Model 3972

Interconnect CAMAC Crate Controller

Instruction Manual

December, 1995

CONTENTS

FIGURES

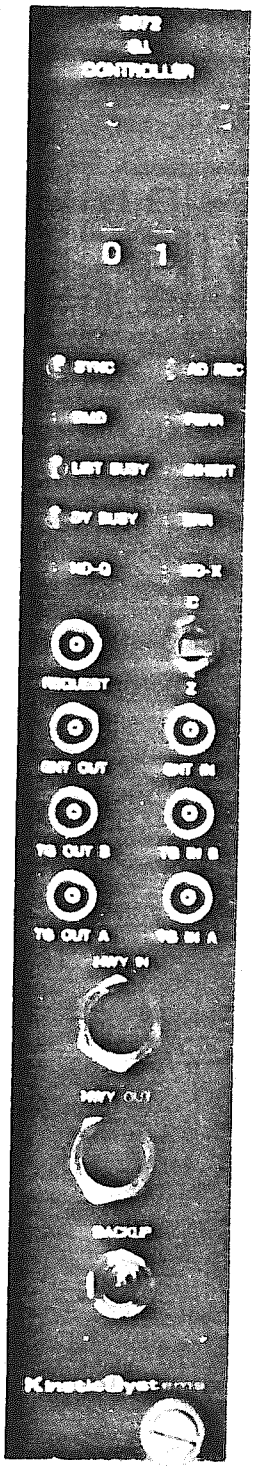
Grand Interconnect Crate Controller

Part of the high-throughput, low-latency Grand Interconnect

3972

Features

- **Interfaces the fiber-optic Grand Interconnect Highway to CAMAC**
- **Supports Level-1 FASTCAMAC protocol**
- **Exhibits 10 Mbyte/s Interconnect Highway speed**
- **Provides full throughput with a 2 km maximum distance between fiber-optic nodes**
- **32K x 32 command memory**
- **External or programmable internal command memory trigger**
- **Multi-buffer read-data RAM (1, 4, or 8 Mbyte options)**
- **24-bit LAM mask**
- **Priority encoded LAM triggers of command list**
- **Dataway short cycles to increase data throughput**
- **Battery back-up option**
- **Main or Auxiliary Controller operation**



General Description *(Product specifications and descriptions subject to change without notice.)*

The 3972-Z1A is a double-width CAMAC crate controller that functions as an interface between the CAMAC Dataway and the fiber-optic Grand Interconnect (GI) Highway. The 3972-Z1B, -Z1C and -Z1D are triple-width controllers with 1, 4 and 8 Mbyte memory, respectively. The GI allows for up to 126 nodes and may include both VXI V160 Slot-0 controllers and other 3972 crate controllers. The maximum distance between any two nodes is 2 km (6560 ft). The 3972 is a slave device on the highway and receives its commands from a 2961 or 2962 Grand Interconnect Highway Driver. CAMAC transactions are initiated either by manipulating registers on the 3972 or by triggering a list of operations from an on-board, 32K x 32 command memory. The contents of this RAM can generate single, block, and in-line CAMAC I/O functions. Q-Scan, Q-Stop, Q-Repeat, and Q-Ignore modes of operations are selected on a command-by-command basis.

The command list can be triggered from an external source (a TTL-level signal applied to the 3972's front panel connector), by the assertion of a CAMAC Look-at-Me (LAM) signal on the Dataway, or by an internal, crystal-controlled, programmable timer. LAM signal triggers, when enabled, point to a fixed location jump table at the start of the command memory. The programmable timer enables the selection of frequencies from 0.06 Hz to 500 kHz in 1 microsecond increments.

CAMAC data buffering in both the read and write data paths allow read and write commands to be mixed in the command memory. The write-data First-In-First-Out (FIFO) memory is a 2 kbyte memory. The read-data memory is a 2 kbyte FIFO and may be expanded to 8 Mbytes using the multi-buffer option. The multi-buffer options of the 3972 are triple-width CAMAC crate controllers that allow many CAMAC read-data words to be read prior to transmission to the host computer.

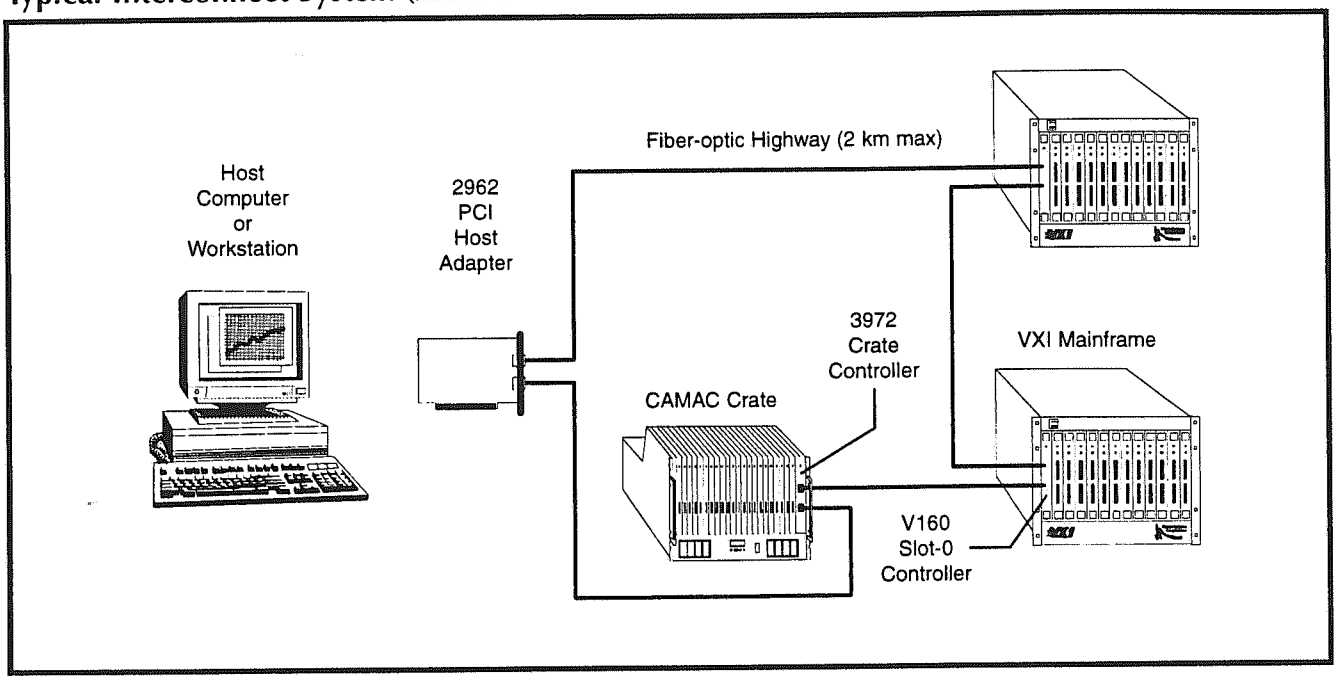
A 24-bit LAM mask is included to simplify host processor software. Selected LAMs are priority encoded and, if enabled, cause Demand messages to be sent on the GI. The Demand ID sent on the GI is the priority encoded value of the selected LAMs.

3972 (continued)

The following shows the list of valid internal N30 commands to the 3972. All accesses return a Q=1 and X=1.

F(0) A(0)	Read List Data Buffer
F(0) A(1)	Read Acquired Multi-buffer Data
F(0) A(8)	Read Multi-buffer Memory Control Register
F(0) A(9)	Read Multi-buffer Memory Address Register
F(0) A(10)	Read Multi-buffer Interval Counter
F(0) A(11)	Read Multi-buffer End Address
F(1) A(0)	Read Control/Status Register
F(1) A(1)	Read Delay Count Register
F(1) A(3)	Read Broadcast Trigger Mask Register
F(1) A(4)	Read List Memory Address Register
F(1) A(5)	Read List Memory Data Register
F(1) A(6)	Read Timer Control Register
F(1) A(7)	Read Timer Data Register
F(1) A(8)	Read Total Transfer Count Register
F(1) A(9)	Read List Transfer Count Register
F(1) A(10)	Read Demand FIFO Register
F(1) A(11)	Read DSP Communication Register
F(1) A(12)	Read LAM Status
F(1) A(13)	Read Demand LAM Mask Register
F(1) A(14)	Read DSP LAM/Trigger Mask Register
F(1) A(15)	Read List Trigger Source Register
F(16) A(0)	Write List Data Buffer
F(17) A(0)	Write Control/Status Register
F(17) A(1)	Write Delay Count Register
F(17) A(2)	Write Trigger Source Register
F(17) A(3)	Write Broadcast Trigger Register
F(17) A(4)	Write List Memory Address Register
F(17) A(5)	Write List Memory Data Register
F(17) A(6)	Write Timer Control Register
F(17) A(7)	Write Timer Data Register
F(17) A(11)	Write DSP Communication Register
F(17) A(13)	Write Demand Communication Register
F(17) A(14)	Write DSP LAM/Trigger Mask Register
F(17) A(15)	Write List Trigger Source Register
F(25) A(0)	Execute List

Typical Interconnect System (mixed CAMAC and VXIbus I/O chassis shown)



Specifications for Fiber-Optic Link

Port connectors	ST-style
Optical wavelength	1300 nm
Maximum distance per link	2 km
Cable core diameter	50, 62.5 or 100 micrometer

Power Requirements

+6 volts:	-Z1A	7.2 A
	-Z1B	9.5 A
	-Z1C	9.6 A
	-Z1D	10.9 A

Ordering Information

Model 3972-Z1A	Crate Controller, for use with Grand Interconnect
Model 3972-Z1B	Crate Controller, for use with Grand Interconnect, with 1 Mbyte memory
Model 3972-Z1C	Crate Controller, for use with Grand Interconnect, with 4 Mbyte memory
Model 3972-Z1D	Crate Controller, for use with Grand Interconnect, with 8 Mbyte memory

Related Products

Model 2961	VME Bus Adapter to the Grand Interconnect
Model 2962	PCI Bus Adapter to the Grand Interconnect
Model V160	Grand Interconnect VXI Slot-0 Controller
Model 5802-Lxyz	Cable - 50 micrometer fiber-optic
Model 5802-Nxyz	Cable - 62.5 micrometer fiber-optic
Model 5857-Axyz	Cable - 1-contact LEMO to unterminated
Model 5857-Bxyz	Cable - 1-contact LEMO to 1-contact LEMO
Model 5857-Hxyz	Cable - 1-contact LEMO to BNC shielded
Model 5910-Z1A	Connector - 1-contact LEMO

Installation

Before the 3972 is installed in a CAMAC chassis, several strap options must be configured. The following chart shows the default values for each strap selection as setup at the factory.

Strap Option	Default Configuration
P2 Hold	Enabled
Q-Repeat Timeout	Enabled
Q-Repeat Timeout Value	250 milliseconds
Controller Configuration	Main Crate Controller
Trigger Out B Termination	Installed
Trigger Out A Termination	Installed
UART Baud Rate	9600

To modify any of the above parameters, please refer to the next section of this manual entitled Strap Options.

After all the user selectable parameters have been set, install the 3972 into the CAMAC chassis. If the 3972 is to be used as a Main Crate Controller, install the 3972 into slots 24 and 25. If a 3972 is used with the Multibuffer Option, the 3972 will occupy slots 23, 24 and 25. If the 3972 is to be used as an Auxiliary CAMAC Controller, insert it into any slot in the CAMAC chassis that does not interfere with slots 24 and 25 of the Main Crate Controller station.

The Request/Grant LEMO connections must now be made to establish the priority for arbitration of the CAMAC Dataway. If the 3972 is to be used as a Main Crate Controller, a single pin LEMO cable is used to connect the 3972 Request LEMO to the 3972 Grant In LEMO connector.

If Auxiliary Controllers are used in the crate along with the 3972 as a main, the Grant Out LEMO of the 3972 is connected to the Grant In LEMO of the second highest priority controller. The Grant Out of the second highest priority controller is connected to the Grant In of the third highest priority controller and so on. The Grant Out LEMO of the controller with the least priority has its Grant Out LEMO unconnected. As with all Auxiliary Controllers, the Auxiliary Controller Bus (ACB) must be bussed to all controllers in the chassis. The 40 position ACB connector is located on the right-most board of the 3972. This connector must be bussed to all other controllers in the chassis with a 40 position flat ribbon cable. The ACB bus is used in the arbitration sequence, it allows the auxiliary controllers to access the LAM signals, and also allows the auxiliary controllers to assert encoded Station Number signals to be driven to each slot by the main controller.

If the 3972 is used as an auxiliary controller in a chassis where the main controller uses Auxiliary Controller Lockout (ACL) to gain control of the Dataway, the 3972 must have the Request LEMO jumpered to the Grant In LEMO. The Grant Out LEMO of the 3972 is then connected to the Grant In LEMO of the next highest priority controller. This sequence continues until all controllers have been connected. The ACB must then be bussed to all controllers in the chassis. The Main Crate Controller that used ACL for arbitration has an adapter card which transforms a high density connector to the 40 position ACB. This ACB connector is then bussed to all other controllers in the chassis by means of a 40 position flat ribbon cable.

After the Data arbitration scheme has been established, the highway connections may then be made. The top fiber-optic connector of the 3972 is for the Highway In. This port is connected to either a Highway Out port of the previous slave (3972/V160) or the Highway Out port of a highway driver (2961/2962). The lower fiber-optic connector is for the Highway Out. This port is connected to the Highway In of the next slave device (3972/V160) or the Highway In of the highway driver (2961/2962).

The only remaining connection is located on the back of the A-board. The A-Board is the left-most board of a 3972 without a multibuffer or the center board of 3972 with a multibuffer. This is a 10 position connector that provides access to the two RS232 serial ports on the 3972. This connection is used for remote diagnostics.

After the 3972 has been configured and the correct cable connections are made, power up all the slave devices and the highway driver. A quick check of highway integrity may be viewed by examining the SYNC LED on each 3972 or at the highway driver. If the highway driver indicates SYNC, the entire highway is properly connected. If the LED on the driver indicates NO SYNC, follow the Highway Out path from the driver until the first slave is reached. If the SYNC LED is not illuminated, the cable from the highway driver to the first slave is bad. Follow the highway until a slave device is found that does not have the SYNC LED illuminated. When this slave is found, the previous link of the highway may be damaged.

Strap Options

Before the 3972 is installed in the chassis, several strap options must be configured. The following chart shows the strap options, their default configuration and the card on which they are located. The B-board is always the right-most board when the 3972 is viewed from the front panel. The A-board is the left-most board of the 3972 when no multibuffer option is used. When the multibuffer option is installed, the A-board is the center board.

Strap Option	Default Configuration	Strap Location	Strap/Switch Designator
P2 Hold	Enabled	A	STP1
Q-Repeat Timeout Enable	Enabled	B	STP3
Q-Repeat Timeout Value	250 milliseconds	B	STP4,5,6
Trigger Out B Termination	Installed	B	STP2
Trigger Out A Termination	Installed	B	STP1
UART Baud Rate	9600	A	SW1
Controller Configuration	Main Crate Controller	A and B	N/A

For the locations of the straps and resistor packs on the 3972, please refer to the two diagrams on the following pages. Any strap non described in the following text should not be altered. These straps are for testing purposes only. The diagrams referred to above show the factory straps and their default positions.

P2 Hold

The CAMAC Dataway connector, at each normal station, includes two free bussed signals, P1 and P2. The 3972 responds to the assertion of P2 as follows: If a module asserts the P2 Hold signal before the 3972 asserts the CAMAC Dataway Strobe S1 during an operation, the completion of the Dataway cycle is delayed by the 3972. During this time, the 3972 continues to assert the CAMAC Busy signal and the addressed Station Number signal on the Dataway. Once the P2 Hold signal is negated, the 3972 completes the delayed operation by asserting S1 and continuing. The P2 Hold option allows asynchronous CAMAC operations to slow modules which incorporate the P2 Hold method for determining module readiness. When these modules are accessed, they assert P2 before S1 time of the Dataway cycle. Thus, the slow module "holds" up the cycle until it is ready to transfer data.

A strap located on the 3972 A-board is used to enable or disable this feature. The P2 Hold option is enabled by placing the strap into the ENA position and disabled by placing the strap into the DIS position.

Q-Repeat Timeout Enable

The Q-Repeat Timeout Enable strap is used to enable or disable the termination of a Q-Repeat mode operation if a Q=1 response is not received within the selected timeout period. Timeout periods range from 25 milliseconds to 250 milliseconds. If a Q-repeat operation is to be executed to devices which may not return a Q=1 response for more than 250 milliseconds, this strap should be placed in the disable position. Care should be taken when disabling this timeout since

the 3972 may get placed in an infinite loop waiting for a Q=1 response. A reset from the highway driver is the only mechanism to clear this condition.

The Q-Repeat Timeout Enable strap is located on the B-board and has a designation of STP3. The Q-Repeat Timeout is enabled by installing the strap into the left-most position and disabled when placed in the right-most position.

Q-Repeat Timeout Value

The Q-Repeat Timeout Value strap selection is used to set the maximum allotted time for a Q-Repeat mode of operation to receive a Q=1 response for a given CAMAC data word. The available selections are 25 milliseconds, 100 milliseconds, and 250 milliseconds. If all the modules to be accessed using a Q-Repeat mode return a Q=1 in less than 25 milliseconds, the Q-Repeat Timeout Value would then be set to 25 milliseconds. The setting of this strap is really determined by the longest expected Q=1 response time for a module accessed in Q-Repeat mode.

The following chart shows the various Q-Repeat Timeout Values and the straps that configure them. This strap option is located on the B-board. Note that only one strap should be loaded for this selection.

Q-Repeat Timeout Value	Strap Location Jumpered
25 milliseconds	STP4
100 milliseconds	STP5
250 milliseconds	STP6

Trigger Out B Termination

The 3972 can assert the Trigger Out B front panel signal when the Trigger Source Register is written by the host computer, by the DSP during a list execution, or by a highway trigger command. This signal is a low-true pulse for a duration of approximately 1 microsecond. The open-collector driver used for this signal is an F38. A 1 Kohm pullup resistor to +5 volts can be strapped to this output. If required, the strap may be placed in the opposite position to disconnect the 1 Kohm pullup.

This strap is located on the B-board and is labeled STP2. Placing the jumper to the left-most position connects the pullup to the output. Placing the jumper to the right-most position disconnects the pullup from the output.

Trigger Out A Termination

The 3972 can assert the Trigger Out A front panel signal when the Trigger Source Register is written by the host computer, by the DSP during a list execution, or by a highway trigger command. This signal is a low-true pulse for a duration of approximately 1 microsecond. The open-collector driver used for this signal is an F38. A 1 Kohm pullup resistor to +5 volts can be strapped to this output. If required, the strap may be placed in the opposite position to disconnect the 1 Kohm pullup.

This strap is located on the B-board and is labeled STP1. Placing the jumper to the left-most position connects the pullup to the output. Placing the jumper to the right-most position disconnects the pullup from the output.

UART Baud Rate Selections

The 3972 contains two RS232 serial channels for communicating with the on-board DSP and executing diagnostics. Currently, only the A channel is used for communication. The B channel is reserved for future enhancements.

A switch located on the A-board is used to define the baud rate settings for the RS232 channels. The switch, labeled SW1, contains eight positions, four for channel A and four for channel B. The four left-most switch positions are used for channel B baud rate selections and the four right-most switch positions are used for channel A. A switch position in the 'open' position yields a logical one ('1') and a switch in the 'closed' position yields a logical zero ('0').

The following chart shows the various baud rates obtained from the binary combinations of the switch settings for channel A.

SW5	SW6	SW7	SW8	Baud Rate
0	0	0	0	50 bps
0	0	0	1	110 bps
0	0	1	0	134.5 bps
0	0	1	1	200 bps
0	1	0	0	300 bps
0	1	0	1	600 bps
0	1	1	0	1200 bps
0	1	1	1	1050 bps
1	0	0	0	2400 bps

SW5	SW6	SW7	SW8	Baud Rate
1	0	0	1	4800 bps
1	0	1	0	7200 bps
1	0	1	1	9600 bps
1	1	0	0	38.4k bps
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

The following chart shows the various baud rates obtained from the binary combinations of the switch settings for channel B.

SW1	SW2	SW3	SW4	Baud Rate
0	0	0	0	50 bps
0	0	0	1	110 bps
0	0	1	0	134.5 bps
0	0	1	1	200 bps
0	1	0	0	300 bps
0	1	0	1	600 bps
0	1	1	0	1200 bps
0	1	1	1	1050 bps
1	0	0	0	2400 bps
1	0	0	1	4800 bps
1	0	1	0	7200 bps
1	0	1	1	9600 bps
1	1	0	0	38.4k bps
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

Auxiliary Controller Configuration

The 3972 may be configured as a Main or Auxiliary CAMAC Crate Controller. The 3972 is shipped from the factory configured as a Main Crate Controller. If the 3972 is to be used as an Auxiliary Crate Controller, the socketed Dataway termination resistor packs must be removed from both the A- and B-boards. The A-board contains six resistor packs that need to be removed and the B-board contains nine. The diagrams on the following pages show the location of the resistor packs along with keying information.

To configure the 3972 for Auxiliary Controller operation, follow the procedure below.

1.) Remove the screws that retain the rear bracket (panel) to the 3972.

2.) Remove the screws that hold the front panel onto the 3972 A-board.

3.) Gently lift up on the rear portion of the A-board in a hinging action since the board cannot be lifted straight up because of the fiber-optic components.

4.) Remove the nine resistor packs as shown in the diagrams on the following pages. The resistor packs on the B-board are RP1, RP2, RP3, RP4, RP5, RP6, RP8, RP9, RP10.

5.) Remove the following resistor packs on the A-board. RP1, RP2, RP3, RP4, RP5 and RP6.

6.) Carefully reinstall the A-board onto the B-board. First, slip the two fiber-optic ports through the front panel. Then, in a hinging motion, slide the A-board onto the B-board making sure all of the interconnection pins are aligned. If any significant pressure is felt when installing the board, re-examine the pins to make sure they are in alignment. It may also help to slightly loosen the two front panel screws on the B-board to allow the front panel to move forward.

7.) Reinstall the front panel screws, the rear panel, and the rear panel screws.

8.) Store the resistor packs away in case it is necessary to configure the 3972 as a Main Crate Controller.

To support multiple controllers in a CAMAC crate, a 40-position connector is required on the rear of each controller for connection to the Auxiliary Controller Bus (ACB). The interconnection is made via a 40 conductor flat ribbon cable. The ACB includes three control signals, five encoded-N signals, and 24 auxiliary LAM signals. The three control signals are Request, Request Inhibit and Auxiliary Controller Lockout (ACL). These signals are described in the following text.

All ACB signals are driven with either tri-state or open-collector drivers. Refer to the Rear Panel Connectors section of this manual for the signal allocation on the ACB connector. The three control signals, ACL, Request, and Request Inhibit are used in the arbitration sequence for control of the CAMAC Dataway. Two modes, Request/Grant and Auxiliary Controller

Lockout, exist for CAMAC Dataway arbitration. The Request/Grant protocol is used by the 3972. The 3972 may also be used in a CAMAC chassis containing a controller that uses the ACL arbitration mode to gain control of the Dataway.

The remaining signals on the ACB are the five encoded-N signals (EN16, EN8, EN4, EN2, EN1) and the 24 auxiliary Look-At-Me (LAM) signals (AL24 through AL1). An auxiliary controller, when executing an addressed CAMAC operation, transmits the binary encoded Station Number to the Main Crate Controller via the ACB encoded-N signals. The main controller decodes these signals and asserts the appropriate CAMAC Station Number (N) signal onto the Dataway from the control station (Slot 25). The CAMAC signals LAM24 through LAM1 (L24-L1) are retransmitted by the main controller to other auxiliary controllers in the crate via the ACB signals AL24 through AL1.

Request/Grant Arbitration

The Request/Grant arbitration mode for control of the Dataway involves the use of the three signals; Request, Request Inhibit, and ACB Grant. The signal Request is common to all controllers via the ACB, creating a wired-OR signal. Also, the signal Request is extended to a single pin LEMO connector on the front panel of each controller supporting the Request/Grant protocol. The signal Request Inhibit is common to all controllers via the ACB. The ACB Grant signal is connected in a daisy-chained fashion between all controllers supporting the Request/Grant protocol. The front panel single pin LEMO connectors, labeled Grant In and Grant Out, are used for connection of the signal ACB Grant as described below. If the 3972 is the only controller in the CAMAC chassis, the connection between the front panel connectors Request and Grant In must be made.

From	To
Request of the highest priority controller	Grant In of the highest priority controller
Grant Out of the highest priority controller	Grant In of the second highest priority controller
Grant Out of the second highest priority controller	Grant In of the third highest priority controller

This pattern continues until the lowest priority controller has been reached. There is no connection to the Grant Out LEMO connector on the lowest priority controller.

Both signals, ACL and Request Inhibit, must be false before a controller may request control of the CAMAC Dataway. If both signals are false, the controller may then assert Request. The assertion of Request is received by the highest priority controller at its front panel LEMO connector Grant In (since Request is jumpered to Grant In on its front panel).

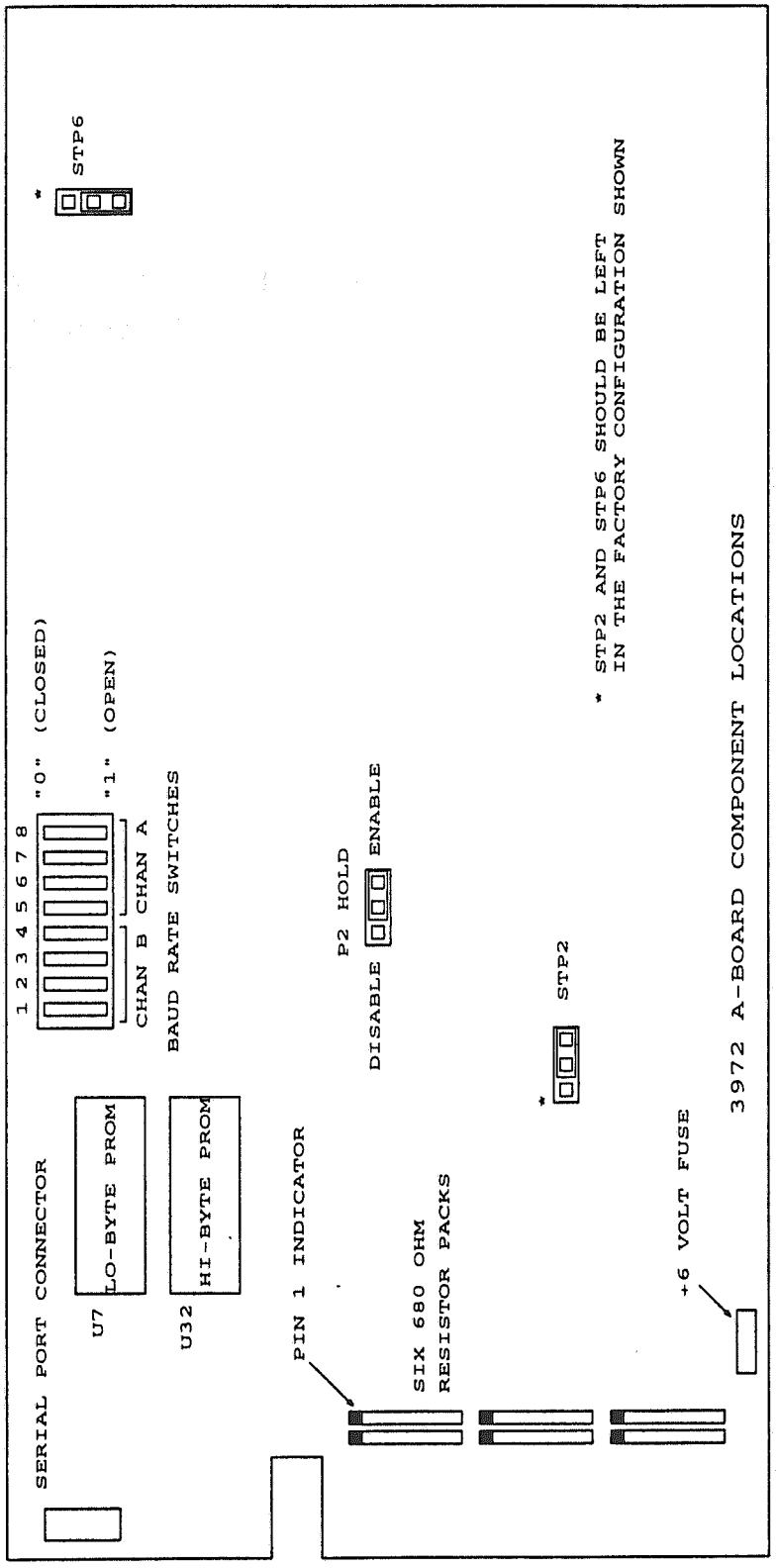
If the highest priority controller did not request control of the Dataway, it asserts the ACB Grant Out signal on its front panel LEMO connector. The Grant signal propagates to other controllers in the order established during the configuration. When the requesting controller receives the assertion of the Grant, the controller accepts it and does pass the Grant any further. The requesting controller then asserts the Request Inhibit signal and takes ownership of the Dataway.

When two or more controllers assert the Request signal at the same time, control of the Dataway is given to the highest priority requesting controller. The propagation of the ACB Grant signal is as described previously. Once the requesting controller has been granted use of the Dataway and asserts Request Inhibit, the other controllers remove their assertion of Request. One or more cycles are then executed by the controller that has the Dataway. To relinquish control of the Dataway, the controller removes the assertion of Request Inhibit signal and the Dataway is then free for further arbitration.

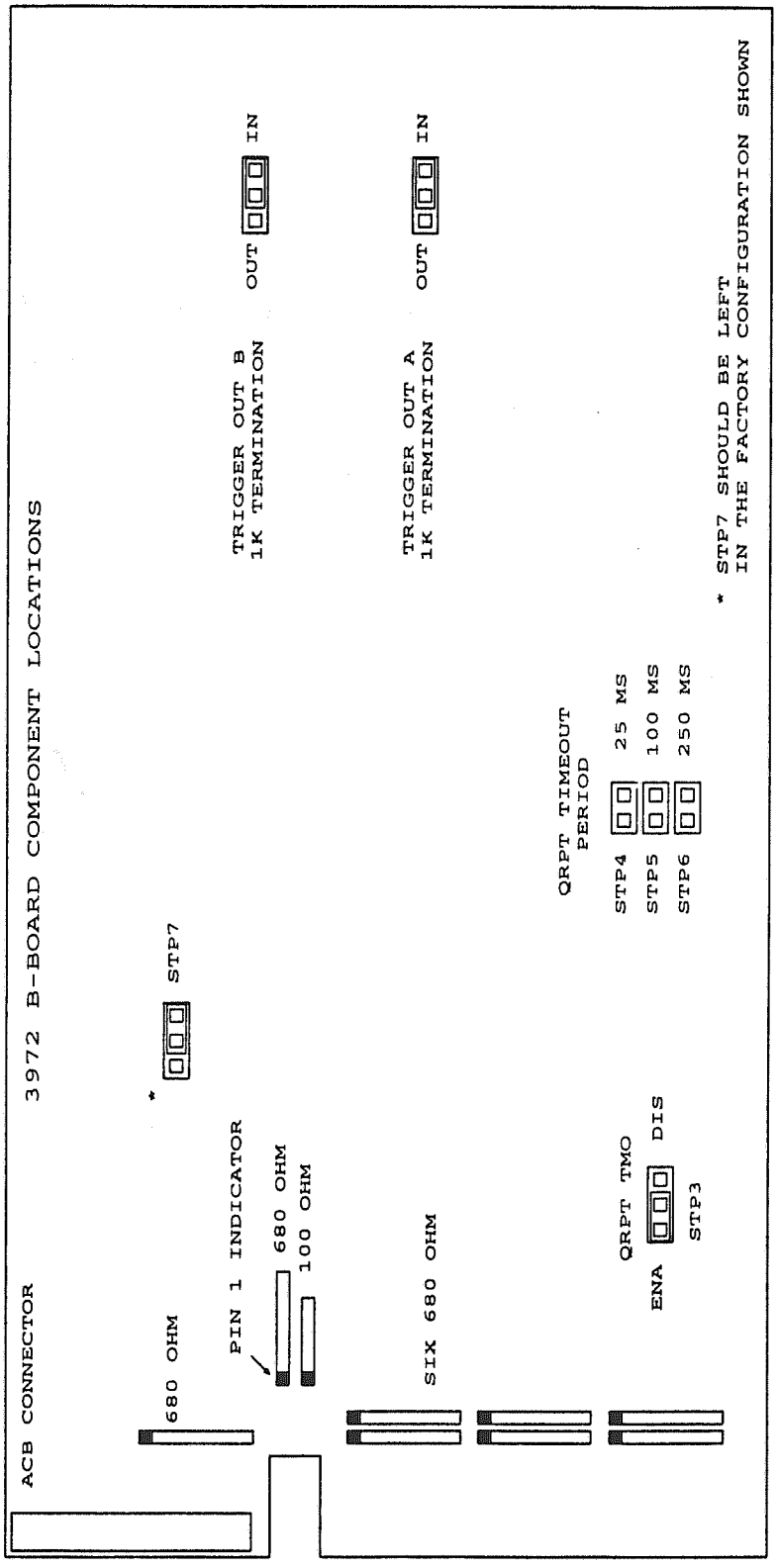
Auxiliary Controller Lockout Arbitration

The Auxiliary Controller Lockout (ACL) mode for gaining control of the CAMAC Dataway involves use of the ACB signal Auxiliary Controller Lockout (ACL). The signal ACL is common to all controllers in a CAMAC chassis via the ACB. Only one controller per chassis is permitted to use this mechanism. All controllers in a chassis respond to the assertion of the ACL signal as follows. When ACL is asserted by a main controller, any CAMAC operation in progress is terminated if the CAMAC Dataway signal Strobe S1 has not been asserted for the current cycle. If the assertion of S1 has already occurred, the CAMAC cycle is allowed to complete as normal. After the cycle is complete, control of the Dataway is given to the controller asserting ACL.

3972 A-Board Component Locations



3972 B-Board Component Locations



3972 Internal Register Descriptions

All access to the internal registers of the 3972 must be 32-bits in length, even if the register is less than 32-bits wide. The registers must be accessed with single transfers with the exception of the following which may be accessed using block transfer modes.

Read List Data Buffer
Write List Data Buffer
Read List Memory Data Register
Write List Memory Data Register
Read Acquired Multibuffer Data

The following shows the list of valid internal N30 commands to the 3972. All accesses return a Q=1 and X=1.

F(0) A(0)	Read List Data Buffer
F(0) A(1)	Read Acquired Multibuffer Data
F(0) A(8)	Read Multibuffer Memory Control Register
F(0) A(9)	Read Multibuffer Memory Address Register
F(0) A(10)	Read Multibuffer Interval Counter
F(0) A(11)	Read Multibuffer End Address
F(1) A(0)	Read Control/Status Register
F(1) A(1)	Read Delay Count Register
F(1) A(3)	Read Broadcast Trigger Mask Register
F(1) A(4)	Read List Memory Address Register
F(1) A(5)	Read List Memory Data Register
F(1) A(6)	Read Timer Control Register
F(1) A(7)	Read Timer Data Register
F(1) A(8)	Read Total Transfer Count Register
F(1) A(9)	Read List Transfer Count Register
F(1) A(10)	Read Demand FIFO Register
F(1) A(11)	Read DSP Communication Register
F(1) A(12)	Read LAM Status
F(1) A(13)	Read Demand LAM Mask Register
F(1) A(14)	Read DSP LAM/Trigger Mask Register
F(1) A(15)	Read List Trigger Source Register
F(16) A(0)	Write List Data Buffer
F(17) A(0)	Write Control/Status Register
F(17) A(1)	Write Delay Count Register
F(17) A(2)	Write Trigger Source Register
F(17) A(3)	Write Broadcast Trigger Register
F(17) A(4)	Write List Memory Address Register
F(17) A(5)	Write List Memory Data Register

F(17) A(6) Write Timer Control Register
F(17) A(7) Write Timer Data Register
F(17) A(11) Write DSP Communication Register
F(17) A(13) Write Demand LAM Mask Register
F(17) A(14) Write DSP LAM/Trigger Mask Register
F(17) A(15) Write List Trigger Source Register

F(25) A(0) Execute List

The following sections describe the registers available on the 3972.

Control/Status Register

The Control/Status Registers is used to monitor and control basic internal 3972 functions. This register is written using an F(17) A(0) command and read with an F(1) A(0). The following shows the bits defined for the Control/Status Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSVD	TMR ENA	LIST BUSY	DMD OFLO	DMD CLR	DMD PEND	DMD MENA	MBM DENA	LAM DENA	MBM ENA	DLY ENA	L24	RD INH	SET INH	C	Z

<31:16> These bits are not used and read as zeros.

<15> RESERVED is a write/read bit that is reserved for future use.

<14> TIMER ENABLE is a write/read bit used to enable the free running timer on the 3972. This timer can be used to execute timer initiated list processing functions or assert triggers at a predetermined interval. The timer is enabled when this bit is set to a one and disabled with a zero.

<13> LIST BUSY is a read-only bit that is set to a one while the list processor is busy. When a list processing operation is complete, this bit is read as a zero.

<12> DEMAND OVERFLOW is a read-only bit that is set to a one when the Demand FIFO is full, with 2048 demand sources, and another demand source is received. This bit is cleared on power-up and by writing the DEMAND CLEAR bit in this register to a one. Note that this bit is not cleared when the Demand FIFO contents are read. It must be explicitly cleared by writing to the DEMAND CLEAR bit in the Control/Status Register.

<11> DEMAND CLEAR is a write-only bit that is used to reset the Demand FIFO Register, the DEMAND OVERFLOW bit, and the DEMAND PENDING bit. This bit is not latched and read as a zero.

<10> DEMAND PENDING is a read-only bit that is set as long as there is at least one demand contained in the Demand FIFO Register. When the Demand FIFO Register is empty, this bit is returned as a zero.

<9> DEMAND MESSAGE ENABLE is a write/read bit used to enable the generation of demand messages onto the highway when a demand source is written into the Demand FIFO Register. If this bit is set to a one, demand messages are transmitted onto the highway as they are received. Setting this bit to a zero prevents the 3972 from generating highway demand messages.

<8> MULTIBUFFER MEMORY DEMAND SOURCE ENABLE is a write/read bit used to enable the multibuffer demand source for entry into the Demand FIFO. The multibuffer may generate a demand source when the Multibuffer Interval Counter is exhausted. Setting this bit to a one enables the source and set to a zero to disable the source.

<7> Look-At-Me (LAM) DEMAND SOURCE ENABLE is a write/read bit used to enable a LAM source for entry into the Demand FIFO. If a LAM is sourced and this bit is set to a one, the LAM demand source is transferred to the Demand FIFO. A zero in this bit location disables LAM sources from generating demand messages.

<6> MULTIBUFFER ENABLE is a write/read bit that is used to enable the multibuffer option. This option is used for storing up to 4 megabytes of CAMAC read data before transmission to the host computer. Setting this bit to a one enables the multibuffer and is set to a zero to disable multibuffer operation.

<5> DELAY ENABLE is a write/read bit used to enable the programmable delay associated with the receipt of a Broadcast Trigger Message. The delay is programmable through the Delay Count Register. Setting this bit to a one enables the delay once the message is received and a setting of zero removes the delay.

<4> LAM24 is a write/read bit used to assert the LAM signal in slot 24 when the 3972 is configured as a main crate controller. Setting this bit to a one asserts the LAM24 signal and a zero clears the signal.

<3> READ INHIBIT is a read-only bit that reflects the state of the CAMAC INHIBIT signal. If the INHIBIT signal is asserted on the CAMAC Dataway, this bit is read back as a one. A zero is returned if the INHIBIT signal is negated.

<2> SET INHIBIT is a write/read bit used to assert the CAMAC INHIBIT signal. Setting this bit to a one asserts INHIBIT and a zero prevents the 3972 from asserting INHIBIT.

<1> Z (INITIALIZE) is a write-only bit that is used to generate a CAMAC Initialize (Z) cycle when written to a one. This bit is not latched and read as a zero.

<0> C (CLEAR) is a write-only bit that is used to generate a CAMAC Clear (C) cycle when written to a one. This bit is not latched and read as a zero.

Total Transfer Count Register

The Total Transfer Count Register is a read-only register and is used to verify the number of transfers remaining to be executed after a block transfer operation terminates due to an error. This register is read by an F(1) A(8) command.

The 3972 Total Transfer Count Register (TTCR) is loaded by a 32-bit value that accompanies any write/read request message directed at the 3972. This 32-bit value is in two's complement format and represents the total number of transfers to execute for the requested operation. Each transfer the 3972 executes causes the TTCR to be incremented by one. The requested operation continues until the TTCR is incremented to zero or an error occurs. If an error occurs during a block transfer operation requested from the highway, the TTCR may be read to determine the number of transfers not executed. This data may then be subtracted from the original transfer count to find the number of valid transfers executed.

The following diagram shows the bit pattern for the Total Transfer Count Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TTCR 31	TTCR 30	TTCR 29	TTCR 28	TTCR 27	TTCR 26	TTCR 25	TTCR 24	TTCR 23	TTCR 22	TTCR 21	TTCR 20	TTCR 19	TTCR 18	TTCR 17	TTCR 16
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
TTCR 15	TTCR 14	TTCR 13	TTCR 12	TTCR 11	TTCR 10	TTCR 9	TTCR 8	TTCR 7	TTCR 6	TTCR 5	TTCR 4	TTCR 3	TTCR 2	TTCR 1	TTCR 0

<31:0> TOTAL TRANSFER COUNT 31 through 0 are read-only bits which reflect the number of transfers remaining after a block transfer operation terminates.

List Transfer Count Register

The List Transfer Count Register is a read-only register used to verify the number of valid transfers remaining to be executed after a block transfer operation executed from the list has terminated due to an error. This register is read using an F(1) A(9) command.

The List Transfer Count Register is loaded with the transfer count specification when a Block Transfer instruction is encountered during a list processing operation. This transfer count specifies the number of Dataway operations to execute for the block transfer. If an error occurs during the block operation, the List Transfer Count Register may be read to determine the

number of transfers remaining. The format of the data in this register is two's complement. The data read from this register may then be subtracted from the original transfer count to determine the number of valid transfers that were executed.

The following diagram shows the bit pattern for the List Transfer Count Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LTCR 31	LTCR 30	LTCR 29	LTCR 28	LTCR 27	LTCR 26	LTCR 25	LTCR 24	LTCR 23	LTCR 22	LTCR 21	LTCR 20	LTCR 19	LTCR 18	LTCR 17	LTCR 16
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
LTCR 15	LTCR 14	LTCR 13	LTCR 12	LTCR 11	LTCR 10	LTCR 9	LTCR 8	LTCR 7	LTCR 6	LTCR 5	LTCR 4	LTCR 3	LTCR 2	LTCR 1	LTCR 0

<31:0> LIST TRANSFER COUNT REGISTER 31 through 0 are read-only bits which indicate the number of transfer remaining to execute after a block transfer list operation is terminated. The format of this data is two's complement.

Demand FIFO Register

The Demand FIFO Register (DFR) is a read-only register used to "hold" demand messages prior to transmission onto the highway. This register is read with an F(1) A(10) command.

The Demand FIFO Register is capable of retaining 2048 demand entries. These entries are generated by either CAMAC Look-At-Me (LAM) signals, the List Processor, or by the multibuffer memory. If the DEMAND MESSAGE ENABLE bit in the Control/Status Register is set to a one, every entry that is placed in the Demand FIFO is transmitted onto the highway as a demand message. If polling is desired, the DEMAND MESSAGE ENABLE bit may be set to a zero and the demand entries read out through the Demand FIFO Register. A test for additional demand entries must be made through the CSR using the DEMAND PENDING bit.

Demand sources are written into the FIFO as they occur. A CAMAC LAM is encoded before it is entered into the DFR. The entry reflects the slot in which the LAM was sourced. The actual demand FIFO entry is one less than the LAM source generated. For example, an entry of 8 indicates that slot 9 in the CAMAC chassis asserted a LAM. The valid range of numbers for the CAMAC LAMs is from 0 to 23. Any values greater than this are generated by other mechanisms on the 3972. The other mechanisms include the List Processor writing the Demand FIFO and a Multibuffer Flag entry.

The Demand LAM Mask Register must be loaded prior to any CAMAC LAMs generating demand messages. This register must be loaded with the mask pattern for allowing the selected LAM signals to generate entries in the Demand FIFO. Refer to the Demand LAM Mask Register section of this manual for additional information.

The List Processor is another source of data for the Demand FIFO. The List Processor may be instructed to write a particular data pattern into the FIFO and pass that information to the host in the form of a demand message. A list instruction is available for writing a data pattern into the Demand FIFO. The range of data for this source is 0 to 255. Care should be taken when using data patterns of 0 through 23 and 32 since they cannot be differentiated from CAMAC LAM patterns and the Multibuffer Flag entry.

The last source of a demand is from the multibuffer. The multibuffer circuitry can write a demand entry when the Multibuffer Interval Counter expires. When this counter is exhausted, a Demand Message may be sent to the host computer indicating that a segment of the multibuffer has been filled and needs to be read out. This demand source is enabled using the MULTIBUFFER MEMORY DEMAND SOURCE ENABLE bit in the Control/Status Register. The data entered into the FIFO for this source is 32.

When the Demand FIFO is enabled to generate Demand Messages, the data contained in the FIFO is returned to the host in the Demand Identification field. Please refer to the Interconnect Highway Driver manual for additional information on reading Demand Messages.

The following chart shows the various demand sources and the demand identification fields they generate.

Demand Source	Demand Data Range
CAMAC LAMs 1 Through 24	0 Through 23
List Processor Write Demand FIFO Instruction	0 Through 255
Multibuffer Interval Counter	32

Demand LAM Mask Register

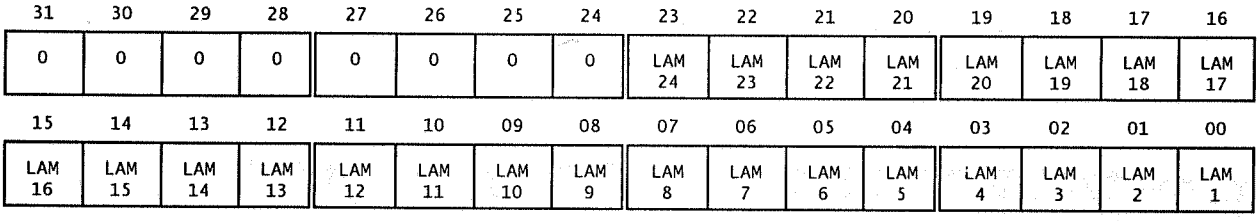
The Demand LAM Mask Register is a write/read register used to enable preselected LAMs to enter the Demand FIFO when they are asserted. The Demand LAM Mask Register is written with an F(17) A(13) and read with an F(1) A(13) command.

This register is used to select the CAMAC Look-At-Me (LAM) signals that are allowed to be written into the Demand FIFO. Once a selected LAM is asserted and written to the FIFO, it may be transmitted to the host computer as a Demand Message. The DEMAND ENABLE bit in the Control/Status Register must be set to a one in order to generate a Demand Message.

This mask register allows either all or a subset of the CAMAC LAMs to be handled by the host computer. The DSP LAM/Trigger Mask Register has a similar selection mechanism. These two registers allow routing of the LAM sources to either the Demand FIFO, the List Processor or both. Setting a bit to a one in these registers allows the signal to be routed to the appropriate location.

Model 3972

The following diagram shows the bit pattern for the Demand LAM Mask Register.



<31:24> These bits are not used and read as zeros.

<23:0> LOOK-AT-ME (LAM) 24 through 1 are write/read bits used to enable the LAM source to be written into the Demand FIFO.

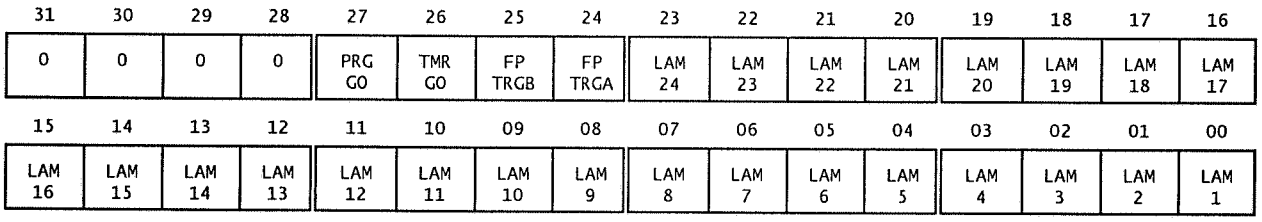
DSP LAM/Trigger Mask Register

The DSP LAM/Trigger Mask Register is a write/read register used to enable preselected LAMs to be latched and presented to the (List Processor) when they are asserted. The DSP LAM/Trigger Mask Register is written with an F(17) A(14) and read with an F(1) A(14) command.

This register is used to select the CAMAC Look-At-Me (LAM) signals that are allowed to be latched and presented to the DSP. Once an enabled LAM is asserted and latched, it is available to the DSP. This register can also be enabled to allow the DSP to examine the list trigger sources. This allows the DSP to execute conditional instructions based on the source of list triggers. The DSP can differentiate between list initiated operations from LAMs, front panel triggers, timer initiated triggers, and programmed I/O initiation from the host.

This mask register allows either all or a subset of the CAMAC LAMs to be handled by the DSP. The Demand LAM Mask Register has a similar selection mechanism. These two registers allow routing of the LAM sources to either the Demand FIFO, the DSP or both. Setting a bit to a one in these registers allows the signal to be routed to the appropriate location.

The following diagram shows the bit pattern for the Demand LAM Mask Register.



<31:28> These bits are not used and read as zeros.

<27> PROGRAMMED I/O GO is a write/read bit is used to enable the list trigger source from programmed I/O to be seen by the DSP.

<26> TIMER GO is a write/read bit used to enable the list trigger source from a timer expiration to be seen by the DSP.

<25:24> FRONT PANEL TRIGGER IN B and A are write/read bits to used to enable the list trigger source from the front panel triggers to be seen by the DSP.

<23:0> LOOK-AT-ME (LAM) 24 through 1 are write/read bits used to enable the entry of the LAM source to be latched and seen by the DSP.

LAM Status Register

The LAM Status Register is a read-only register used to obtain the current status of the LAM signals on the CAMAC Dataway. This register is read with an F(1) A(12). The following diagram shows the bit layout for the LAM Status Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	LAM 24	LAM 23	LAM 22	LAM 21	LAM 20	LAM 19	LAM 18	LAM 17

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
LAM 16	LAM 15	LAM 14	LAM 13	LAM 12	LAM 11	LAM 10	LAM 9	LAM 8	LAM 7	LAM 6	LAM 5	LAM 4	LAM 3	LAM 2	LAM 1

<31:24> These bits are not used and read as zeros.

<23:0> LAM STATUS 24 through 1 are read-only bits which reflect the current status of the 24 CAMAC LAM signals.

Trigger Source Register

The Trigger Source Register is a write-only register used to assert the front panel trigger signals, execute a list processing operation, and reset the internal time stamp counters. This register is written with an F(17) A(2) command.

When this register is written, any bit set to a one causes a 1 microsecond pulse to be applied to that signal. The following diagram shows the bit pattern for the Trigger Source Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	0	0	0	RST TSTP	LIST GO	FP TRGB	FP TRGA

<31:4> These bits are not used and must be set to zero.

<3> RESET TIME STAMP is a write-only bit used to reset the internal time stamp counters on the 3972.

<2> LIST GO is write-only bit used to initiate a list processing operation at the address specified in the List Memory Address Register.

<1:0> FRONT PANEL TRIGGER OUT B and A are write-only bits used to assert the two front panel signals TRIGGER OUT B and TRIGGER OUT A respectively.

Broadcast Trigger Mask Register

The Broadcast Trigger Mask Register is a write/read register used to specify the signal(s) to assert when the 3972 receives a Broadcast Trigger message. This register is written with an F(17) A(3) and read with an F(1) A(3) command.

An Interconnect Highway Driver may issue a Broadcast Trigger message which informs all slaves connected to the highway to assert the data contained in the Broadcast Trigger Mask Register. On the 3972, this register contains two front panel trigger outputs, an execute list bit, and a clear time stamp bit. Any bit set to a one in this register is asserted once a Broadcast Trigger message is received.

Associated with the Broadcast Trigger mechanism is the Delay Timer. The Delay Timer, if enabled, causes the 3972 to delay the assertion of the Broadcast Trigger Mask Register bits until a predetermined interval of time elapses. The delay data is loaded into the Delay Count Register.

The following diagram shows the bit pattern for the Broadcast Trigger Mask Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
0	0	0	0	0	0	0	0	0	0	0	0	RST TSTP	LIST GO	FP TRGB	FP TRGA

<31:4> These bits are not used and must be set to zero.

<3> RESET TIME STAMP is a write-only bit used to reset the internal time stamp counters on the 3972.

<2> LIST GO is write-only bit used to initiate a list processing operation at the address specified in the List Memory Address Register.

<1:0> FRONT PANEL TRIGGER OUT B and A are write-only bits used to assert the two front panel signals TRIGGER OUT B and TRIGGER OUT A respectively.

Delay Count Register

The Delay Count Register is a write/read register used to specify a delay count associated with the receipt of a Broadcast Trigger Message from the highway. This register is written with an F(17) A(1) command and read with an F(1) A(1).

When the 3972 receives a Broadcast Trigger Message, it may assert the data contained in the Broadcast Trigger Mask Register immediately or insert a delay. The delay count is specified by using this register and allows a delay of 500 nanoseconds to 13.1 milliseconds in 200 nanosecond increments.
After the trigger message is received and the DELAY ENABLE bit in the Control/Status Register is set to a one, the delay count is decremented until is reaches zero. At this time, the bits of the Broadcast Trigger Mask Register that are set to a one are asserted. Refer to the Broadcast Trigger Mask Register section of this manual for additional information.

There is a 500 nanosecond minimum delay time within the 3972 before the Broadcast Trigger data is sourced. To calculate the actual delay time, use the following formula.

Time = (500 nanoseconds) + (Delay Count * 200 nanoseconds)

The following diagram shows the bit pattern for the Delay Count Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
DCT 15	DCT 14	DCT 13	DCT 12	DCT 11	DCT 10	DCT 9	DCT 8	DCT 7	DCT 6	DCT 5	DCT 4	DCT 3	DCT 2	DCT 1	DCT 0

<31:16> These bits are not used and read as zeros.

<15:0> DELAY COUNT 15 through 0 are write/read bits used to specify the delay count used once a Broadcast trigger is received and the DELAY COUNT ENABLE bit in the Control/Status Register is set.

List Memory Address Register

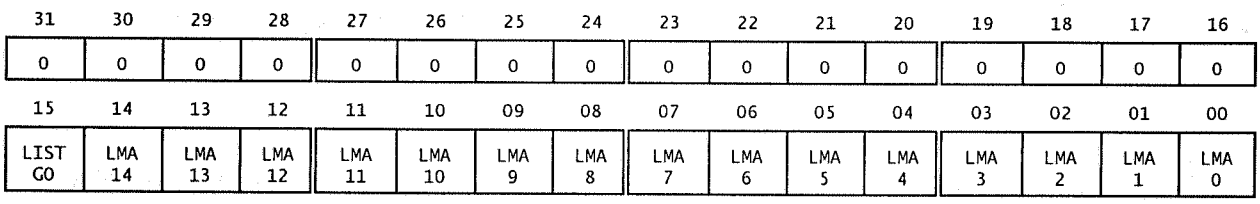
The List Memory Address Register (LMA) is a write/read register used to specify the list address prior to writing or reading data from the List Memory Data Register and for specifying the initial address for list processing execution. This register is written with an F(17) A(4) and read with an F(1) A(4) command.

The List Memory Address Register is written with the memory address to be used for subsequent List Memory Data Register write and read operations. Before accessing the 32Kx32 list memory, the LMA must be written with the desired location to access. After the address is loaded, the List Memory Data Register (LMD) may then be accessed. After a write or read operation to the LMD, the LMA is automatically incremented to the next address location.

This register is also used for specifying the initial address at which list processing operations are to begin. Once a list processing operation has ended, this register may be read to find the last address accessed by the list processor. Since the list processor autoincrements the list memory address, the data read after a list processing operation completes must be decremented by one to point to the last element actually processed.

The LIST GO bit in this register may be set to a one to initiate a list processing operation. This allows one write operation to the 3972 to set the initial address for list processing and initiate the list.

The following diagram shows the bit pattern for the List Memory Address Register.



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
LIST GO	LMA 14	LMA 13	LMA 12	LMA 11	LMA 10	LMA 9	LMA 8	LMA 7	LMA 6	LMA 5	LMA 4	LMA 3	LMA 2	LMA 1	LMA 0

<31:16> These bits are not used and read as zeros.

<15> LIST GO is a write-only bit used to initiate a list processing operation at the specified address. This bit is not latched and read as a zero.

<14:0> LIST MEMORY ADDRESS 14 through 0 are write/read bits used to specify list memory location to access for List Memory Data write and read operations. These bits are also used to specify the address at which list processing operations are to start.

List Memory Data Register

The List Memory Data Register (LMD) is a write/read register used to access the 32Kx32 list processing memory. This register is written with an F(17) A(5) and read with an F(1) A(5) command.

The list memory is loaded prior to list triggering with the CAMAC NAFs and other commands to be executed. Before a write or read operation is executed to this register, the List Memory Address Register (LMA) must be loaded with the initial address to access. After a write or read to the LMD, the LMA is automatically incremented to the next address location. This allows for sequential access to the LMD without having to reload the LMA each time.

This register may be accessed using block transfer type instructions. The following diagram shows the bit pattern for the List Memory Data Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LMD 31	LMD 30	LMD 29	LMD 28	LMD 27	LMD 26	LMD 25	LMD 24	LMD 23	LMD 22	LMD 21	LMD 20	LMD 19	LMD 18	LMD 17	LMD 16
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
LMD 15	LMD 14	LMD 13	LMD 12	LMD 11	LMD 10	LMD 9	LMD 8	LMD 7	LMD 6	LMD 5	LMD 4	LMD 3	LMD 2	LMD 1	LMD 0

<31:0> LIST MEMORY DATA 31 through 0 are write/read bits used to specify the instruction that the list processor executes during a list processor operation.

List Data Buffer

The List Data Buffer is a write/read register used to transfer data to/from CAMAC modules that are accessed when a list operation is executed. This register is written with an F(16)A(0) command and read with an F(0)A(0) command.

After a command list has been downloaded into the 3972, a mechanism must exist to transfer data to/from the 3972 for the list instructions. The data transfers occur through the List Data Buffer Register. If a legal write list has been downloaded to the 3972, the CAMAC write data must be supplied by executing a block write operation to the List Data Buffer Register (LDBR). If the list was previously idle, the write operation to this register will start the list processing. Data supplied by the host computer is used as the write data stream for the list operations. If a legal read list has been downloaded to the 3972, the list may be started by asynchronous means or by the host computer reading the LDBR. Once the list is executing, read data obtained as the result of CAMAC activity is passed to the host computer. The only exception to this is if the multibuffer option is used and enabled. In these cases, an asynchronous trigger of the list causes the CAMAC read data obtained to be placed in the multibuffer memory for later readout by the host computer.

The following diagram shows the bit layout for the List Data Buffer Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LDB 31	LDB 30	LDB 29	LDB 28	LDB 27	LDB 26	LDB 25	LDB 24	LDB 23	LDB 22	LDB 21	LDB 20	LDB 19	LDB 18	LDB 17	LDB 16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LDB 15	LDB 14	LDB 13	LDB 12	LDB 11	LDB 10	LDB 9	LDB 8	LDB 7	LDB 6	LDB 5	LDB 4	LDB 3	LDB 2	LDB 1	LDB 0

<31:0> LIST DATA BUFFER 31 through 0 are write/read bits used to transfer data to/from the 3972 during list execution.

List Trigger Source Register

The List Trigger Register is a write/read register used to enable the various sources of triggering list execution. This register is written with an F(17) A(15) and read with an F(1) A(15) command.

This register is used to select which asynchronous event may initiate list execution. These events include the Look-At-Me (LAM) events and the front panel trigger inputs. This register allows any or all of the events to start list processing. If it is desired to have different lists to be executed based on the actual trigger source, the DSP LAM/Trigger Mask Register must be used in combination with special DSP instructions to differentiate between the sources.

The following diagram shows the bit pattern for the List Trigger Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	FP TRGB	FP TRGA	LAM 24	LAM 23	LAM 22	LAM 21	LAM 20	LAM 19	LAM 18	LAM 17
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
LAM 16	LAM 15	LAM 14	LAM 13	LAM 12	LAM 11	LAM 10	LAM 9	LAM 8	LAM 7	LAM 6	LAM 5	LAM 4	LAM 3	LAM 2	LAM 1

<31:26> These bits are not used and read as zeros.

<25:24> FRONT PANEL TRIGGER IN B and A are write/read bits used to enable/disable the assertion of the front panel trigger input signals from initiating list processing operations. Setting a bit to a one enables the assertion of the corresponding signal to start a list and a zero disables the source.

<24:0> LOOK-AT-ME (LAM) 24 through 1 are write/read bits used to enable/disable the assertion of the corresponding CAMAC LAM signals from initiating list processing operations. Setting a bit to a one enables the assertion of the corresponding signal to start a list and a zero disables the source.

Timer Control Register

The Timer Control Register is a write/read register used to specify the signals to be asserted once the programmable timer has expired. This register is written with an F(17) A(6) and read with an F(1) A(6) command.

The Timer Control Register contains bits to enable individual signals to be asserted once the timer expires. These signals include front panel trigger out, reset time stamp, and executing a list operation. The Timer Data Register is first loaded with the interval at which the timer operates and is enabled using the TIMER ENABLE bit in the Control/Status Register. After the timer expires, any bit set to a one in the Timer Data Register causes the corresponding signal to be asserted.

The following diagram shows the bit pattern for the Timer Data Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
0	0	0	0	0	0	0	0	0	0	0	0	RST TSTP	LIST GO	FP TRGB	FP TRGA

<31:4> These bits are not used and must be set to zero.

<3> RESET TIME STAMP is a write-only bit used to reset the internal time stamp counters on the 3972.

<2> LIST GO is write-only bit used to initiate a list processing operation at the address specified in the List Memory Address Register.

<1:0> FRONT PANEL TRIGGER OUTB and A are write-only bits used to assert the two front panel signals TRIGGER OUT B and TRIGGER OUT A.

Timer Data Register

The Timer Data Register is a write/read register used to establish the timing interval for asserting the signals described in the Timer Control Register. This register is written with an F(17) A(7) and read with an F(1) A(7) command.

This register is used for specifying the interval used for sourcing front panel trigger outputs or initiating list execution. This register is composed of a 24-bit counter that is clocked by a 1 megahertz clock. The allowable range for the timer is from 5 microseconds to 16.77 seconds in 1 microsecond increments. After the timer is enabled in the Control/Status Register, the timer is counted down until is reaches zero. When the counter expires, the data contained in the Timer Control Register is sourced and the timer is reloaded. This continues until the timer is disabled.

The following diagram shows the bit pattern for the Timer Data Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	TDR 23	TDR 22	TDR 21	TDR 20	TDR 19	TDR 18	TDR 17	TDR 16
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
TDR 15	TDR 14	TDR 13	TDR 12	TDR 11	TDR 10	TDR 9	TDR 8	TDR 7	TDR 6	TDR 5	TDR 4	TDR 3	TDR 2	TDR 1	TDR 0

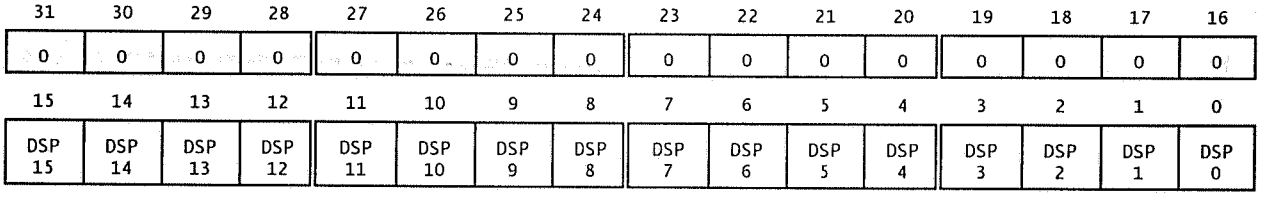
<31:24> These bits are not used and read as zeros.

<23:0> TIMER DATA REGISTER 23 through 0 are write/read bits used to specify the rate at which the timer operates. The value loaded in this register is the number if microseconds used as the interval.

DSP Communication Register

The DSP Communication Register is a write/read register used to communicate with the DSP on the 3972. This register is written with an F(17)A(11) command and read with an F(1)A(11) command. This register is currently reserved but shown here for completeness. This register may be used at a later date to implement functions that require direct handshaking from the host computer and the DSP. Note that a read from this register does not yield the data written. The data written to this register is read by the DSP and the data read from this register is data written by the DSP.

The following shows the bit pattern of the DSP Communication Register.



<31:16> These bits are not used and read as zeros.

<15:0> DSP 15 through 0 are used as a communication path to/from the DSP.

Execute List

The F(25) A(0) command is used to initiate a list processing operation at the address specified in the List Memory Address Register. Since this is a CAMAC control operation, no data accompanies this operation.

The following register descriptions are applicable only to 3972 models that include the Multibuffer Memory option.

Multibuffer Memory Address Register

The Multibuffer Memory Address Register is used to specify an address within a buffer of acquired data prior to readout. This register is written with an F(16)A(9) and read with an F(0)A(9) command. To read a buffer location, the Multibuffer Memory Address Register (MMAR) must first be loaded with the address of the location to be accessed. This address points to a 32-bit value in the buffer. The Multibuffer Memory Data Register may then be read to retrieve the buffer contents.

When the Multibuffer Memory Data Register (MMDR) is read, the MMAR is automatically incremented to the next buffer location. This permits successive reads of the MMDR without having to reload the MMAR for each access.

Refer to the Multibuffered Memory Option section of this manual for additional information. The following diagram shows the bit layout for the Multibuffer Memory Address Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	MBM A19	MBM A18	MBM A17	MBM A16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MBM A15	MBM A14	MAM A13	MBM A12	MBM A11	MBM A10	MBM A9	MBM A8	MBM A7	MBM A6	MBM A5	MBM A4	MBM A3	MBM A2	MBM A1	MBM A0

<31:20> These bits are not used and read as zeros.

<19:0> MULTIBUFFER MEMORY ADDRESS 19 through 0 are write/read bits used to specify the initial readout address when accessing the Multibuffer Memory Data Register.

Multibuffer Memory Data Register

The Multibuffer Memory Data Register is a read-only register that is used to retrieve data acquired during multibuffered data acquisition. The register is read with an F(0) A(1) command.

The buffer memory is used to 'hold' read data acquired during multibuffer read operations. The buffer allows up to 4 megabytes of data to be stored before readout by the host computer. The following chart shows the bit pattern for the Multibuffer Memory Address Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBM D31	MBM D30	MBM D29	MBM D28	MBM D27	MBM D26	MBM D25	MBM D24	MBM D23	MBM D22	MBM D21	MBM D20	MBM D19	MBM D18	MBM D17	MBM D16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MBM D15	MBM D14	MBM D13	MBM D12	MBM D11	MBM D10	MBM D9	MBM D8	MBM D7	MBM D6	MBM D5	MBM D4	MBM D3	MBM D2	MBM D1	MBM D0

Multibuffer Memory Control Register

The Multibuffer Memory Control Register is a write/read register used to monitor the status of the multibuffer flags and to clear the flags once the multibuffer data has been read. This register is written by an F(16)A(8) and read with an F(0)A(8) command.

The flag bits in this register are incrementally set as the Multibuffer Interval Counter (MIC) expires. The MIC is decremented for each 32-bit entry placed into the buffer memory. When

the MIC is decremented to zero, a flag bit is set indicating that the current buffer segment is ready for readout. Starting with FLAG1, the bits are sequentially set until FLAG8 is set. After FLAG8 is set, the count rolls over and the next flag to be set is again FLAG1. It is not necessary for an application to use all eight buffer flags. The number of flags used depends on the segmentation of the buffer. If the Multibuffer Interval Counter is set to one-fourth the setting of the Multibuffer End Address, only four flags are used in the sequence. Once a FLAG has been set, it must be cleared after the corresponding buffer segment has been read out. If the host computer cannot sustain the required readout rate, a buffer overrun condition may occur. The overrun condition is set when the 3972 tries to fill a segment of the buffer memory and the corresponding flag bit has not been cleared.

This register also contains a feature to clear the flags automatically as the individual buffer segments are read out by the host computer. When one of the flag bits is set, the host can read the acquired buffer segment which will automatically clear the flag bit. This allows the flag bits to be cleared without the need of a write operation to selectively clear the flag. When flags are cleared in this manner, the read operation of the buffer memory should be executed in increments equal to the individual buffer segments. Trying to read multiple buffer segments with one block operation will result in erroneous flag indications.

The following diagram shows the bit pattern for the Multibuffer Memory Control Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	CLR ENA	OVER RUN	FLAG 8	FLAG 7	FLAG 6	FLAG 5	FLAG 4	FLAG 3	FLAG 2	FLAG 1

<31:10> These bits are not used and read as zeros.

<9> CLEAR ENABLE is a write/read bit used to enable/disable the automatic clearing of multibuffer flag bits after a buffer segment is read by the host computer.

<8> OVERRUN is a read/write-to-clear bit used to indicate a buffer overrun condition. This condition is generated when the host computer fails to read a buffer segment before the 3972 tries to fill it with read data. This bit can be cleared by writing a one to this bit location.

<7:0> FLAG 8 through 1 are read/write-to-clear bits used to indicate when a segment of the buffer memory has been filled with CAMAC read data. Setting any of these bits to a one when writing to this register causes the corresponding bit to be reset to zero.

Multibuffer Interval Counter

The Multibuffer Interval Counter is a write/read register used to define the size of individual buffer segments. This register is written with an F(16)A(10) and read with an F(0)A(10).

The Multibuffer Interval Counter specifies the number of 32-bit data entries placed in the memory before a multibuffer flag is set. One to eight buffer segments of equal size may be defined. The size of each segment is determined by the setting in the Multibuffer Interval Counter (MIC). The individual buffer size can be calculated using the following relationship:

Individual Buffer Size = [(Buffer End Address +1) ÷ (Number of Desired Buffers)] -1

Note that the individual buffer size is the same as what should be loaded into the MIC. For example, a total of 64 32-bit entries are to be shared between two buffers. The Multibuffer End Address should be set to one less than the total number of entries ($63 = 3F_{16}$). The individual buffer size becomes:

Individual Buffer Size = $(64 \div 2) - 1 = 1F_{16}$.

The following diagram shows the bit pattern for the Multibuffer Interval Counter.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	BIC 19	BIC 18	BIC 17	BIC 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
BIC 15	BIC 14	BIC 13	BIC 12	BIC 11	BIC 10	BIC 9	BIC 8	BIC 7	BIC 6	BIC 5	BIC 4	BIC 3	BIC 2	BIC 1	BIC 0

<31:20> These bits are not used and read as zeros.

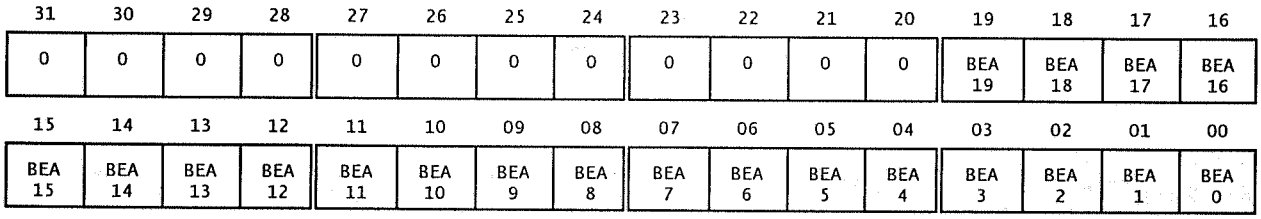
<19:0> BUFFER INTERVAL COUNT 19 through 0 are write/read bits used to specify the interval at which buffer flags are set.

Multibuffer End Address

The Multibuffer End Address is a write/read register used to specify the last buffer memory address to be accessed for storing read data before the buffer address rolls over to zero. This register is written with an F(16)A(11) and read with an F(0)A(11).

The Multibuffer End Address (MEA) is used to define the size of the entire DRAM memory that is to be used for storing multibuffered data. It is not necessary for the 3972 to use the entire memory region. This register must be loaded with the last physical address to be accessed before the address counter is reset to zero and the buffer fills from the beginning.

The following diagram shows the bit layout for the Multibuffer End Address Register.



<31:20> These bits are not used and read as zeros.

<19:0> BUFFER END ADDRESS 19 through 0 are write/read bits used to specify the last location accessed when storing read data before the buffer address is reset to zero.

List Memory Instruction Format

The 3972 contains a 32K x 32 word memory used to hold the list instructions executed by the list processor. The list processor consists of a hardware state machine and a DSP. The hardware processor is used to interpret and perform operations that execute CAMAC cycles directly. When the hardware processor finds an instruction it is not capable of executing, the DSP is started and processes the special instruction.

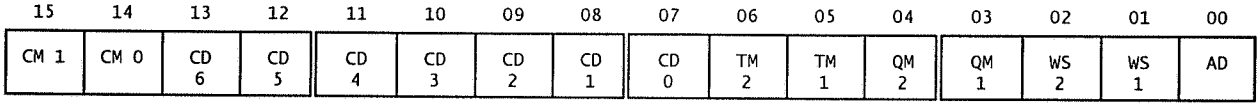
The hardware list processor can execute the three basic instructions. These instructions are as follows:

1.) Single CAMAC Operation
2.) Block CAMAC Operation
3.) Single Inline Write Operation

The remaining instructions are executed by the DSP. Once the DSP has finished interpreting and executing a list instruction, the hardware processor is restarted.

Basic Instruction Format

The list memory is organized as 32-bit words (longwords). The lower 16-bits of the first longword of a command instruction specifies the type of instruction. This 16-bit word is referred to as the instruction header. The following diagram shows the format of the instruction header.



Bits 15 and 14 of the instruction header specify the basic type of instruction as follows:

CM1	CM0	Instruction Type
0	0	CAMAC Transfer
0	1	Reserved
1	0	Special Instruction
1	1	Reserved

Bits 13 through 7 are only used to further resolve special instructions. These bits are not used for CAMAC transfer instructions.

Bits 6 and 5 are used to specify the Transfer Mode used during a CAMAC operation and are defined as follows:

TM2	TM1	Transfer Mode
0	0	Single CAMAC Operation
0	1	Block CAMAC Operation
1	0	Single Inline Write CAMAC Operation
1	1	Reserved

Single CAMAC operations are used to transfer an individual word of CAMAC data to/from a module. Block transfer operations are used to transfer multiple data words to/from a CAMAC module. Both of these transfer modes require transfer of data to/from a data buffer. The data for Single Inline Write operations is contained in the list. This instruction is useful for allowing CAMAC write operations in a CAMAC read list since bidirectional data transfers are not supported during list mode operations.

Bits 4 and 3 are used to specify the Q-Mode to be used for the CAMAC operation and are defined as follows:

QM2	QM1	CAMAC Q-Mode
0	0	Q-Stop Operation
0	1	Q-Ignore Operation
1	0	Q-Repeat Operation
1	1	Q-Scan Operation

Model 3972

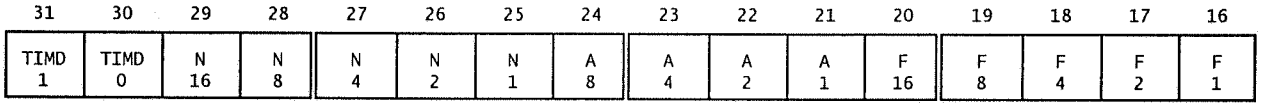
Please refer to the CAMAC Q-Modes section of this manual for additional information on the Q-Modes.

Bits 2 and 1 are used to specify the CAMAC data word size to be used for a transfer and are defined as follows:

WS2	WS1	CAMAC Data Word Size
0	0	24-Bit
0	1	Reserved
1	0	16-Bit
1	1	Reserved

Bit 0 is used to control whether the occurrence of a No-X during a CAMAC operation causes an error to be generated and the list terminated. Setting this bit to a one disables list termination on a No-X condition and setting the bit to a zero enables list termination on a No-X condition. Note that this bit has no effect for Q-Scan modes of operation.

The high 16-bit word of the first 32-bits of a CAMAC data transfer instruction contains the Station Number (N), Subaddress (A) and Function Code (F) to be executed during the addressed CAMAC operation. Two other bits are found in this field which indicate the CAMAC timing parameters used for the operation. These modes include Normal CAMAC Timing, Enhanced CAMAC Timing and Fast CAMAC Timing. The following diagram shows the layout of the high 16-bits of the first 32-bits of a CAMAC instruction.



Bits 31 and 30 are used to specify the timing mode associated with a CAMAC operation. The binary combination of these bits select the various timing selections as follows:

TIMD1	TIMD0	CAMAC Timing Mode
0	0	Normal CAMAC Timing
0	1	Enhanced CAMAC Timing
1	0	Fast CAMAC Timing
1	1	Reserved

Please refer to the CAMAC Timing section of this manual for additional information on these timing modes.

Bits 29 through 25 are used to specify the CAMAC Station Number (N) to be asserted during the addressed CAMAC operation. These bits allow for a N specification in the range of 0 through 31. Station Numbers 1 through 23 are the only valid entries in this field. N(30) is a legal Station Number specification at the host driver but the internal registers of the 3972 cannot be accessed by list processing operations executed by the 3972.

Bits 24 through 21 are used to specify the CAMAC Subaddress (A) to be accessed during the addressed CAMAC operation. Valid Subaddresses range from 0 to 15.

Bits 20 through 16 are used to specify the CAMAC Function Code (F) to be accessed during the addressed CAMAC operation. The binary combination of the F16 and F8 bits determine the type of operation as follows:

F16	F8	Operation
0	0	CAMAC Read Operation
0	1	CAMAC Control Operation
1	0	CAMAC Write Operation
1	1	CAMAC Control Operation

CAMAC Read operations are data transfers from a CAMAC module to the host computer. CAMAC Write operations are data transfers from the host computer to a CAMAC module. CAMAC Control operations are dataless operations. A Dataway cycle is executed but no data transfer takes place.

CAMAC Single Transfer Instructions

The Instruction Header and the CAMAC NAF specification are common for all addressed CAMAC operations executed from the list. This information occupies a complete 32-bit longword is the list memory. For the CAMAC Single Transfer mode, this is the only required entry. For each Single Transfer encountered in the list, a single CAMAC operation is executed. The direction of the transfer is controlled by the CAMAC Function Code specified for the command. To transfer multiple data words to/from a CAMAC module, the Block Transfer instruction must be used.

The following diagram shows the composite format for the CAMAC Single Operation. Note that the TM2 and TM1 bits have been set to zero to indicate a Single Operation.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMD 1	TIMD 0	N 16	N 8	N 4	N 2	N 1	A 8	A 4	A 2	A 1	F 16	F 8	F 4	F 2	F 1
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	0	0	QM 2	QM 1	WS 2	WS 1	AD

CAMAC Block Transfer Instruction

CAMAC Block Transfer instructions allow multiple data words to be transferred to/from a CAMAC module with a single instruction. The first longword of a Block Transfer instruction contains the Instruction Header as previously described. The second longword of a Block Transfer instruction contains a transfer count specification. The transfer count, which is in two's complement format, specifies the maximum number of transfers to execute for the block operation. This transfer count specifies the number of data words to transfer, regardless of the CAMAC data word size.

When the list processor encounters a block transfer instruction, it loads the List Transfer Count Register with the transfer count specification found in the list. The List Transfer Count is then incremented as each data word is transferred to/from the CAMAC module. The Block Transfer continues until the List Transfer Count is incremented to zero or an error occurs. If the transfer terminates prematurely, the List Transfer Count may be read to determine the number of data words remaining to transfer.

Note that this mode is used for CAMAC Read and Write operations only. CAMAC Control commands cannot be executed with this instruction format.

The following diagram shows the composite format for the CAMAC Block Operation. Note that the TM2 bit is set to zero and the TM1 bit is set to a one to indicate a Block Operation.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMD 1	TIMD 0	N 16	N 8	N 4	N 2	N 1	A 8	A 4	A 2	A 1	F 16	F 8	F 4	F 2	F 1
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	0	1	QM 2	QM 1	WS 2	WS 1	AD

2nd Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCR 31	TCR 30	TCR 29	TCR 28	TCR 27	TCR 26	TCR 25	TCR 24	TCR 23	TCR 22	TCR 21	TCR 20	TCR 19	TCR 18	TCR 17	TCR 16
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
TCR 15	TCR 14	TCR 13	TCR 12	TCR 11	TCR 10	TCR 9	TCR 8	TCR 7	TCR 6	TCR 5	TCR 4	TCR 3	TCR 2	TCR 1	TCR 0

CAMAC Single Inline Write Instruction

The CAMAC Single Inline Write Instruction allows a predetermined data word to be placed in the list and written to the specified NAF. Most often, this type of instruction is used for module initialization or as a mechanism for executing a CAMAC write operation within a CAMAC read list. This command may also be used to execute CAMAC control functions. The first longword of a Single Inline Write instruction contains the Instruction Header as previously described. The second longword contains the 24-bits of CAMAC write data.

The following diagram shows the composite format of the Single Inline Write instruction.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMD 1	TIMD 0	N 16	N 8	N 4	N 2	N 1	A 8	A 4	A 2	A 1	F 16	F 8	F 4	F 2	F 1
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	1	0	QM 2	QM 1	WS 2	WS 1	AD

2nd Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	WT 24	WT 23	WT 22	WT 21	WT 20	WT 19	WT 18	WT 17
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
WT 16	WT 15	WT 14	WT 13	WT 12	WT 11	WT 10	WT 9	WT 8	WT 7	WT 6	WT 5	WT 4	WT 3	WT 2	WT 1

If it is necessary to embed a CAMAC Control operation in a list, the Single Inline Write format may be used. Since CAMAC Control operations are dataless, the 2nd longword may be filled with any data pattern. Even though the data portion of the instruction is not used, it must still be included.

Special Instructions

Several 'special' instructions are implemented by the 3972. These instructions are not handled by the hardware list processor but by the DSP. When the hardware list processor encounters any special instructions, it signals the DSP. The DSP then examines the special instruction and executes the operations. Once complete, the DSP turns list processing control back to the hardware machine.

During normal list processing operations the only special instruction that must be used is the HALT. The HALT instruction indicates that list processing should cease. This HALT instruction must be the last list instruction loaded following a valid list.

The following chart shows the special instructions implemented by the 3972. The hex data shown corresponds to the first 16-bit word of the special instruction (instruction header).

Special Instruction (hex)	Function
8000	Halt
8002	Store Time Stamp
8003	Clear Time Stamp
8042	Source Trigger
8043	Set LAM 24
8080	Mark List Address
8081	End Of List
8100	Write Reply Short Word
8101	Write Reply Long Word
8102	Write Demand FIFO

Halt Instruction

The Halt Instruction has a value of 8000_{16} and must be placed at the end of a list sequence. This special instruction informs the list processor to cease processing until retriggered. The following diagram shows the bit pattern for the Halt Instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Store Time Stamp Instruction

The 3972 contains a time stamping mechanism to indicate relative time. The opcode for this instruction is 8002_{16}. This time stamp can be cleared by a list instruction, a highway trigger message, or by a host request. Once the time stamp is cleared, the 24-bit counter is incremented at an interval of 1 microsecond. When the counter is incremented to $FFFFFF_{16}$, it is reset to zero and counting continues. This time stamp counter does not return absolute time but can be used to determine the relative time that elapses between events, as long as the events are not farther apart than 16.77 seconds.

The Store Time Stamp instruction is used to read the value of this counter and store the result. When a list is triggered without using the multibuffer options and this instruction is executed, the time stamp data is written to the highway reply FIFO. Therefore, this instruction can only be used when the list being executed contains read data to be returned to the host. When a list is triggered and multibuffer operations are used, the time stamp data is placed into the multibuffer memory. Care must be taken when using this instruction that room is allocated in the reply buffer for the time stamp data.

The following diagram shows the format of the Store Time Stamp instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Clear Time Stamp Instruction

The Clear Time Stamp instruction is used to clear the counters on the 3972 that generate the time stamp. The opcode for this instruction is 8003_{16}. The format of the Clear Time Stamp instruction is shown in the following diagram.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Source Trigger Instruction

The Source Trigger Instruction has an opcode of 8042_{16} and is used to write data to the internal 3972 Trigger Source Register. This register allows the front panel trigger out LEMO's to be pulsed. Please refer to the Trigger Source Register section of this manual for additional details.

The following diagram shows the format of the Source Trigger instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	RST TSTP	LIST GO	FP TRGA	FP TRGA
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

Bit 19 is set to a one to clear the Time Stamp Counters. This bit provides the same function as the Clear Time Stamp instruction.

Bit 18 is set to a one to initiate list processing. This bit is not used since the list is already running when this instruction is interpreted.

Bit 17 is set to a one to apply a pulse on the front panel Trigger Out B LEMO connector.

Bit 16 is set to a one to apply a pulse on the front panel Trigger Out A LEMO connector.

Set LAM24 Instruction

The set LAM24 instruction has an opcode of 8043_{16} and is used to set the LAM24 bit inside the 3972. The assertion of this signal does not assert the LAM signal on the CAMAC Dataway but does set the internal signal. This instruction can be useful for generating demand messages

from the 3972 or to aid in debugging LAM/Demand problems. The following diagram shows the format of the Set LAM24 instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1

Mark List Address Instruction

The Mark List Address instruction has an opcode of 8080_{16} and is used to save the current List Memory Address (LMA) of address where the Mark List Instruction resides. This value is then used by the End Of List (EOL) Instruction to reload the List Memory Address when the EOL instruction is executed. It is recommended that all lists start with the Mark List Address instruction and terminate with an End Of List instruction. This insures proper operation should a list be retriggered.

The following diagram shows the bit pattern for the Mark List Address instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

End Of List Instruction

The End Of List instruction has an opcode of 8081_{16} and is used to terminate list execution and restore the List Memory Address pointer to the value that was saved after the last execution of a Mark List Address instruction. This provides a mechanism for restoring the List Memory Address pointer for subsequent list triggers.

The following diagram shows the bit pattern for the End Of List instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

Write Reply Short Word Instruction

The Write Reply Short Word instruction has an opcode of 8100_{16} and is used to write a 16-bit data value to the read reply path. The reply path may be either the Highway Reply FIFO or the Multibuffer Memory. If non-multibuffered operations are executed and this instruction is encountered, the 16-bit data word accompanying this instruction is written to the Highway reply FIFO and transmitted back to the host. If the Multibuffer Memory option is installed and enabled, this instruction will cause the 16-bit value to be written to the multibuffer memory. Care must be taken when using this instruction to ensure that room is allocated in the reply path for this data element.

The following diagram shows the format of the Write Reply Short Word instruction.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

2nd Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
RFD 15	RFD 14	RFD 13	RFD 12	RFD 11	RFD 10	RFD 9	RFD 8	RFD 7	RFD 6	RFD 5	RFD 4	RFD 3	RFD 2	RFD 1	RFD 0

Bits 15 through 0 of the second longword contain the data to be written to the reply path.

Write Reply Long Word Instruction

The Write Reply Long Word instruction has an opcode of 8101_{16} and is used to write a 32-bit data value to the read reply path. The reply path may be either the Highway Reply FIFO or the Multibuffer Memory. If non-multibuffered operations are executed and this instruction is encountered, the 32-bit data word accompanying this instruction is written to the Highway reply FIFO and transmitted back to the host. If the Multibuffer Memory option is installed and enabled, this instruction will cause the 32-bit value to be written to the multibuffer memory. Care must be taken when using this instruction to ensure that room is allocated in the reply path for this data element.

The following diagram shows the format of the Write Reply Long Word instruction.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

2nd Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RFD 31	RFD 30	RFD 29	RFD 28	RFD 27	RFD 26	RFD 25	RFD 24	RFD 23	RFD 22	RFD 21	RFD 20	RFD 19	RFD 18	RFD 17	RFD 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RFD 15	RFD 14	RFD 13	RFD 12	RFD 11	RFD 10	RFD 9	RFD 8	RFD 7	RFD 6	RFD 5	RFD 4	RFD 3	RFD 2	RFD 1	RFD 0

Bits 31 through 0 of the second longword contain the data to be written to the reply path.

Write Demand FIFO Instruction

The Write Demand FIFO instruction has an opcode of 8102_{16} and is used to write an 8-bit data pattern to the Demand FIFO on the 3972. This pattern may then be sent back to the host in the form of a demand message, if enabled. Care should be taken when using this instruction to ensure that the pattern written to the Demand FIFO can be differentiated from LAM sources and the multibuffer source.

The following diagram shows the format of the Write Demand FIFO instruction.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0

2nd Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	DMD 7	DMD 6	DMD 5	DMD 4	DMD 3	DMD 2	DMD 1	DMD 0

Bits 7 through 0 of the second longword are the data bits to be written to the Demand FIFO.

List Processing Operation

The 3972 provides a convenient mechanism for executing a series of CAMAC operations through its list processing hardware. These lists of commands may contain various operations including CAMAC operations, branch instructions or trigger instructions. Together, these operations can perform a complex sequence of events without the need for host intervention. With list processing, up to 126 slave nodes may be simultaneously executing unique lists as directed by a single host.

To use list processing on the 3972, the list elements must first be downloaded. This is accomplished by using the List Memory Address Register (LMA) and the List Memory Data Register(LMD). To download a list, the LMA must first be loaded with the initial address for list placement within the memory. This value can range from 0 to $7FFF_{16}$. After the LAM is loaded, the LMD may then be sequentially loaded by executing either multiple single write operations to it or a block transfer with the entire list. After the list has been downloaded, the LMA must then be loaded to point to the starting location of the list for subsequent list triggers.

The contents of any individual list must cause only unidirectional data transfers to/from the host. Lists are referred to as either CAMAC write lists or CAMAC read lists. CAMAC write lists contain only CAMAC write and control operations. CAMAC read lists can contain CAMAC read operations, CAMAC control operations, and CAMAC Single Inline Write operations since these transfers do not require the write data to come from the host computer.

After a write list is triggered, the host computer must supply the write data by executing a write operation to the internal List Write/Read Data Buffer Register. If the write list is triggered from the host computer, it most often is triggered by writing to the List Write/Read Data Buffer Register. If an asynchronous mechanism is used to trigger the list, the host must be informed that write data is necessary for the list processing to proceed. This connection may be established by having the list processor write to the Demand FIFO to generate a demand message. When the host receives this demand message, it must address the 3972 and supply the CAMAC write data.

The same holds true for a read list. After the read list is triggered, the host computer must establish a connection to the 3972 by reading the List Write/Read Data Buffer Register. The only exception to this is when multibuffer operations occur. Since all read data acquired during

a list processing operation with the multibuffer enabled is sent to the multibuffer memory, the connection to the host is not required to execute the read list. Multibuffer operations cannot be used with write lists; but a Single Inline Write operations may be embedded in a read list.

A list that is asynchronously triggered and does not require data transfers to/from the host is executed to completion. Since no data transfers are executed to/from the host, the list processor does not need to make a connection between the list data path and the host computer.

Triggering of the list can come from the following sources:

1.) Accessing the List Write/Read Data Buffer Register -- F(16)A(0) or F(0)A(0)
2.) Receiving the Execute List Command -- F(25)A(0)
3.) Writing the List Memory Address with bit 15 set to a one -- F(17)A(4)
4.) Writing to the Trigger Source Register with bit 2 set to a one -- F(17)A(2)
5.) Receiving an Addressed Trigger highway message with bit 2 set to a one
6.) Receiving a Broadcast Trigger highway message with bit 2 in the Broadcast Trigger Mask Register set to a one.
7.) Using the Automatic Timer expiration with bit 2 in the Timer Control Register set to a one.
8.) The occurrence of a pre-selected LAM source as configured in the List Trigger Source Register.
9.) A pre-selected front panel trigger in pulse as configured in the List Trigger Source Register.
10.) A pre-selected LAM source as configured in the DSP LAM/Trigger Register and the appropriate list in the 3972. (Must be supported by firmware)

List Processing Example

As an example, assume it is desired to read a two channel analog-to-digital converter (ADC) located in slot 6 of a CAMAC crate. 1024 samples of the digitized analog data are to be taken from each channel. The ADC module is accessed by the following commands:

F(2)A(0)	-	Reads the digitized data. Q-response of 1 indicates valid data
F(17)A(0)	-	Selects the channel to read Data = 1 selects channel 1. Data = 2 selects channel 2.
F(24)A(0)	-	Disables ADC conversions.
F(26)A(0)	-	Enables ADC conversions.

For this example application, a F(17)A(0) is first executed with data = 1 to select the desired channel. An F(26)A(0) is then executed to enable conversions. A Standard CAMAC Block Transfer is then executed in the Q-Repeat mode to read the 1024 converted samples. After the samples are read, an F(24)A(0) is executed to disable the conversions. These steps are then repeated for channel number 2.

Model 3972

For this example, the following commands are executed:

Single Inline Write	N(6) F(17) A(0) DATA(1)
Single Inline Write	N(6) F(26) A(0) DATA(not required for control operation)
Block Transfer	N(6) F(2) A(0) TRANSFER COUNT (1024)
Single Inline Write	N(6) F(24) A(0) DATA (not required for control operation)
Single Inline Write	N(6) F(17) A(0) DATA(2)
Single Inline Write	N(6) F(26) A(0) DATA(not required for control operation)
Block Transfer	N(6) F(2) A(0) TRANSFER COUNT (1024)
Single Inline Write	N(6) F(24) A(0) DATA (not required for control operation)

After the list is formed with these parameters, the actual list is as follows:

Instruction Data	Instruction Description
$001101C2_{16}$	N(6) A(0) F(17) Single Inline Write Q-Ignore 24-Bit
00000001_{16}	DATA(1)
$001A01C2_{16}$	N(6) A(0) F(26) Single Inline Write Q-Ignore 24-Bit
00000000_{16}	DATA(0)
$000201B2_{16}$	N(6) A(0) F(2) Block Transfer Q-Repeat 24-Bit
$FFFFFC00_{16}$	Transfer Count (-1024)
$001801C2_{16}$	N(6) A(0) F(24) Single Inline Write Q-Ignore 24-Bit
00000000_{16}	DATA(0)
$001101C2_{16}$	N(6) A(0) F(17) Single Inline Write Q-Ignore 24-Bits
00000002_{16}	DATA(2)
$001A01C2_{16}$	N(6) A(0) F(26) Single Inline Write Q-Ignore 24-Bits
00000000_{16}	DATA(0)
$000201B2_{16}$	N(6) A(0) F(2) Block Transfer Q-Repeat 24-Bits
$FFFFFC00_{16}$	Transfer Count (-1024)
$001801C2_{16}$	N(6) A(0) F(24) Single Inline Write Q-Ignore 24-Bits
00000000_{16}	DATA(0)
00008000_{16}	HALT

Timer Initiated List Processing

The 3972 provides an operating mode whereby a list of commands can be repetitively executed at a predetermined rate. This rate is referred to as the Tic Rate. The rate can range from a minimum of 80 microseconds to a maximum of 16.77 seconds.

The Timer Data Register and the Timer Control Register are used to control the Tic Rate. The timer rate is controlled through the Timer Data Register. The Timer Control Register is used to specify the signals to be asserted once the timer expires. To enable the list to be triggered from the timer, the LIST GO bit in the Timer Control Register must be set to a one. After the list is loaded and the timer setup is complete, the timer initiated list is enabled by starting the timer through the Control/Status Register.

The format of the timer initiated list is similar to what has been described for synchronous list triggering with one exception. If a single list of instructions is to be executed at the Tic Rate, the Mark List Address instruction and the End-Of-List instructions must be used. The Mark List Address instruction is the first element in the list and informs the list processor to save the List Memory Address of the location of the instruction. Following the Mark List Address instruction is the CAMAC and special instructions to be executed. The last element in the list is the End-Of-List instruction. When the list processor encounters this instruction, the List Memory Address is loaded with the value that was saved when the last Mark List Address instruction was processed. The list processor is then halted and awaits a subsequent trigger.

Timer Initiated List Processing Example

As a simple example to demonstrate Timer Initiated List Processing, assume a list that resides at list memory location 100_{16} and consists of two Single Inline Write operations. This first single inline is a write to N(1) F(16) A(0) with data of 123456_{16} and the second inline is a write to N(2) F(16) A(0) with data of $ABCDEF_{16}$. The timer initiated list would appear as follows:

List Memory Address	List Memory Data	Instruction Description
100_{16}	00008080_{16}	Mark List Address Instruction
101_{16}	02100040_{16}	N(1) A(0) F(16) Single Inline Write Q-Ignore 24-Bit
102_{16}	00123456_{16}	DATA(123456_{16})
103_{16}	04100040_{16}	N(2) A(0) F(16) Single Inline Write Q-Ignore 24-Bit
104_{16}	$00ABCDEF_{16}$	DATA ($ABCDEF_{16}$)
105_{16}	00008081_{16}	End Of List

Notice that the first instruction in the list is the Mark List Address. When the list processor encounters this instruction the current value of the List Memory Address Register (100_{16}) is

saved to be used when the End Of List instruction is found. After the two single inline write operations occur, the list processor finds the End Of List instruction, reloads the List Memory Address Register with the saved value (100_{16}), and ceases execution. Subsequent triggers for list execution will already have the correct list pointer loaded.

CAMAC Q-Modes

The 3972 provides four Q-Modes available for accessing a CAMAC module. These modes include Q-Stop, Q-Ignore, Q-Repeat and Q-Scan. These modes are useful for performing CAMAC operations based on the CAMAC Q and X-responses received. These modes may be used for any CAMAC operation, single transfer and block transfer alike. The transfers may be executed by a host computer or by the list processor on the 3972. The following sections detail the four Q-Modes.

CAMAC Q-Stop Mode

The CAMAC Q-Stop mode is selected by setting the QM2 and QM1 bits to a zero when executing an operation. During Q-Stop operations, the command (NAF) specified is repeated until a CAMAC Q-response of zero is received or the transfer count is exhausted. This provides an efficient mechanism to transfer data to/from a CAMAC module without knowing the amount of data available within the module. In these cases, the transfer count specification is set large enough to transfer the entire data buffer and the transfer will terminate if a No-Q condition is found or the transfer count is exhausted. When a No-Q is returned and the operation terminates, the List Transfer Count Register can be read to determine the number of data words not transferred.

In the case of Single Transfer Operations, the NAF is executed once and the status set accordingly. If a Q-response of zero is returned, the operation terminates in an error. If a Q-response of one is returned, the operation was successful and no error indication is set.

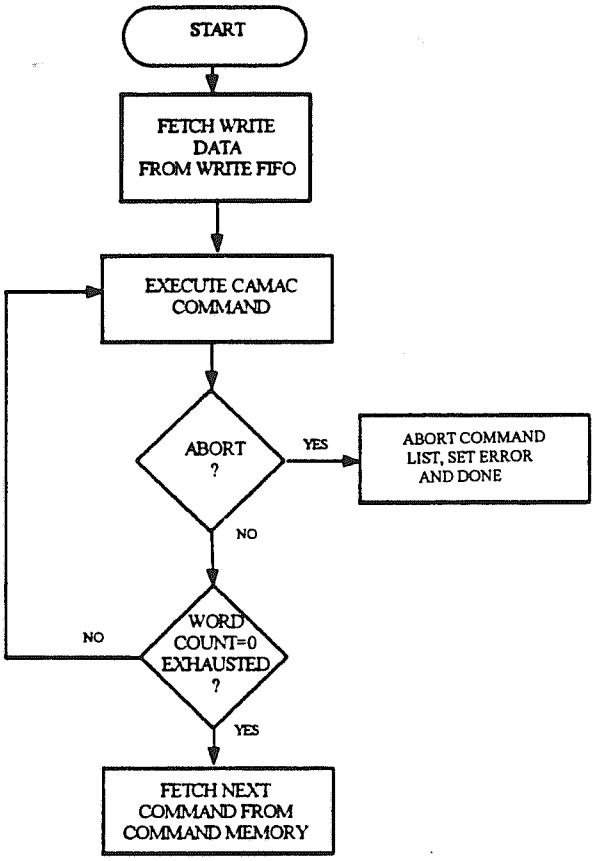
For Q-Stop transfer modes, an error is described as follows:

ERROR = NO-Q
# NO-X * !AD

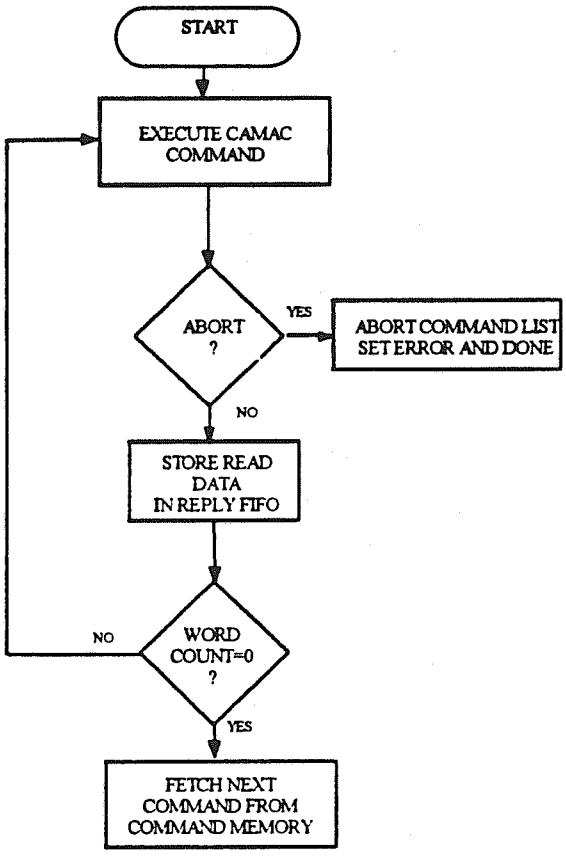
Model 3972

The following is a simplified flow diagram showing the sequences the 3972 uses to execute Q-Stop write and read transfers.

Q-STOP/Q-IGNORE FLOW DIAGRAM



IGNORE - Q WRITE COMMANDS



IGNORE - Q READ COMMANDS

CAMAC Q-Ignore Mode

The CAMAC Q-Ignore mode is selected by setting the QM2 bit to a zero and the QM1 bit to a one when executing an operation. During Q-Ignore operations, the command (NAF) specified is repeated for each data word until the transfer count is exhausted. This mode ignores the Q-response from the previous cycle to determine if the operation should terminate. A block operation will terminate if an error is found.

When executing Single Transfer Operations, the NAF is executed once and the status set accordingly. The Q-response is ignored for a status response but a X-response of zero with Abort Enabled causes an error to be generated.

For Q-Ignore transfer modes, an error is described as follows:

ERROR = NO-X * !AD

The flow diagram on page 48 shows the sequences the 3972 used to execute Q-Ignore write and read transfers.

CAMAC Q-Repeat Mode

The CAMAC Q-Repeat mode is selected by setting the QM2 bit to a one and the QM1 bit to a zero when executing an operation. During Q-Repeat operations, the command (NAF) is repeated for each data word until a Q-response of one is received. A Q-response of one either causes new write data to be fetched or read data to be stored. The command is repeated for each data word until the transfer count is exhausted or an error occurs. If a Q-response is not received before the strap-selectable timeout period, the transfer terminates with a Q-repeat Timeout.

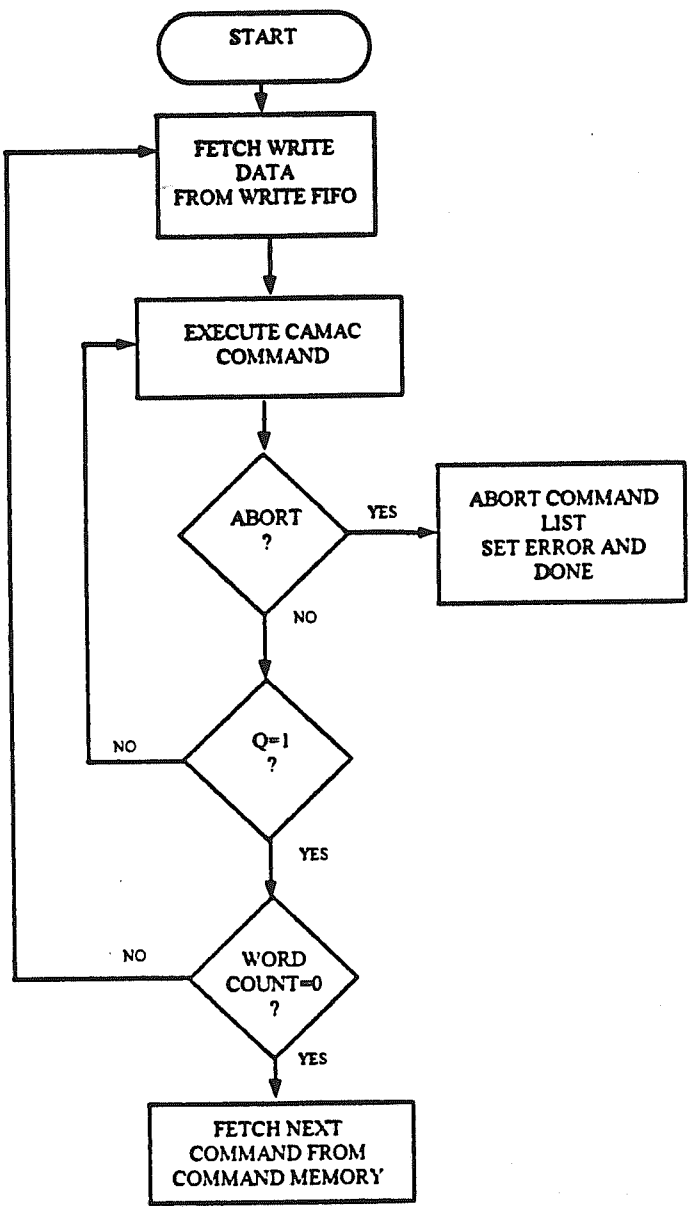
When executing Single Transfers Operations, the NAF is repeated until a Q-response of one is received for only one data word.

For Q-Repeat transfer modes, an error is described as follows:

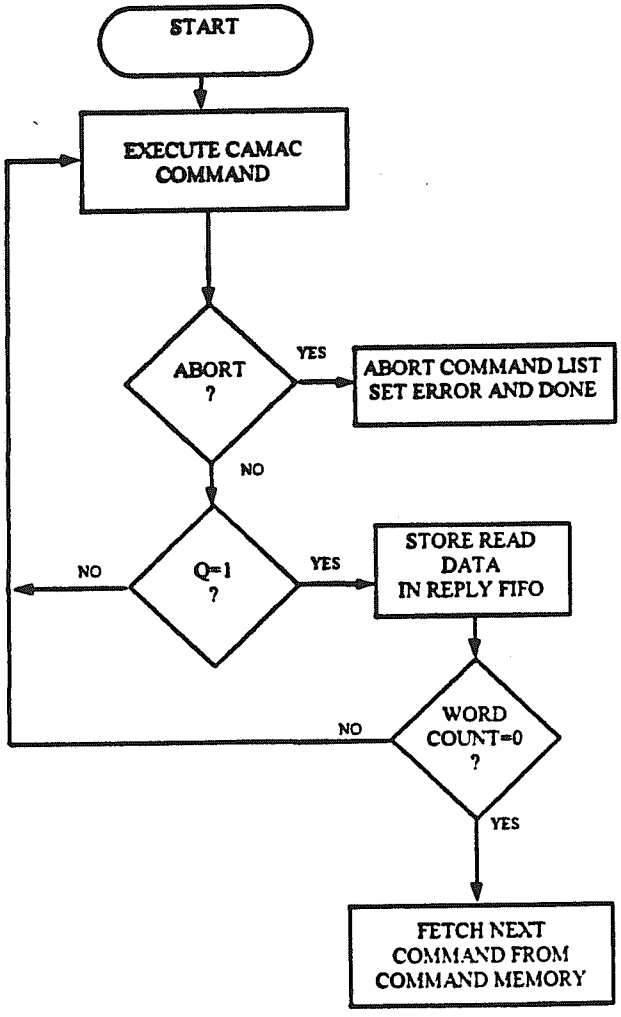
ERROR = Q-REPEAT TIMEOUT
# NO-X & !AD

The following is a simplified flow diagram showing the sequences the 3972 uses to execute Q-Repeat write and read operations.

Q-REPEAT FLOW DIAGRAM



Q-REPEAT WRITE COMMANDS



Q -REPEAT READ COMMANDS

CAMAC Q-Scan Mode

The CAMAC Q-Scan mode is selected by setting both the QM2 and QM1 bits to a one when executing an operation. During Q-Scan operations, the 3972 uses the Q-response from the previous cycle to determine the Station Number (N) and Subaddress (A) for the next operation in the block. A Q-response of zero indicates that the last valid Subaddress of the current Station Number has been accessed. The 3972 responds to the Q=0 response by resetting the Subaddress and incrementing the Station Number. A Q=1 response indicates that the last operation was executed to a valid Subaddress. The controller responds to the Q=1 by either storing the read data of fetching additional write data. After the Q=1 is received, the controller updates the CAMAC address as follows: the Subaddress is incremented or, if it was 15, it is reset to zero and the Station Number incremented.

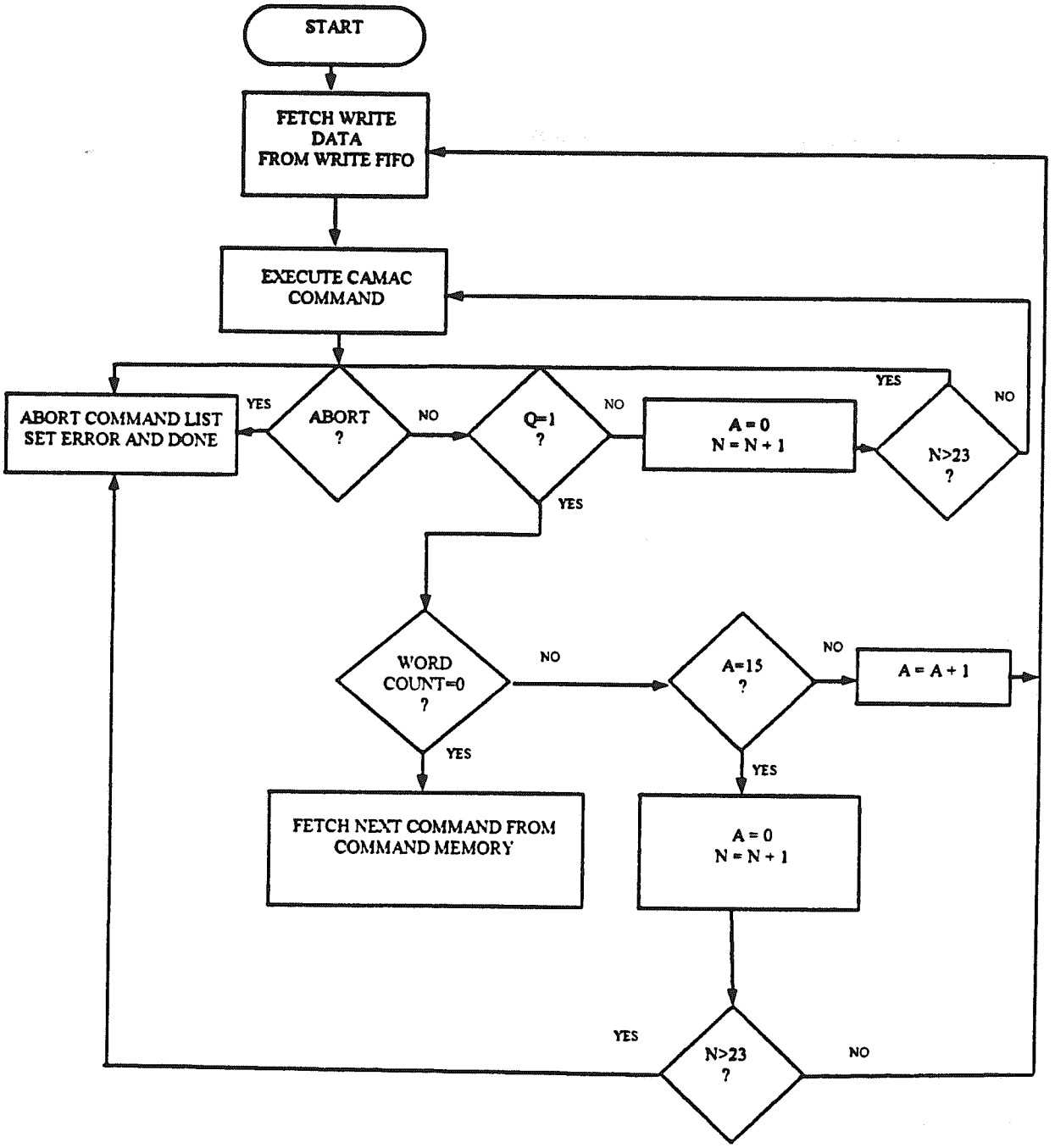
If, due to a programming error or a module failure, the Station Number is incremented beyond Station Number 23, the transfer is terminated and an error condition reported. The following equation describes error during Q-Scan operations.

ERROR = N>23

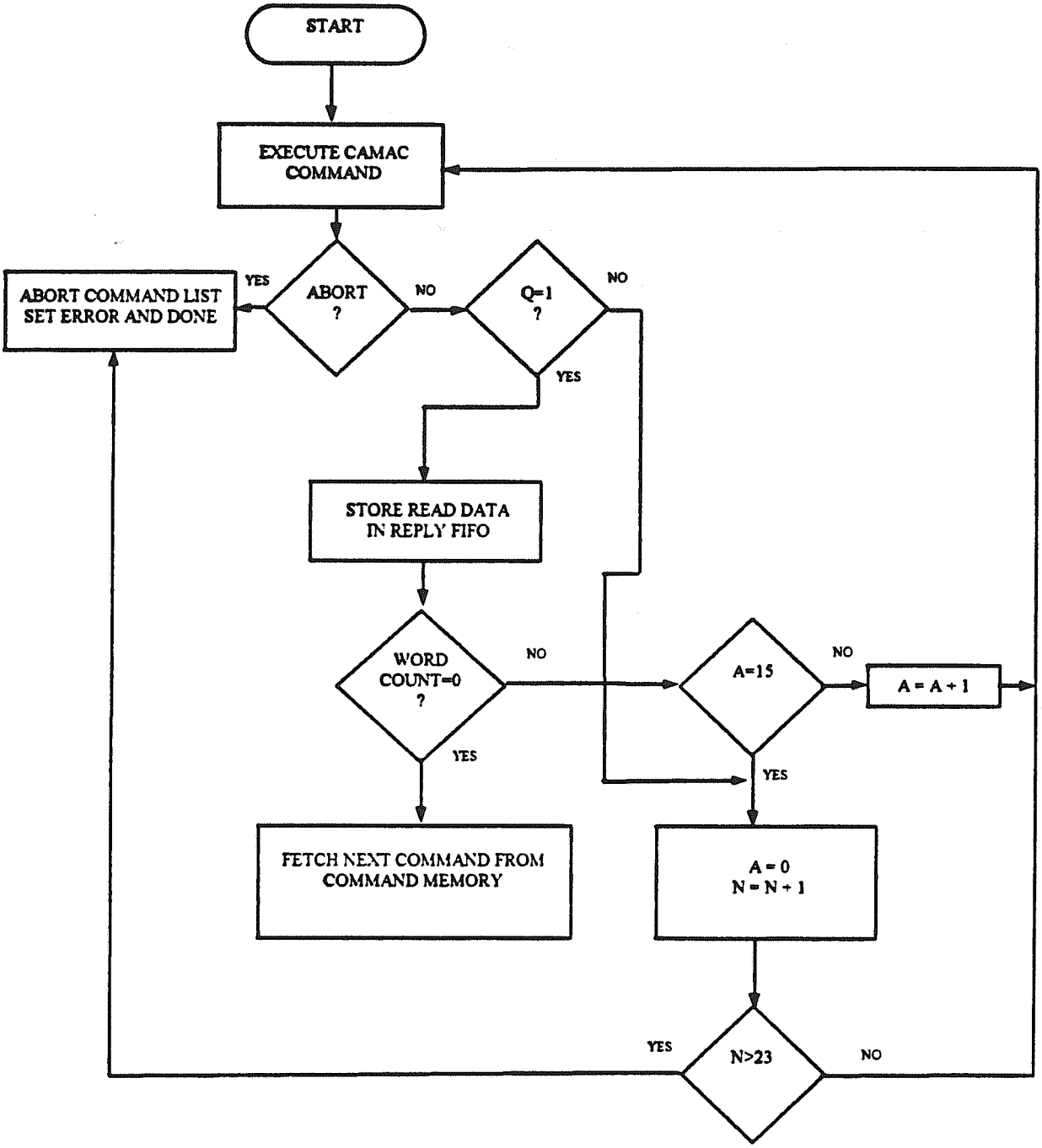
Model 3972

The following diagrams show the basic flow the 3972 uses to execute Q-Scan write and read operations.

Q-SCAN WRITE FLOW DIAGRAM



Q-SCAN READ FLOW DIAGRAM



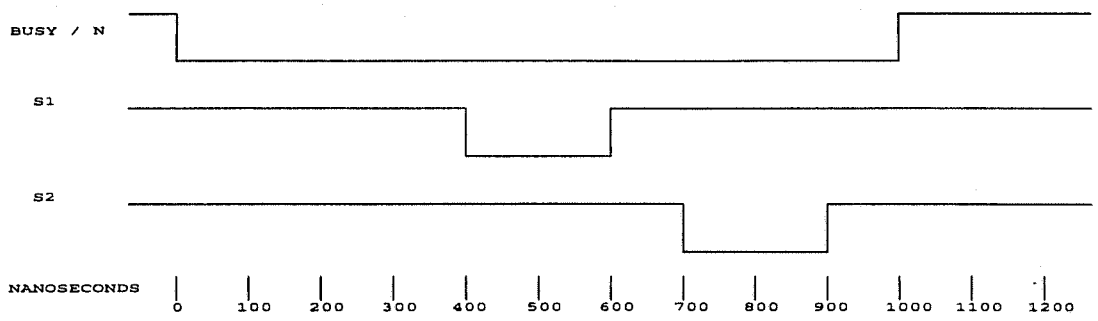
CAMAC Dataway Cycle Timing Modes

The 3972 provides three CAMAC Dataway Timing Modes. These modes consist of normal CAMAC timing, enhanced CAMAC timing and fast CAMAC timing. These three modes are selected by the binary combination of bits 15 and 14 of the CAMAC NAF specification, which correspond to bit locations 31 and 30 of a list instruction. The mnemonics for these bits are TIMD1 and TIMD0 and have the following definitions:

TIMD1	TIMD0	CAMAC Timing Mode
0	0	Normal CAMAC Timing
0	1	Enhanced CAMAC Timing
1	0	Fast CAMAC Timing
1	1	Reserved

Normal CAMAC timing consists of the minimum CAMAC timing parameters. All modules manufactured are able to respond to this timing mode. If there is any question of a modules' ability to respond in the other timing modes, this one should be selected. The Dataway cycle is 1 microsecond long and is shown in the following diagram.

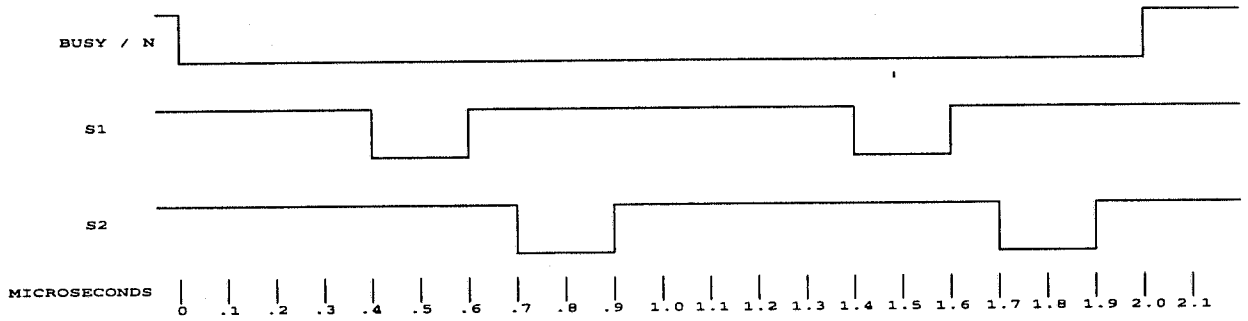
NORMAL CAMAC TIMING DIAGRAM



Model 3972

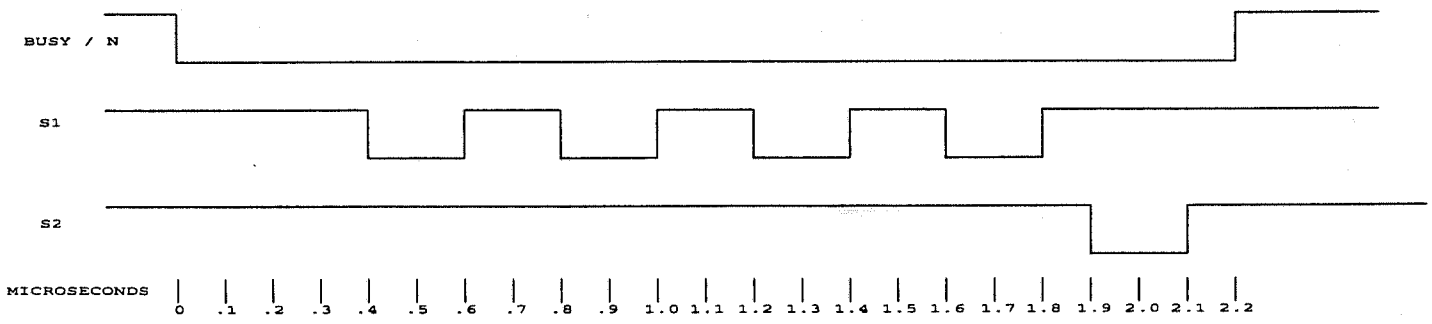
Each Dataway cycle executed by the 3972 enters the Dataway arbitration phase before it is executed. The Enhanced CAMAC timing arbitrates only once for the Dataway before the cycle(s) are executed. Also, the Station Number signal (N) does not make any transitions during a block transfer to a module. Care must be taken when using this timing mode since not all CAMAC modules manufactured can respond to enhanced timing. The following diagram shows an Enhanced CAMAC timing cycle.

ENHANCED CAMAC TIMING DIAGRAM



Fast CAMAC is a recent development and has not been implemented on many modules. This timing mode can provide a 2.5 times increase in throughput when large block transfers are executed. Only CAMAC read operations may be executed using this timing mode. Selecting this timing mode for CAMAC write operations will result in erroneous behavior. During fast mode operations, the Dataway arbitration occurs only once at the beginning of the block cycle. Once arbitration is complete, the Dataway cycle starts as any other cycle. After Strobe S1 is asserted for 200 nanoseconds, it is then negated for 200 nanoseconds and then reasserted. This results in S1 strobe signals occurring at an interval of 400 nanoseconds. After the transfer count has expired for the block operation, the cycle terminates as usual with the Strobe S2 and the negation of N and BUSY. The following diagram illustrates a Fast CAMAC timing sequence.

FAST CAMAC TIMING DIAGRAM



Multibuffer Memory Option

The 3972 can be ordered with a Multibuffer Memory option which contains up to 4 megabytes of memory used to hold CAMAC read data. This memory allows the 3972 to acquire CAMAC read data and buffer it before the host computer must intervene and read the data. Multibuffering is a scheme where a number of buffers of equal size are defined. The 3972 allows for 1 to 8 buffers to be allocated ranging in size from 4 bytes to 4 megabytes.

The total size of a memory buffer can be calculated by the following expression:

Total Buffer Size = [(Individual Buffer Size) x (Number of Desired Buffer Segments)] -1

The Total Buffer Size calculated above is actually a value that represents the last physical address of the buffer memory used for storage before the memory address counter rolls over to zero. This value is then loaded into the Buffer End Address Register to establish the end of the circular buffer. Initially, the received read data from a CAMAC operation is stored at the beginning of the buffer (buffer address 0). As additional read data words are received, sequential memory locations of the buffer are filled. When the buffer address reaches the Buffer End Address specification, it is reset to zero, moving the pointer to the beginning of the buffer.

The Buffer Interval Counter serves a dual purpose. It is used to indicate when individual buffer segments are filled and also prevents the circular buffer from inadvertently overwriting acquired data. As CAMAC read data is stored, the Buffer Interval Counter is decremented. When the counter reaches zero, signifying the end of a buffer segment, it sets a multibuffer flag. These flags are monitored through the Multibuffer Control Register. Once a flag is set, the Buffer Interval Counter reloads itself to prepare for counting data words stored in the next segment. This process will continue until the Buffer End Address is reached. The circular buffer then rolls over to zero to point to the beginning of the buffer. It is critical to have the Buffer End Address roll over at the same time that the last multibuffer flag is set. If this is not the case, the Buffer Interval Counter will not correctly represent the number of words stored and set flags at the appropriate times. This can be prevented by using the relationship previously described. The Number of Desired Buffer Segments used in the previous calculation is the value placed in the Buffer Interval Counter.

The multibuffer flags can be used by the host in two ways. First, the multibuffer flag bits may be polled by reading the Multibuffer Control Register. The second option is to enable the multibuffer flags to generate a demand message. Demand messages are generated by the 3972 to inform the host of an asynchronous event. After the host computer has determined that a multibuffer segment is full, it may then read out that segment of the multibuffer. After the segment has been read, the host computer must then execute a write operation to the Multibuffer Control Register to clear the full flag of the segment that was just read. Optionally, the flag bit may be reset automatically when the segment is read. The Clear Enable bit in the Multibuffer Control Register is set to a one to enable this feature. The only restriction imposed when using the auto-clear feature is that the entire buffer segment must be read out for each setting of a flag.

When reading the Multibuffer Memory for the first time, the Multibuffer Memory Address must be set to zero to point to the beginning of the buffer. Subsequent read operations cause the memory address to automatically increment. After the entire buffer (all segments) have been read, the Multibuffer Memory Address must again be reset to zero.

Multibuffer Data Storage

The multibuffer memory is organized in 32-bit data words. Each 24-bit CAMAC data word occupies one full 32-bit entry in the memory. Two 16-bit CAMAC data words can also be held in one 32-bit memory word. When a list operation causes a switch from 16-bit data words to 24-bit data words, the memory performs an autoalignment. This is done to insure that all 24-bit data words in the memory are longword aligned. If an even number of 16-bit data words were transferred before switching to 24-bit data words, no alignment is required. The following diagram shows the storage of 16-bit data words, 24-bit data words, and a switch from 16-bit to 24-bit data words that required autoalignment.

<table>
<tr><th>Data Bits 31-24</th><th>Data Bits 23-16</th><th>Data Bits 15-8</th><th>Data Bits 7-0</th></tr>
<tr><td>00000000</td><td colspan="3">24-Bit CAMAC Data Word #1</td></tr>
<tr><td>00000000</td><td colspan="3">24-Bit CAMAC Data Word #2</td></tr>
<tr><td colspan="2">16-Bit CAMAC Data Word #2</td><td colspan="2">16-Bit CAMAC Data Word #1</td></tr>
<tr><td colspan="2">16-Bit CAMAC Data Word #4</td><td colspan="2">16-Bit CAMAC Data Word #3</td></tr>
<tr><td>00000000</td><td>00000000</td><td colspan="2">16-Bit CAMAC Data Word #5</td></tr>
<tr><td>00000000</td><td colspan="3">24-Bit CAMAC Data Word #3</td></tr>
<tr><td colspan="2">16-Bit CAMAC Data Word #7</td><td colspan="2">16-Bit CAMAC Data Word #6</td></tr>
<tr><td>00000000</td><td colspan="3">24-Bit CAMAC Data Word #4</td></tr>
</table>

Autoalignment is also executed when a list processing operation terminates and an odd number of 16-bit data words have been transferred. This insures that each list iteration starts on a longword boundary.

Note: When using the special list instructions to write a data pattern to the multibuffer memory, use only the Write Reply Longword instruction. The shortword version of this instruction does not supply enough data to fill an entire 32-bit word of the memory.

Demands

The 3972 contains a 2048 word deep FIFO used for storing entries for demand sources. These entries may be allowed to generate highway demand messages if enabled. Demand sources can be generated from any of the following sources:

1.) A CAMAC LAM
2.) A Multibuffer Segment being filled with data
3.) The List Processor (DSP) writing to the Demand FIFO.

Before a CAMAC LAM source is stored in the demand FIFO it must first be enabled in the Demand LAM Mask Register. This register is used to enable the individual LAM bits into the FIFO. The global LAM enable must also be set to allow LAM sources to be placed in the FIFO. This enable is the LAM Demand Source Enable bit located in the Control/Status Register.

LAMs are priority encoded before they are entered into the FIFO. The lowest priority LAM is LAM1, which corresponds to slot 1, and the highest priority LAM is LAM24. Once the LAMs are prioritized, a entry for each pending LAM is made into the demand FIFO. The values that are entered for each LAM is the slot number that generated it minus one. For example, slot 5 LAM source has a value of 4. Care should be taken when populating a CAMAC crate to ensure that the frequency of demand sources from higher numbered slots does not prevent lower numbered slots from being serviced.

Another demand source is from the multibuffer memory option. The multibuffer memory demand source is enabled using the Multibuffer Demand Source Enable bit located in the Control/Status Register. The value that is entered into the FIFO for this source is 32 (20_{16}). This source is generated during multibuffer operations when the Buffer Interval Counter is decremented to zero indicating that a segment of the buffer may be read. Please refer to the Multibuffer Memory Option section of this manual for additional information.

The remaining source for demands is the List Processor (DSP). A list instruction exists that allows the DSP to write a data pattern into the demand FIFO. This is useful for passing list generated demand messages to the host computer. Care should be taken when allowing the DSP to write to the demand FIFO to ensure that a unique pattern is written so that the host can differentiate this source from CAMAC LAMs and the multibuffer source. Since this demand source is not asynchronous, there is not enable bit in the Control/Status Register for this source.

After entries are placed into the demand FIFO, the may either be read by the host computer or transmitted to the host interfaces' demand FIFO in the form of a highway demand message. These demand messages are enabled with the Demand Message Enable bit in the Control/Status Register. If demands are not enabled onto the highway, they may be retrieved by reading the Demand FIFO Register. The Control/Status Register contains a Demand Pending bit that is set as long as there is at least one demand pending. This bit should be consulted after reading the Demand FIFO Register to see if additional entries are available for readout.

Highway Triggers

The 3972 can be caused to trigger events when the host driver generates either a Addressed Trigger command or a Broadcast Trigger command. The Addressed Trigger highway command is sourced by a highway driver and is addressed to an individual slave node. Broadcast Trigger highway commands are sourced by a highway driver and are acted upon by all slave nodes on the highway.

The Addressed Slave Trigger is another way that the Trigger Source Register of the 3972 may be written. This command is useful from the perspective of a host driver since this command does not have to address individual registers within a slave node. The Trigger Source Register of a 3972 CAMAC Crate Controller or a V160 VXI Slot 0 Controller can be written with this command. The data associated with this highway operation is applied to the Trigger Source Register and can reset the time stamp, initiate a list processing operation, or source one of the front panel trigger out signals. Please refer to the Trigger Source Register section of this manual for the bit locations of each function in the register.

The Broadcast Trigger highway command is sourced by the host driver and is used to trigger multiple events in multiple slaves with one command. For example, this command allows all slave nodes on the highway to initiate list execution with one highway message. The actual events triggered in a given slave depends on the contents of the Broadcast Trigger Mask Register on each device. For the 3972, these events include clearing the time stamp, initiating list processing, and sourcing a front panel trigger out signal. When the dataless Broadcast Trigger message is received, the Broadcast Trigger Mask Register pattern is applied to the Trigger Source Register. Any bit location that is set to a one causes the corresponding event to be generated. Please refer to the Broadcast Trigger Register and the Trigger Source Register descriptions for the bit layout of each register.

By default, when the 3972 receives a Broadcast Trigger message, it immediately applies the Broadcast Trigger Mask pattern to the Trigger Source Register. This function can be delayed in each slave node by using the Delay Count Register. This register, when enabled, can delay the application of the mask pattern to the source register. This delay is used to enable nearly synchronous triggering. The Delay Count specifies the number of 200 nanosecond increments to delay from receiving the Broadcast Trigger message until the application of the mask data.

One method to determine the setting for the individual slave delay counts would be to initially use the Broadcast Trigger message to have each slave source its front panel trigger output signals. A measurement could then be taken to determine the actual delay count values to load into each delay count register. Another method involves use of the time stamp on each slave node and the multibuffer memory. A small list is downloaded to each slave node that reads the time stamp and place it in the multibuffer memory. The lists in each slave device would then be triggered by a Broadcast Trigger message. The time stamp data is then written into the multibuffer memory on each slave device that the host could read after all the lists are complete. The difference in these numbers would then reflect the time difference in between each slaves' response time.

Front Panel LEDs, Connectors and Switches

The 3972 front panel contains 10 status LEDs, 7 LEMO connections, 2 switches, 2 fiber-optic connectors and a backup power jack. The LEDs are used to reflect various status signals on the 3972. The following is a description of each LED.

SYNC is a green LED that is illuminated when the 3972 is receiving a synchronization message from the host or is involved in a current highway transaction. This signal is merely a static representation of highway sync. During large block transfers executed by a host driver, this LED may not be illuminated since any unaddressed slave during the block transfer is not receiving the sync message or involved in a highway transaction.

ADDRESS RECEIVED is a green LED that is illuminated when the 3972 is addressed by a highway driver.

DEMAND is a red LED that is illuminated as long as at least one demand entry is contained in the demand FIFO.

PARITY ERROR is a red LED that is illuminated when the 3972 detects a parity error on the incoming data stream.

LIST BUSY is a green LED that is illuminated as long as the 3972 is involved in an active list processing operation.

INHIBIT is a red LED that is illuminated as long as the CAMAC Dataway INHIBIT signal is asserted.

DATAWAY BUSY is a green LED that is illuminated when the 3972 is executing CAMAC Dataway cycles.

ERROR is a red LED that is illuminated when a 3972 CAMAC operation results in an error condition. Please refer to the CAMAC Q-Mode section of this manual for error descriptions.

NO-Q is a red LED that is illuminated after a CAMAC cycle is executed and a Q-response of zero was returned.

NO-X is a red LED that is illuminated after a CAMAC cycle is executed and a X-response of zero was returned.

Three additional LEDs are present on 3972 controllers that have the multibuffer option attached. The following describes these LEDs.

MULTIBUFFER WRITE is a green LED that is illuminated as long as data is being transferred into the multibuffer memory.

MULTIBUFFER READ is a green LED that is illuminated as long as data is being read from the multibuffer memory by the host computer.

MULTIBUFFER ERROR is a red LED that is illuminated when an overrun condition was posted by the multibuffer circuitry. This occurs when the host computer cannot read out the memory faster than it is being filled. Please refer to the Multibuffer Memory section of this manual for additional information.

Seven LEMO connectors are located on the front of the 3972. Four of these connectors are used as trigger input/output connectors and the remaining three are used to configure CAMAC Dataway arbitration.

The two front panel LEMO connectors labelled TRIGGER OUT B and TRIGGER OUT A are used to source output pulses to external devices. These signals are asserted when the Trigger Source Register is written by the host computer or a trigger highway command is received. The outputs are TTL open-collector outputs and are driven by 74F38 gates. When a signal is asserted on these connectors, a low going pulse of approximately 1 microsecond is generated. Each of the driving gates has a strap selection that allows a 1 Kohm resistor to +5 volts to be connected to the output. Please refer to the Strap Options section of this manual for additional information.

The two front panel connectors labelled TRIGGER IN B and TRIGGER IN A are used to initiate list processing operations on the 3972. Before these signals are allowed to start a list processing operation, they must be enabled in the List Trigger Source Register. These trigger sources are enabled by setting the corresponding trigger in enable bit to a one. These low-true input signals must be TTL and have a minimum pulse width of 200 nanoseconds. The inputs have a 1 Kohm resistor to +5 volts attached and are fed in to a 74LS14 gate.

The three LEMO connectors that are used to configure the Dataway arbitration are labelled REQUEST, GRANT IN and GRANT OUT. For a crate controller that occupies slots 24 and 25, the REQUEST LEMO is jumpered to the GRANT IN connector. This establishes the main crate controller as the arbiter. The GRANT OUT LEMO of the main crate controller would then be connected to the GRANT IN LEMO of the next highest priority auxiliary controller in the crate. For a crate that has only one CAMAC controller, the GRANT OUT LEMO remains unconnected.

Towards the top of the front panel is the Node Address switch. This switch is used to establish the 3972's node address on the highway. Valid highway addresses range from 1 to $7F_{16}$. All other node addresses are reserved. The two switches are hexadecimal switches. The switch on the left hand side is the most significant digit and the one on the right is the least significant. To increment a switch digit, depress the small tab under the number display. To decrement the switch digit, depress the small tab above the number display.

The Z/C switch is used to manually execute a CAMAC Initialize (Z) cycle or a CAMAC Clear (C) cycle. This is a momentary return-to-center switch. To execute a CAMAC Initialize operation, momentarily press the switch handle down. To execute a CAMAC Clear operation, momentarily lift the switch handle up.

The two fiber-optic connections on the front panel are used to bring the highway into and out of the 3972. These two connectors are ST type connectors which incorporate a push-and-turn mechanism to hold the fiber-optic cable onto the 3972. The top connector is the Highway In and the lower connector is the Highway Out.

The Backup connector, located towards the bottom of the front panel, is used to provide a battery back-up connection for the 3972. This is used to prevent the loss of highway integrity in case of a power failure in the CAMAC crate. When used in conjunction with a fiber-optic switch, this connector serves as an output the switch may use to detect a loss in mainframe power and re-route the highway around the 3972. In addition, the backup connector becomes the power input if battery backup is provided. The 3972 may use the battery power to energize the fiber-optic front end circuitry necessary to keep the highway intact.

Rear Panel Connectors

The 3972 contains two rear panel mounted connectors. The connector on the back of the left-most card (A-board) is a 10 position header. This connector is used as a general purpose RS232 serial interface to the 3972. The serial port is operated under control of the DSP and is useful for executing CAMAC operations through the console. These operations must be executed while there are no operations occurring in the 3972. The following diagram shows the pin allocations of this connector.

Pin Number	Pin Description
1	Transmit Channel A
2	Receive Channel A
3	Ground
4	Transmit Channel B
5	Receive Channel B
6	Ground
7-10	No Internal Connection

The rear panel mounted connector on the right-most card (B-Board) is a 40 position header and is used to bus the Auxiliary Controller Bus (ACB) to other controllers in a CAMAC crate.

Model 3972

The following chart shows the allocation of signals on the connector.

Pin #	Signal	Pin #	Signal
1	Ground	2	Encoded N1
3	Encoded N2	4	Encoded N4
5	Encoded N8	6	Encoded N16
7	Ground	8	Auxiliary Controller Lockout
9	Ground	10	Conditionally Free
11	Ground	12	Request
13	Ground	14	Request Inhibit
15	Ground	16	AL1
17	AL2	18	AL3
19	AL4	20	AL5
21	AL6	22	AL7
23	AL8	24	AL9
25	AL10	26	AL11
27	AL12	28	AL13
29	AL14	30	AL15
31	AL16	32	AL17
33	AL18	34	AL19
35	AL20	36	AL21
37	AL22	38	AL23
39	AL24	40	Ground