, and Command** Accepted, X, are generated only as responses to command (addressed) operations. The timing of these signals in such operations is shown in Fig. 5. Look-at-Me, L, is initiated "voluntarily" by the module. It indicates the existence of a service request from some part of the module.

Q Response

On the Dataway, Q is a bussed line. As with the R lines, any module may have the ability to pull the Q line to the '1' state, but only the addressed module(s) is(are) permitted to do so.

The meaning of the one-bit Q signal can fall into one of three categories.

(1) The function codes F(8) and F(27) are used to test the status of certain features of the modules. The results of the test are transmitted by Q.

(2) The Q response may be used to help coordinate block transfers of data.

(3) In other command operations, the meaning of Q is unspecified. The module designer may either assign a special meaning to Q, or simply ignore it.

Q responses for F(8) and F(27). The Test Look-at-Me function, F(8), is used as its name indicates--i.e., to test the state of a Look-at-Me request signal within the module. An example of the use of this code is given below. The definition of the response is explicit. For example, a Q=0 response is required if the state of the request signal is '0'.

The Test Status function, F(27), can be used to test the state of some feature in the module. The feature to be tested is determined by the designer. The test can be steered to one of up to 16 features in a module by means of the subaddress. The meaning of a '1' or '0' response obviously depends on what is being tested. The description of the module should clearly indicate this information.

Use of Q for Transfers of Blocks of Data. In some systems, the system controller may wish to transfer a block of data (series of datawords) between its memory and CAMAC. In some cases, the module or modules involved may need to exercise a certain amount of control over the block transfer. To accommodate this, the CAMAC specifications define three modes in which the Q-responses from the module(s) control certain aspects of the transfer. These are the Scan Address, Repeat and Stop modes.

A block transfer of data is effected by a series of Read or Write commands issued by the crate controller. In the three defined modes, the module generates a Q-response based on its opinion as to whether a data word was successfully transferred during that particular command cycle. In all modes, Q=1 denotes a successful transfer, and Q=0 means it was unsuccessful.

In the following, the modes are described in terms of Read operations--i.e., the block of data moves from CAMAC to computer.

Address Scan. The Address Scan mode of Q-controlled block transfer is intended for transfers of data to or from any group of registers. In Read transfers, each register contributes one word to the block. However, the registers can be distributed among an array of modules. In each module of the array, there will be from one to 16 registers. There may be empty stations between modules of the array. Under these conditions, the transfer could be effected by having the system controller memorize the location of each register. It could then address a Read command to each occupied location. The Address Scan provides a simpler way in which the system control needs know only the first and last register locations, or else the first location and number of words in the block.

Modules and their registers intended for Address Scan must have the following properties:

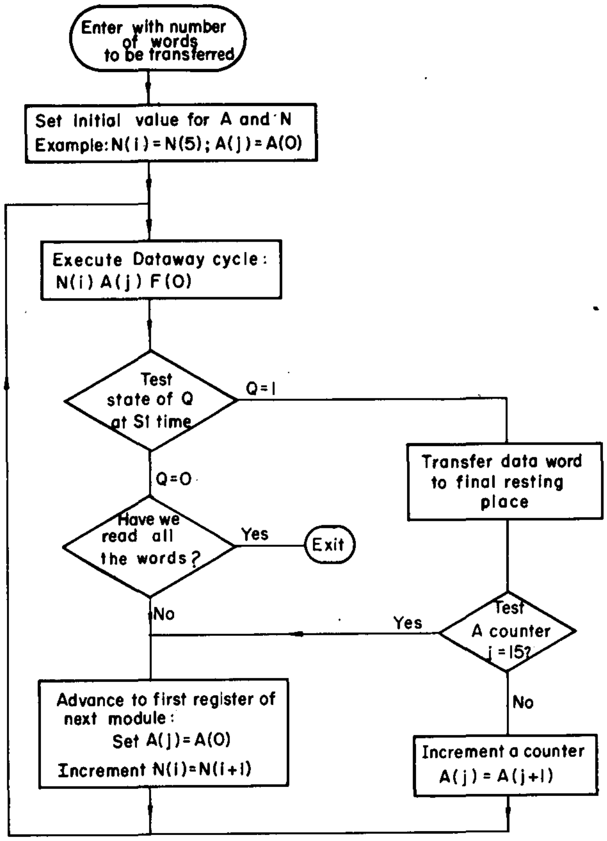
(1) the registers within a module must be located at successive subaddresses starting at A(0);

(2) for Read and Write commands the Q-responses from a module must be '1' for all subaddresses at which the registers are located, and '0' at the lowest numbered subaddress (if any) at which there is no register. (If the module has 16 registers, it gives Q='1' at all subaddresses.)

The system controller can utilize a simple address modification algorithm that is based on these characteristics. An example of such an algorithm is shown in Fig. 6. Note that there are some "wasted" Dataway cycles. If, for example, a module has less than 16 registers, it will give a Q=0 response on the cycle after the last register has been accessed. This is the signal to the controller not to bother with the rest of the subaddresses, but rather to jump to the next station. If the next station is unoccupied, the next cycle is "wasted" to find this out.

Repeat. The Repeat mode is intended to be used to move a block of data to or from a single register in a module. It can be used in situations where the register may not always be ready to effect the transfer. A register intended to be accessed by Repeat mode behaves as follows. When accessed by the Read or Write command, it responds with Q=1 if it was ready and able to complete the transfer. It responds with Q=0 if it was not ready. The algorithm used by the controller can be very simple. In Read, for example, it repeatedly issues the Read commands, but moves a data word to memory only if Q='1' on that cycle.

Stop. There are some situations in which the number of words in a block is not known by the system controller, but rather by the module. The Stop mode provides a way in which the module can pass this bit of information along to the controller. It keeps giving a Q='1' response as long as it has data to send. On the first command received after the last word has been sent, it responds with Q=0. The controller has an easy job with this one.



Enter with number of words to be transferred
Set initial value for A and N
Example: N(i) = N(5); A(j) = A(O)
Execute Dataway cycle:
N(i) A(j) F(O)
Test state of Q at S1 time
Q = 1
Q = O
Transfer data word to final resting place
Have we read all the words?
Yes
Exit
No
Yes
Test A counter j = 15?
No
Advance to first register of next module:
Set A(j) = A(O)
Increment N(i) = N(i+1)
Increment a counter
A(j) = A(j+1)

Fig. 6. An algorithm that could be used by a controller in a Scan Address block transfer. This is for a Read operation. Note the effects of the Q-response test. The "final resting place" could be a computer memory, for example.

The L Request

Situations often occur in which a module needs to notify the system control that it (the module) needs attention. The Look-at-Me (L) signal is provided for this purpose. On the Dataway, the L's are wired in point-to-point fashion--one L line goes from each station to a pin on the connector at station 25. Thus, the CC can immediately identify which module is asking for attention by scanning its 24 L inputs. It can pass this information along to the system control.

There is, of course, no standard response that the system control makes to an L request, since these requests are made for various reasons. Modules containing scalers may signal upon a scaler overflow. A digitizing module may signal when data are ready for transfer. A typewriter control module may signal when it has finished typing a character. Some modules may have several possible reasons for making a request. The system control must therefore be programmed to respond appropriately according to the particular request.

A module may initiate a Look-at-Me Request at any time, without regard* to any command cycles that may be in progress. The L request, once initiated, stays on continuously until reset. It may be reset either by a specific command, such as F(10) or as a by-product of the requested action. In Fig. 7, which illustrates some L logic, FF2 is reset either by an F(10), or by F(0)·A(1). The latter is the command that reads ADC(1), which is the action requested by L source (1).

The logic within a module that is concerned with Look-at-Me's may be simple or complex, depending on how important the L's are to that module. Figure 7 illustrates an example that is not one of the simplest.

Imagine that this module contains three ADC's (analog-to-digital converters), which are given subaddresses 0, 1, and 2. If any one of the three ADC's is called upon (by an external signal) to digitize the amplitude of an input signal, the module sets its L flag when the digitizing is done. The L request notifies system control that a piece of data is now available. However, since the L is only a single bit of information, it cannot indicate which ADC has generated the data. The module may provide for the identification in several ways. Two methods are shown in Fig. 7.**

In the first method, the accessing of features associated with one of the three L sources is directed by a subaddress associated with the appropriate command. For example, the features associated with L source (1) are accessed by A(1) in Fig. 7.

Before seeing how the identification is made using the first method, let us momentarily step back in time. At some point, system control (a computer, perhaps) has "initialized" the L logic in the module, using the procedure in Table III.

The module is now ready for use. Let's say that ADC number 1 gets a pulse to analyze first. When finished with the analysis, it generates an "ADC(1) Data ready" signal, setting flip-flop FF2. After a slight delay, as the signal propagates through two AND and one OR gates (see Fig. 7), L goes to the 1 state. Since the module is in station i, L = L(i). System control recognizes L(i). In order to identify the number of the ADC that has overflowed, it follows the steps in Table IV.

*Well, almost without regard. It is required that L signals be gated off the Dataway during any command operation which would result in the L being reset. One simple (and legal) way of complying with this rule is to gate L with N, as is shown in Fig. 7.

**To facilitate the explanation, Fig. 7 contains a mixture of two methods of controlling L features. These two might be called "Operations on LAMs via Subaddress," and "Operations on LAMs via Databits" (second method in Fig. 7). Normally, a module with several L sources will use one or the other of these methods. Section K5.4.1a of the reference[8] shows complete examples of each of these. The specification[1] (Sec. 5.4.1.2) recommends subaddresses in Group 2 for registers of the second method. Modules with only one L source will usually use the subaddress method.

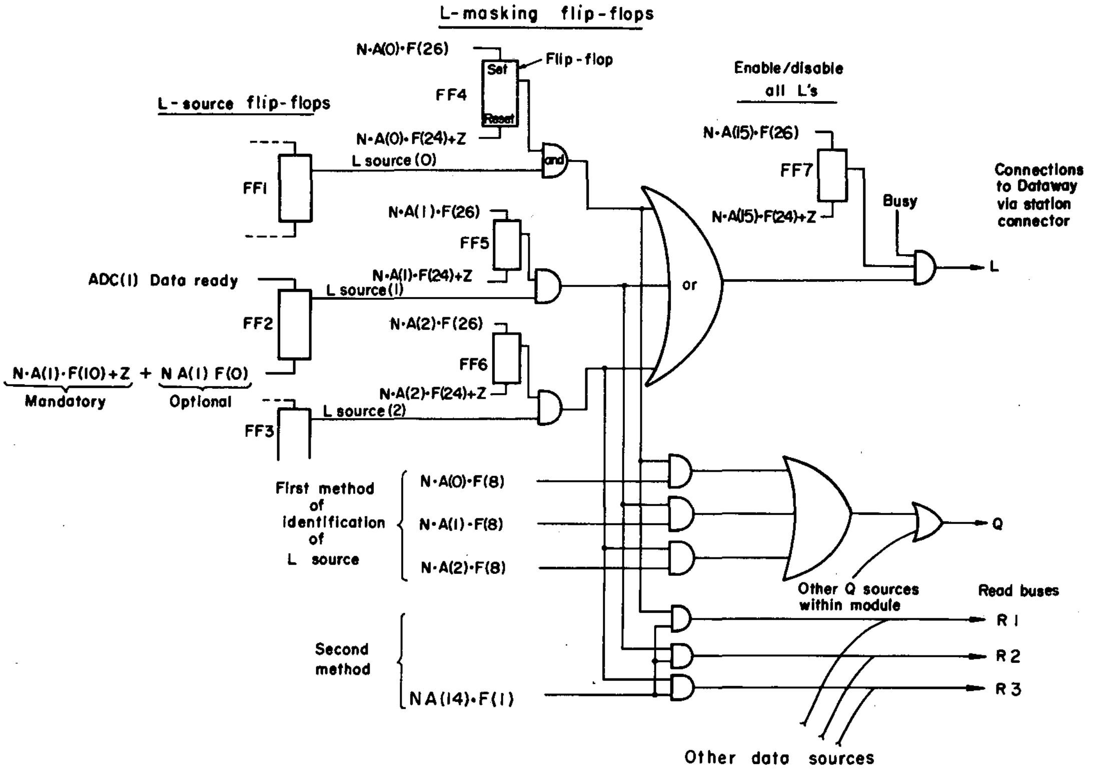


Fig. 7. An example of part of the logic within a module that is associated with the generation of an L signal, and with the identification of the part of the module from which it originated. Operations such as setting or clearing a flip-flop should be done only in conjunction with S1 or S2. To simplify the drawing, this has not been shown.

TABLE III Steps used to initialize the L logic of Fig. 7.

Step	Command	Meaning	Comments
1	— —	— —	Turns on power to system
2.	Z	(Initialize)	Resets flip-flops FF1 through FF7 (also resets ADC data registers)
3	N(i) A(0) F(26)	[Enable L source (0)]	Sets FF4
4	N(i) A(1) F(26)	[Enable L source (1)]	Sets FF5
5	N(i) A(2) F(26)	[Enable L source (2)]	Sets FF6
6	N(i) A(15) F(26)	[Enable all L's]	Sets FF7

TABLE IV Steps used in identifying the source of the Look-at-Me using first method of Fig. 7.

Step	Command	Meaning	Comments
1	N(i) A(0) F(8)	[Test L-source (0)]	During this cycle Q remains in the 0 state
2	N(i) A(1) F(8)	[Test L-source (1)]	During the cycle Q = 1 (aha!)
	Q = 1 is the clue. ADC(1) is the guilty party. Naturally System control does not bother to do Step 3.		
3	N(i) A(2) F(8)	[Test L-source (2)]	This step omitted
The next step is to read the diqital data generated by ADC (1)			
4	N(i) A(1) F(0)	Read ADC (1)	

When our module was designed it was decided that the purpose of signaling via L would be to request that data be read out. Thus, we are justified, according to the specifications,[1] in designing the module in such a way that FF2 is automatically reset by Step 4. L returns to '0' unless another of the ADC's has digitized in the meantime.

The identification process described above takes a maximum of N Dataway cycles if the module has N sources of Look-at-Me. If N is a large number, it may be preferable to use the second method of identification shown at the bottom of Fig. 7. In this method, a single command N(i) A(14) F(1) is used to read a word onto the R bus. In this word a bit is assigned to each L source within the module. The state of up to 24 L sources can therefore be identified in one command cycle.

Command Accepted--X

A quite reasonable question for a system controller to ask is: "Did the command I just issued actually reach a module, and did the module actually execute the command?" In some systems, it may be important to have the answer immediately so that remedial steps can be taken if necessary. The Command Accepted signal is the means by which the module can answer the question. Whenever a module receives a command, it must respond with X=1 if it is able to execute the command. If it cannot, it should generate an X=0. It may not be able to execute because its power supply is turned off, or because the command is one it was never designed to execute, or because it is still busy doing something else. If the execution requires the cooperation of some external device, then the module may reply with X=0 if the external device is inoperative. The system controller may also get an X=0 if the command did not reach a module--e.g., because of software error, it addressed an empty station.* (Note that during Address Scan transfers, it is a normal operation to address an empty station.)

A Note on Selective Set and Selective Clear

The Selective Set and Selective Clear function codes [F(18), F(19), F(21) and F(23)] were created originally to aid in the manipulation and control of modules having multiple sources of Look-at-Me. The "second method" in Fig. 7 showed an example of an operation in which the state of up to 24 L sources can be read in one Dataway cycle. This is done by reading the state of one source onto each R line. However, Fig. 7 omitted to show an example of how up to 24 flip-flops--such as the L-masking flip-flops--can be set or cleared in one cycle. This can be done by using the Selective Set and Clear codes. Note that these are classified as Write commands. Therefore a word of data is sent on the W lines with these commands. Each W bit in the data word is associated with a particular flip-flop. If the command is a Selective Set, then a W bit=1 will set the associated flip-flop; a W=0 will not change its state. In Selective Clear, a W=1 clears, W=0 does nothing.

In effect, the W bits may be considered as another level of addressing. If the 24 flip-flops are considered to constitute a register, then the register may be assigned a subaddress. If a Selective Set command is directed to that register by its assigned subaddress, then the setting action can be steered to individual flip-flops of that register by the W bits. Thus, one can individually set or clear up to 768 individual flip-flops in a module. (16 subaddresses x 24 W bits x 2 groups). This represents a very powerful addressing capability for on-off control applications.

*Sections K5.4.4b, L4.2.3 and Appendix F of the reference[8] discuss further aspects of X, including "normal" X=0 responses during Address Scan, and the lack of X response on older modules.

References

1. "CAMAC - A Modular Instrumentation System for Data Handling - Revised Description and Specification." Identical versions of this document have been issued by the U.S. AEC NIM Committee and by the European ESONE Committee. They are: U.S. AEC Report TID-25875, July 1972; and Euratom Report EUR-4100e, 1972.

2. L. Costrell, "CAMAC Instrumentation System - Introduction and General Description." (See note)

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

4. F. A. Kirsten, "A Short Description of the CAMAC Branch Highway." LBL-1576, February 1973 (See note)

5. S. Dhawan, "CAMAC Crate Controller Type A." (See note)

6. F. A. Kirsten, "Some Characteristics of Interfaces Between CAMAC and Small Computers." LBL-1577, February 1973. (See note)

7. CAMAC Tutorial Issue, IEEE Transactions on Nuclear Science, Vol NS-18, No. 2, April 1971.

8. U.S. AEC Report TID-25877, "Supplementary Information on CAMAC Instrumentation System."

9. R.C.M. Barnes, "The Revised CAMAC Specification EUR-4100e (1972)," CAMAC Bulletin, Issue No. 4 July 1972, pp.6-8.

10. **F. A. Kirsten, CAMAC Specification," LBL-1522 (Also to be published in February 1973 issue of IEEE Transactions on Nuclear Science.)**

11. "CAMAC - Organization of Multi-Crate System, Specification of the Branch Highway and CAMAC Crate Controller Type A. Identical versions of this document have been issued by the U.S. AEC NIM Committee, and by the European ESONE Committee. They are: U.S. AEC Report TID-25876, March 1972; and Euratom Report EUR-4600e, 1972.

12. I. Iselin, B. Lofstedt, and P. Ponting, "CAMAC Dataway - Branch Highway Timing Relationship via the Crate Controller," CAMAC Bulletin, Issue No. 3, March 1972, pp.7-8.

Note: References 2-6 will be published in the April 1973 issue of the IEEE Transactions on Nuclear Science. This issue will be called "CAMAC Tutorial Issue."

LEGAL NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or 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.

TECHNICAL INFORMATION DIVISION
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720