Submitted to IEEE Transactions
on Nuclear Science



UCRL-20214
Preprint



# OPERATIONAL CHARACTERISTICS OF THE CAMAC DATAWAY

Frederick A. Kirsten

December 10, 197[illegible]

AEC Contract No. W-7405-eng-48



LAWRENCE RADIATION LABORATORY
UNIVERSITY of CALIFORNIA BERKELEY

## DISCLAIMER

UNIVERSITY OF CALIFORNIA

Lawrence Radiation Laboratory
Berkeley, California

June 25, 1971

# ERRATA

TO: All recipients of UCRL-20214

FROM: Technical Information Division

SUBJECT: UCRL-20214: "Operational Characteristics of the CAMAC Dataway" by Frederick A. Kirsten
December 10, 1970

Please make the following corrections on subject report:

Page 1, 1st column, 3rd para., line 5: "of" replaces "ot"

Page 1, 1st column, 4th para., line 2: "The" replaces "Ths"

Page 3, Fig. 2: The bar marked "Data" should have arrow heads at both ends

Page 4, 1st column, 2nd para., line 3: "to represent" replaces "to re represent"

Page 4, 1st column, 3rd para., line 1: "whereas A and F" replaces "whereas A and N"

Page 4, 2nd column, 3rd para., line 5: "F(18)" replaces "F(17)"

Page 5, 2nd column, 1st para., line 10: "F(26)" replaces "F(2)"

Page 6, 1st column, 6th para., line 6: "at any time" replaces "at any time the"

Page 8, Fig. 5: a) Reset to FF4 should be: $N \cdot A(0) \cdot F(24) + Z$

b) Flip-flop on right should be labelled FF7 instead of FF6

Page 9: Top line of table should read: "During this cycle $\underline{Q}$ remains in the 0 state."

8/24/71

# OPERATIONAL CHARACTERISTICS OF THE CAMAC DATAWAY

Frederick A. Kirsten
Lawrence Radiation Laboratory
Berkeley, California

December 10, 1970

## Summary

The basic purpose of CAMAC is to provide a standardized method for transmitting 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 in the transfer of digital information. The CAMAC specification[1] contains the rules for both of these aspects of CAMAC. This paper describes the features and uses of the control language in a less formal way than must be used in the specification. Examples are given of the interplay of control and synchronization signals between the modules concerned.

Other papers in this series[2-4] 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 paper is designed to supplement the CAMAC specification. The reader is advised to obtain a copy of that document, and to read it first. The reader's attention is also called to other references [5,6] containing informal descriptions ot the CAMAC system.

## Introduction

### Levels of Standardization

The value of any standard depends on how faithfully it is followed. Ths interchangeability of CAMAC modules and designs is one of the most important goals of this standardization effort. 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 of commercially available components. The time required to design a new module can be reduced if a consistent 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. The system uses this language to carry on a conversation. The conversation may be simplified if there is consistency in usage of certain terms. The simplification can result in significant savings in hardware and software used to control the systems. The consistency has, as "positive" ingredients, common usage and common sense; and as a "negative" ingredient, the compromising and modification of the basic standard.

### Some Definitions

It may be helpful to begin the description of Dataway operations by (re)defining some commonly used terms.

Crate. The standard CAMAC crate is a 19-inch rack-mounted structure that has 25 stations (slots) for accommodating standard plug-in modules, and contains the wiring for interconnecting the modules (Dataway). The stations are numbered from 1 through 25 (left to right, viewed from the front). Each has an 86-pin card-edge connector wired to the Dataway.

In contrast to the NIM system, in which the bin wiring is used mainly for carrying power to the modules and any data transfers are via extra connectors, the CAMAC Dataway wiring carries both data and control signals as well as power.

Dataway. The Dataway is the wiring that interconnects the 25 card-edge connectors. It includes signal wires and power wires. Most of the wiring on connectors 1 through 24 is bussed--i.e., the same signal is available at the same pin on all those connectors. The wiring for station (N) and Look-at-Me (L) signals is point-to-point, fanning out from the connector at station 25.

The Dataway wiring may be divided into three categories: data transfer, control, and power. For data transfer, two 24-bit parallel unidirectional data buses (highways) are provided. One is for data transfers from Crate Controller to module. The other is for module to Crate Controller transfers.

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 plugs in to at least two Dataway sockets (e.g., sockets 24 and 25). 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" 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 intself. 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[7] for the CC that interfaces the crate to the CAMAC Branch Highway. This is known as the Crate Controller Type A.

Module. In this paper, module is used to mean a plug-in other than the Crate Controller. In usual circumstances, modules are slaves, answering (with one exception, the Look-at-Me signal) only to the master, 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, digitizing signals, controlling external devices, etc. The other (part B) is the part that "speaks CAMAC." Its job is to facilitate the conversation between part A of itself and the "brains" of the system, be it computer or other form 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 that 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 many cases, the primary system control is the program (software) of a computer. In this case, the program decides which are the necessary commands, and orders some piece of hardware to carry them out. In other systems, the control is purely 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. In the simplest situation, the Crate Controller itself is in sole control; in others, the Crate Controller is a tool of the system control.

## Elements of Communication on the Dataway

Table I lists the signals available at the 86-pin connectors of all stations except station 25. It also gives the purpose of each signal. As shown in Fig. 1, the Dataway wiring is a combination of bussed and point-to-point connections. All signal wires, except the N and L lines, are bussed--i.e., each signal appears at each of stations 1 through 24. A separate

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

| Title | Designation | Use in module |
|---|---|---|
| **Commands, addressed** | | |
| Function codes | F1,2,4,8,16 | Binary coded. Defines the functions to be performed in module. There are 32 possible function codes. |
| **Commands, unaddressed** | | |
| Initialize | Z | Sets module to a defined state. |
| Inhibit | I | Disables features for duration of signal. |
| Clear | C | Clears registers. |
| **Addressing** | | |
| Station number | N | Selects the module. There is an individual line from Crate Controller to each station. |
| Subaddress | A1,2,4,8 | 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 | S1 | A strobe for timing the first phase of an operation. |
| Strobe 2 | S2 | For timing second phase 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. |
| Response | Q | A reply by module to certain commands issued by Crate Controller. |
| 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.

N and a separate L line are wired from each of stations 1—24 to station 25. The unique feature of station 25 is that it has access to all 24 N lines (one for each of stations 1 through 24), and all 24 L lines.

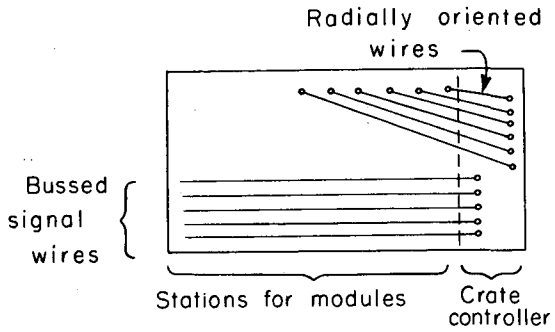


Fig. 1 A sketch illustrating a small part of the Dataway wiring. Both point-to-point (radially oriented) and bussed signal wires are shown. Bussed signals are available at all stations; point-to-point signals travel between specific stations and the Crate Controller.

This configuration of N and L wires has two important consequences:

(a) A module is addressed by the number of the station it is plugged into.

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

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 bus, which provides for a single bit of status data, flowing from module to CC.

However, the control provisions are highly asymmetric. Nearly all facilities are for commands from the CC to the modules. The only signal for control originating in the module is the L (Look-at-Me). 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 16-bit computer words.)

## Dataway Operations

The family of Dataway operations encompasses all the "things" that can be accomplished via the Dataway.

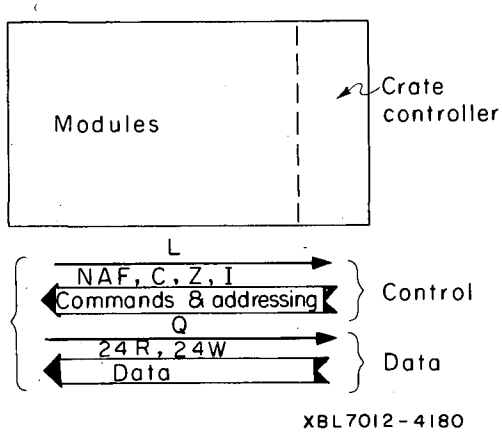


Fig. 2 This sketch summarizes the capability of the Dataway transmitting data and controlling signals from Crate Controller to module and vice versa.

The members of the family fall into different groupings according to which characteristic is under discussion. A study of Table II will bring this out. The significance of the names and the characteristics are discussed below. Let us note in passing that the group labeled "NAF" is the major category, and is largely the reason for CAMAC's existence. There are more than 12000 different NAF operations that can be done in a fully loaded crate.

### Commands

Most Dataway operations are commands issued by the Crate Controller. Addressed commands are of the form NAF, and are directed to a specific location. Unaddressed commands (C, Z, and I) are sent out to any module willing and able to obey them. All addressed commands obey the standard Dataway cycle timing and in this paper are therefore categorized as "cyclic." Of the unaddressed commands, C and Z follow a modified form of the cycle timing and are called (in this paper) paracyclic. The command I (and also the L request) have no specific timing rules and are called acyclic.

Addressed commands. An addressed command consists of the codes (numbers) for each of the signals N, A, and F. N and A together denote the address of a specific register or piece of logic within a module --N is a station number in the crate, and A is a subaddress within a module. The N signal is received only by the addressed station; the A signals are bussed and therefore are available to every module. The binary-coded value of the subaddress is carried on four Dataway wires designated A1, A2, A4, and A8. There are 16 possible subaddresses.

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, $1 < i < 23$;

TABLE II. Classification of CAMAC Dataway operations

Class of operation

| | N A F | C | Z | I | L |
|---|---|---|---|---|---|
| Type of operation | Command → | | | | Service request |
| Addressing | Addressed | Unaddressed → | | | |
| Timing | Cyclic | Para-cyclic → | | ← Acyclic | |
| Originating device | Crate Controller (CC) → | | | CC or module | Module |

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

F(k) = the operation to be executed by the logical device at subaddress A(j) of the module in station N(i), $0 < k < 31$.

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

There is a possible confusion in symbology. This may arise because of the use of both ordinal and binary codes. The form F(k) will be used to re represent the ordinal value (e.g., one out of 32) 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 an "Increment Preselected Registers" cycle, we ask the CC to execute function code F(25). To do so, the CC sends out the binary code for F(25) which has F1 = F8 = F16 = logic 1, and F0 = F2 = F4 = logic 0.

Note that, whereas A and N are binary coded on the Dataway, N is carried in an ordinal code, with a separate wire for each value of N--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 Increment, where it may be desired to have the operations carried out simultaneously in a number of modules. Both A and F are carried on the Dataway in a binary form, although in writing the values or consulting the function code table we prefer ordinal representations.

It may also 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 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 ≡ N(10).

Function codes. A complete list of the 32 function codes is shown in Table III, reproduced from the specification.[1] Half of the codes have been assigned specific meanings. The others are either reserved (may not be used until assigned specific meaning by the CAMAC specification), or are usable for nonstandard functions. Additional details of the standard functions are given in the specification.

Note that the codes may be categorized into three groups:

(a) One group transfers data from modules to CC via the R lines. These are characterized by F16 = F8 = 0.

(b) Another group transfers data from CC to module via the W lines. These all have F16 = 1, F8 = 0.

(c) The third group is not involved with data transfers. These 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. Some tend to think of them as representing data and control registers, respectively. Others think of Group 2 as permitting an expansion of the number of addressable Read/Write registers to 32. 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 are transferred into the addressed register. This is sometimes called the "jam transfer" operation. Selective overwrite means that certain bits of the register are written into as determined by a preloaded mask register, contained in the module.

Note carefully that there is absolutely no intention to force 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(17) (Selective Overwrite 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.

Transmission of commands on Dataway. Figure 3 depicts the flow of an NAF command from CC to module. The individual components of the command are held in the N, A, and F registers. In this example, the registers are all assumed to be binary. At some point in time, the command exists only in the "mind" of the system controller. In response to a whim--ours not to

TABLE III. 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 | Additional functions that use the R lines | 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 | Additional functions | 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 Overwrite Group 1 Register | | 1 | 0 | 0 | 1 | 0 |
| 19 | Selective Overwrite Group 2 Register | | 1 | 0 | 0 | 1 | 1 |
| 20 | Nonstandard | Additional functions that use the W lines | 1 | 0 | 1 | 0 | 0 |
| 21 | Reserved | | 1 | 0 | 1 | 0 | 1 |
| 22 | Nonstandard | | 1 | 0 | 1 | 1 | 0 |
| 23 | Reserved | | 1 | 0 | 1 | 1 | 1 |
| 24 | Disable | These functions do not use the R or W lines | 1 | 1 | 0 | 0 | 0 |
| 25 | Increment Preselected Registers | | 1 | 1 | 0 | 0 | 1 |
| 26 | Enable | | 1 | 1 | 0 | 1 | 0 |
| 27 | Test Status | | 1 | 1 | 0 | 1 | 1 |
| 28 | Nonstandard | Additional functions | 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 |

reason why--the control causes the registers to be loaded with the binary values for N, A, and F. The wiring for this loading is not shown.

Let us assume that the decimal values for N, A, and F are 1, 4, and 16 respectively. The number for N is converted from a binary code to a one-out-of-24 code by the N decoder. Therefore; 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 NA(4) F(16)--Write into the register at subaddress 4 of this module. Note the uses of binary and ordinal codes.

Typical modules are quite specialized, and therefore are wired to react to only a few of the many possible CAMAC commands. In system operation, a module may receive, either accidentally or intentionally, a command to which it is not expected to respond. For this reason, it is highly recommended that function codes be fully decoded--i.e.,that all five elements of the binary code for F are used in the decoding of commands. In order that the module (see Fig. 3) respond only to function code 16, it is necessary that the signal F(16) becomes 1 if and only if F1 = F2 = F4 = F8 = 0 and F16 = 1. This condition is satisfied only if all five of the signals F1, F2···F16 are used by the F decoder.

It should be obvious that the signal N must be included as part of the command recognition. It is not necessary that a subaddress be used if omission of the subaddress does not result in ambiguities. Thus, a minimum form of the command is N(i)·F(k) [but never A(j)·F(k) or F(k)].

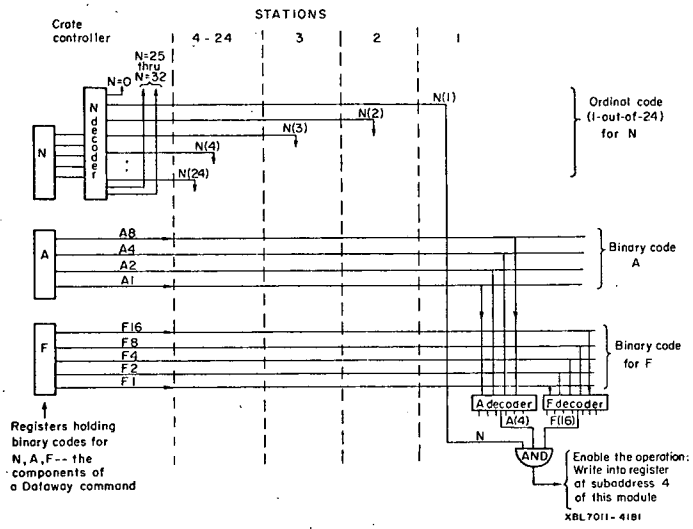


Fig. 3 Showing how an addressed command is transmitted over the Dataway, and how it is interpreted in the module.

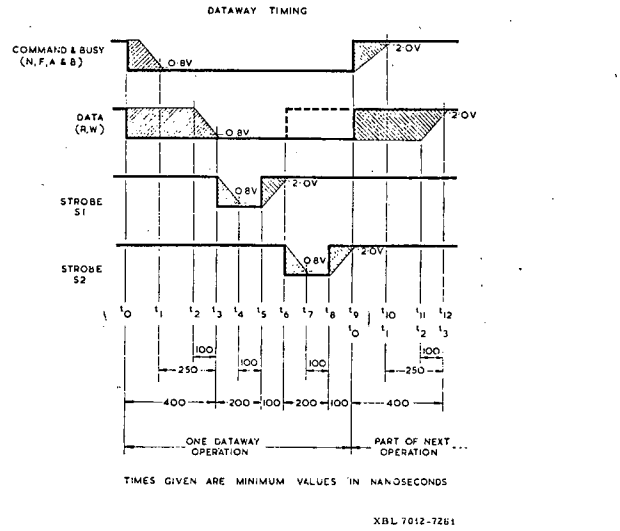


Fig. 4 Timing of a Dataway operation.

Unaddressed commands--C, Z, and I. The three unaddressed commands are C (Clear), Z (Initialize), and I (Inhibit). The signals for these commands are each carried on one bussed line, available at all stations. Issuing the command consists of setting the appropriate line to the logic 1 state.

The commands C and Z may cause irreversible operations in the modules, such as erasure of data or resetting control flip-flops. To provide additional safety, therefore, C and Z are required to be accompanied by the strobe S2. The modules must gate these commands with S2 to insure against false responses. Since both commands need S2, they can be generated only by the Crate Controller. The duration of the commands is decided by the CC, but need not be the same as for a standard Dataway cycle (paracyclic).

Initialize is used to force the system into a defined and operable state when power is first turned on, or whenever a complete restart is required. It may cause erasure of data and may force control bistables into a reset state. The specifications require module designers to insure that Z places all registers into a "defined state." This is a rather ambiguous statement. The intent is to insure that Z places the system into a completely passive and inactive condition--a condition that enables it to respond properly to following Dataway commands.

The specifications give the module designer freedom in how he uses Clear (C) in his design. Common usage tends to limit it to clearing data registers.

Inhibit (I) is used to control activities such as data taking within the modules that use it. The module designer has freedom in choosing which activities, if any, are affected by Inhibit.

In practice, Inhibit will usually be generated by the CC, although in theory any module may be designed to generate it. This is because of a unique property of the Inhibit command. Its timing is completely independent of other Dataway operations. The I command may be issued or removed at any time the, the duration of the signal being purely at the "pleasure" of the module or controller that generates it.

The use of Inhibit and Clear in a system may be a bit tricky, since they are unaddressed, i.e., applied simultaneously to all modules in a crate. Thus, any module in the crate that uses either command in a way that is undesired in the particular system will lead to difficulties in applying them. It is quite possible, of course, to design into a module a bistable that controls acceptance of these unaddressed commands. The bistable can be set or cleared with function codes F(2) Enable and F(24) Disable. Provision for testing the state of the bistable can be implemented by using F(27) Test Status.

Timing

In Table II the timing characteristics of Dataway operations are categorized into three groups. Within this paper, the groups are named cyclic, paracyclic, and acyclic. The subject of timing is discussed in more detail by Larsen.[3]

Cyclic operations--NAF. Cyclic operations are those following the Dataway cycle timing shown in Fig. 4. The solid lines in Fig. 4 show the nominal times at which signal transitions occur. The cross-hatched areas indicate the limits of initiation and completion of the transitions. The values of 2.0 V and 0.8 V correspond to the limits of signal voltages that an input must accept--$V_{IH}$ and $V_{IL}$ in IC terminology.

The timing of the cycle is fixed by the CC. It determines the timing of the salient points of the cycle, which are:

(a) The initiation of the cycle--the time at which the codes for NAF and the Busy signal are placed on the Dataway:

(b) the timing of the strobes S1 and S2: and

(c) the termination of the cycle--the time at which NAF and B are removed.

Note that the timing is completely under the control of the CC. The module is completely slave. There is no "hand-shaking" between CC and module such as exists on the Branch Highway.[7] The module has no

way to interrupt or delay completion of a cycle. (This means a module must be designed to have an "access time" of 250 nsec or less for data transfers--i.e., the addressed register must be connected to the R or W bus within 250 nsec after the command has been established.)

The shortest cycle is 1 μsec in duration. Longer cycles are permissible. The strobe signals, S1 and S2, are generated by the CC at the appropriate times during the cycle. Both signals are bussed, on two separate lines. They may be used in the modules for timing parts of Dataway operations. S1 may be used to control operations that do not change the state of the signals on any Dataway lines. For example, it may be used to strobe data from the R or the W lines into registers. As shown in Fig. 4, the cycle timing relationships are such that the R and W lines must have reached their final states before S1 begins. S2 may be used for operations that result in changes of states of Dataway signals. For example, it can be used to clear a register connected to the R lines [cf. function code F(2)]. (If a register is not connected to the Dataway, it may be cleared or incremented at S1.)

Paracyclic operations. Initialize and Clear commands are classified as paracyclic. The duration of these cycles is fixed by the CC, but is not necessarily the same duration as the standard Dataway timing cycle. Both commands are accompanied by S2 delayed from the initiation of the command signals by a suitable time, such as 700 nsec (cf. Fig. 4).

Acyclic operations. Signals that are allowed to change state at any time are the Inhibit (I) and the Look-at-Me's (L's) (provided Busy (B) = '0'). The maximum or minimum length of these signals is not defined. Other Dataway operations may be concurrent with acyclic operations.

## The Q and L Signals

The Q and L signals have a more detailed interaction with the system control than other CAMAC signals. Both are status signals generated by the modules, and are used to give indications regarding certain conditions in the module. They behave differently. The Q signal is always given (by the module) in response to a Dataway operation ordered by the CC. The L signal is initiated "voluntarily" by the module. It is the only way a module can voluntarily request service from the system control, via the CC.

## 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 logic 1 state, but only the module(s) addressed during the current operation is (are) permitted to do so. The Q response signal is interpreted by the CC (or system control) in one of two ways, according to the current function code:

(a) Response to a valid Read or Write Command.

If the current function code represents a Read operation [function codes F(0) through F(7)] or a Write operation [codes F(16) through F(23], the addressed module must put a logic 1 on the Q bus if it can obey the command. The CC can therefore examine the state of the Q bus to see if commands of this category are being fulfilled. For example, if the CC has issued a Read command, it knows the data it has just read from the R lines are OK if it sees Q = 1. If it sees Q = 0, it must assume the data are invalid. A Q = 0 response to a command N(i) A(j) F(k), where $0 \leq k \leq 7$ or $16 \leq k \leq 23$, will result if:

(i) no module is present in station N(i); or

(ii) a module is present, but is not wired to perform the function F(k); or

(iii) it can perform an F(k), but has no subaddress A(j).

The specifications require that subaddresses used with Read or Write function codes be assigned in sequence starting with subaddress A(0). This has useful properties in connection with "pseudosequential addressing" as described below.

(b) As a status response to a non-Read/Write command.

There are certain Dataway commands that ask a question of the module to which they are addressed. The response to the question is given on the Q bus. For some commands [F(8) and F(27)], the module is required to give an appropriate Q response. For other commands, A Q response is optional.

An example of logic in a module associated with the Q response for function code F(8) (Test Look-at-Me) is shown in Fig. 5. An explanation is given in conjunction with a description of the L signal.

## 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 radial 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 be programmed to respond appropriately according to which module makes the request. If there is only one possible reason for a request from the module, system control can service it immediately. If there is more than one reason, the system must make further investigation to determine the exact meaning of the request.

Let us consider a specific example: A certain 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, but is unable to indicate which ADC has generated the data. The module may provide for the indication in several ways. Two ways are shown in Fig. 5, which shows part of the logic of

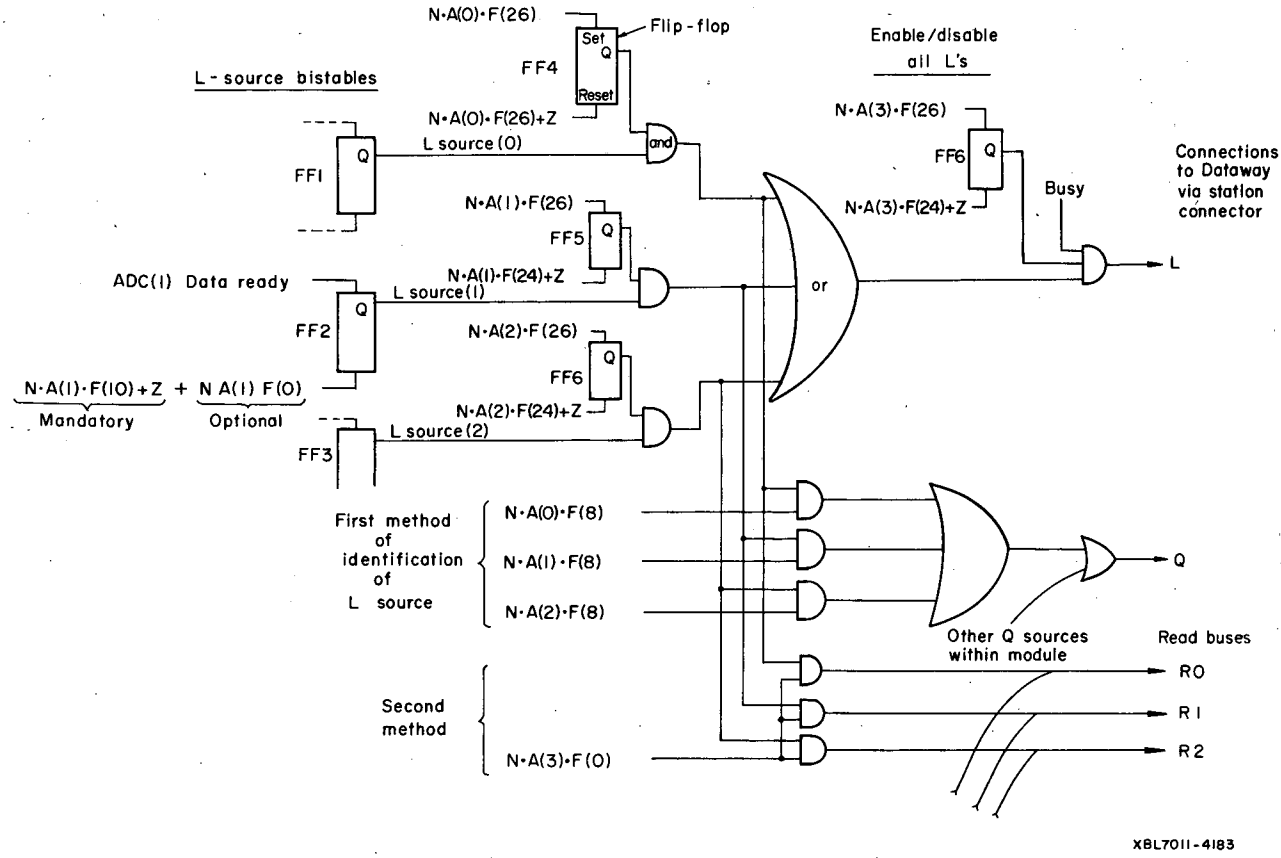


Fig. 5 An example of the logic within a module that is associated with the generation of an L signal, and with the identification of the source of the signal.

the module. [Both of these ways (and there may be others) are "CAMAC-Compatible"--i.e., they both use the standard CAMAC "language." However, Fig. 5 illustrates two "conversations" that can take place in the language. This is an example of the difference between Dataway compatibility and system compatibility--the former is concerned with the language; the latter with the conversations.]

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

The ADC's are now ready for use. Let's say that ADC number 1 gets a pulse to analyze first, setting flip-flops FF2. After a slight delay, as the signal propagates through two AND and one OR gates (see Fig. 5) 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 enters the algorithm shown in Fig. 6. The following Dataway commands which interact with the "First Method of Identification of L source" shown in Fig. 5, will result:

| 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(3) F(26) | [Enable all L's] | Sets FF7 |

| Step | Command | Meaning | Comments |
|---|---|---|---|
| 1 | N(i) A(0) F(8) | [Test L-source (0)] | During this cycle 0 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 digital 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 FF1 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. 5. In this method, a single command N(i) A(15) F(0)--subaddress 15 is arbitrarily chosen--is used to read a word onto the R bus. In this word a bit is assigned to each L source. Up to 24 L sources can be identified in one cycle.

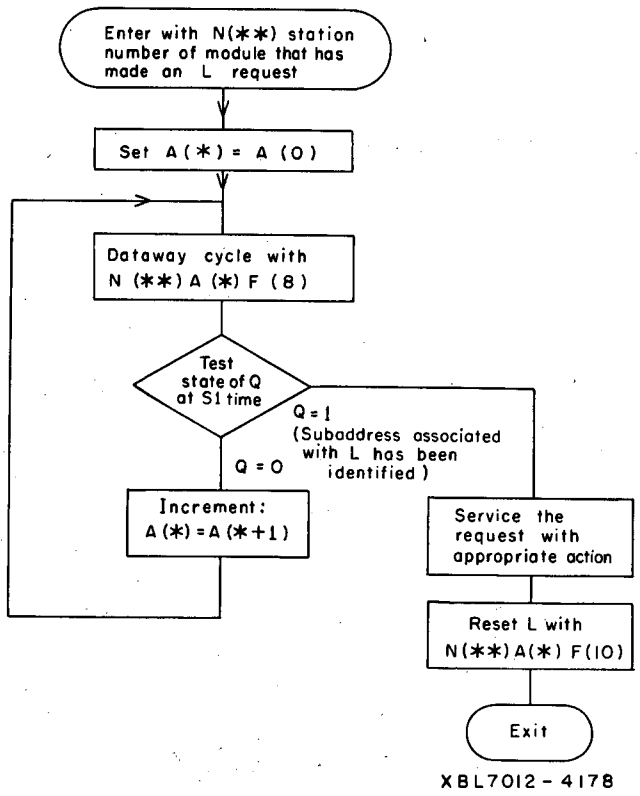


Fig. 6 An algorithm that can be used to identify which part of a module has generated an L request

## Sequential Addressing

Before CAMAC was introduced, a number of laboratories and companies had designed simple standard readout schemes using sequential addressing. In sequential addressing, data words are always read from successive hardware locations. Usually a single bus line carries an "Advance" pulse that causes the words to be placed on the data buses in sequence. Very simple hardware controllers are used.

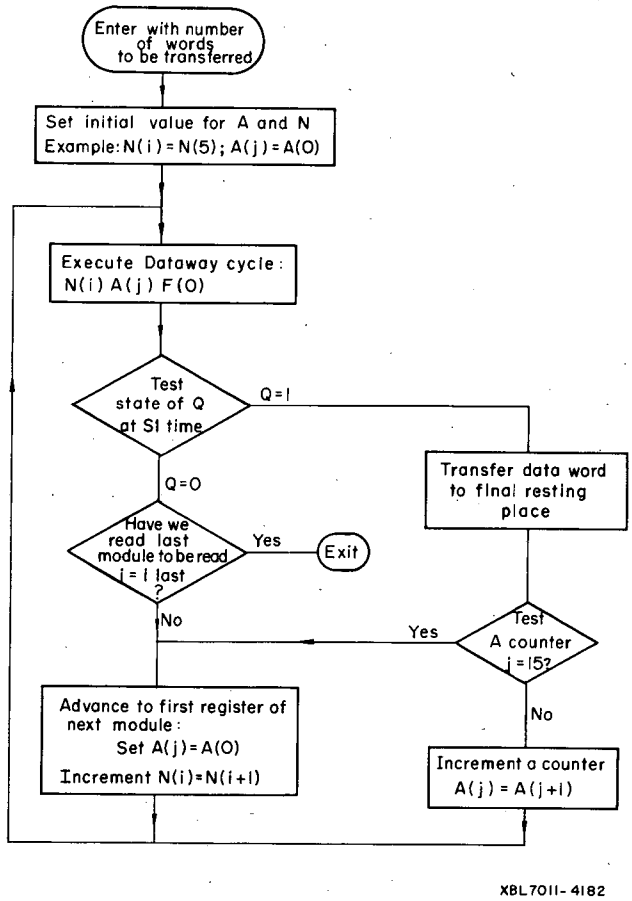


Fig. 7 This algorithm illustrates the way in which the Q response is used in the "pseudo-sequential" addressing.

CAMAC contains a provision for simulating sequential addressing for Read or Write operations. It uses one of the modes of Q response. As mentioned above, the subaddress of data registers associated with function codes F(0) through F(7) (Read functions) and F(16) through F(23) (Write functions) must be assigned in sequence, starting with subaddress A(0). During the cycles in which these data registers are read from or written into, the module must generate a response Q = 1. At the first subaddress at which there is no data register, the module must generate a response Q = 0. This Q = 0 response is a clue to the

system control indicating that all the data registers in that module have been read out. The system control therefore increments the N portion of the address, and resets the A portion to 0, in order to read data from the first register of the module in the next higher-numbered station.

Figure 7 shows an algorithm that can be used to read from a block of modules. Given a starting address--i.e., the address in the crate of the first data register to be read--the algorithm causes each successive data register in each successive module to be read.

This algorithm can, of course, be implemented in the software of a controlling computer. However, it is particularly intended to permit simple hardware control of the block transfer of data from a series of modules. In CAMAC literature, the word "autonomous" is often used in this connection, meaning that the control mechanism is self-contained (in hardware).

Although the sequential addressing scheme can work for either reading or writing, it seems obvious that it will be applied mainly to reading of data. Again, all registers involved must be sequentially located in the crate. The scheme tends to make the module boundaries "transparent." Controllers built for the Branch Highway will often incorporate an extended sequential addressing in which crate boundaries are also transparent.

References


1. "CAMAC, A Modular Instrumentation System for Data Handling, Description and Specification," EURATOM Report EUR 4100e, March 1969. Note that this has been reprinted in the U. S. (August 1970) with the NIM Committee comments and endorsement. Requests for information in the U. S. should be addressed to Louis Costrell, Chairman, AEC Committee on Nuclear Instrument Modules, Radiation Physics Building, National Bureau of Standards, Washington, D. C. 20234

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." (See Note)

5. R. C. M. Barnes and I. N. Hooton, "The CAMAC System of Modular Instrumentation" IEEE Trans. Nucl. Sci. NS-16, Oct. 1969, pp 76-80.

6. I. N. Hooton and R.C.M. Barnes, A Standardized Dataway Highway for On-Line Computer Applications", Proc. Fall Joint Computer Conferences, San Francisco, California, Dec. 1968, pp 1077-1088. (In this paper, the Dataway system is called IANUS. The name was later changed to CAMAC.)

7. "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 Ref. 1.


NOTE: References 2-4 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.

Reference 4 will also be printed as UCRL-20217.

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