Model 2961

VME Interconnect Highway Driver

Instruction Manual

October, 1997

*****Special Option*****

Model 2961-S001

VME Interconnect Host Adapter

February, 1997

Special Option

Model 2961-S001

The Model 2961-S001 is the same as the Model 2961-Z1A except that it has 2 Wide VXI front panel.

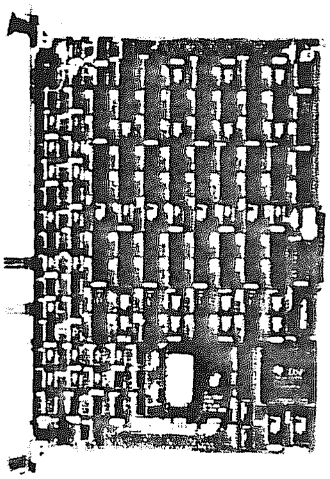
CONTENTS

VME Host Adapter for the Grand Interconnect

Allows a computer VME bus to host the Grand Interconnect **2961**

Features

- **Provides a VME interface for the Grand Interconnect highway**
- **Can be used with any combination of VXI or CAMAC chassis**
- **Provides access to 126 nodes**
- **Uses fiber-optic highway transmission**
- **Provides full throughput with up to two kilometers distance between fiber-optic nodes**
- **Exhibits 10 Mbytes/s highway throughput and 15 Mbytes/s VME DMA burst rate**
- **Includes a high-speed command processor**
- **Includes multi-buffer functionality**



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

The Grand Interconnect (GI) connects multiple I/O chassis to a host computer. It provides high-throughput, deterministic data acquisition along with control capabilities. A complete GI system includes an Interconnect Host Adapter (IHA), a fiber-optic highway, and up to 126 I/O nodes that may include either or both VXI Slot-0 controllers and CAMAC crate controllers. The system supports distances between nodes of up to 2 kilometers (6560 ft).

The 2961 Interconnect Host Adapter is a double-width 6U VME card which provides the interface between the VME bus and the Grand Interconnect highway. Communication between this IHA and the host computer is via a 32-bit data path that is used to configure the system and initiate highway operations. A DMA (direct memory access) mechanism is incorporated to transfer data over the highway using 32-bit data transfers. The 2961 IHA supports VME DMA transfers, yielding a 15 Mbytes/s burst transfer rate.

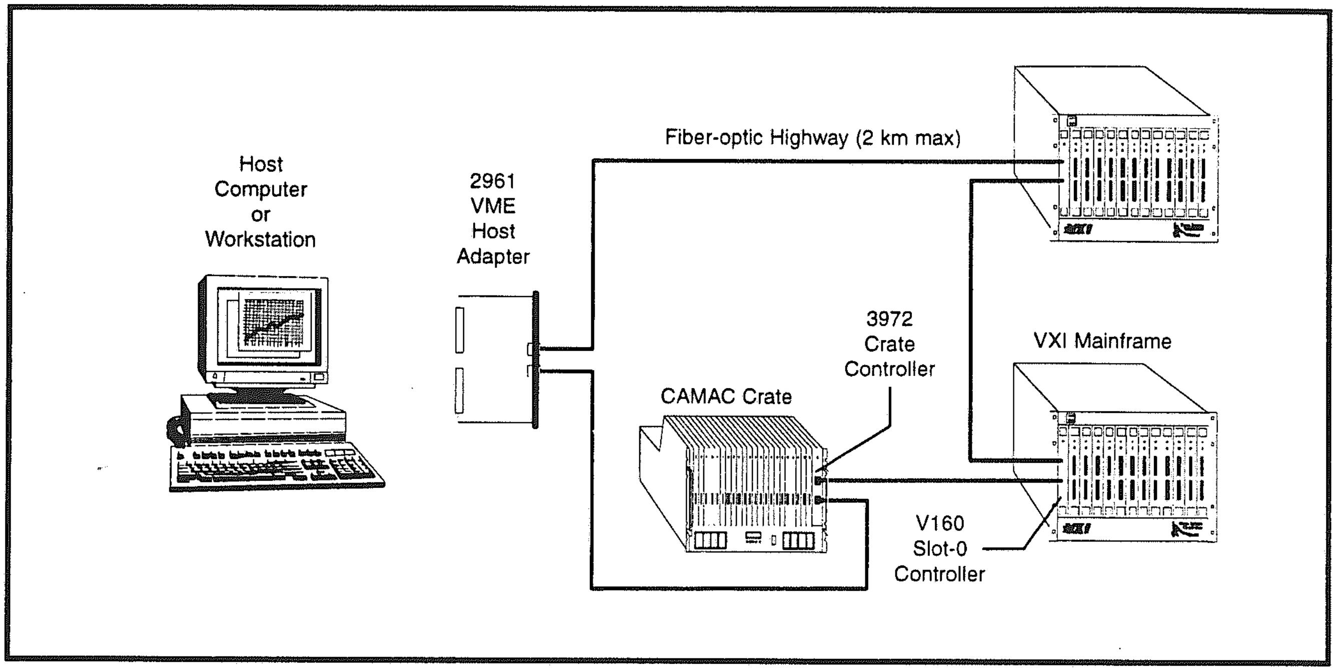
Data transfers to and from a VXI or CAMAC chassis are executed in a pipeline mode. These transfers will occur at full highway speed *if both the source and destination can maintain the data rate.* Full highway speed for the 2961 is 10 Mbytes/s over the interconnect highway. When transferring data to and from a CAMAC chassis, the 2961 provides the Q-Ignore, Q-Stop, Q-Repeat, and Q-Scan modes of operation.

The 2961 contains a 2048-word FIFO memory that stores asynchronous demand messages received from a chassis controller. These demand messages are generated from an interrupt in a VXI chassis or from a LAM (Look-At-Me) in a CAMAC chassis. The demand data identifies the chassis that sourced the demand message as well as the source of the interrupt or LAM within that chassis.

The Interconnect Host Adapter includes a command processor to execute a preloaded list of Interconnect operations. This list of operations is read from a 32K x 32 RAM memory and then executed. Command processing may be initiated by software control or by an internal, crystal-controlled, programmable counter/timer. The timer provides for the selection of triggering rates from 0.06 Hz to 500 kHz in 1-microsecond increments.

The 2961 uses host computer memory to implement multi-buffer functionality. The multi-buffer activity includes DMA data chaining and the setting of buffer segment flags.

System Configuration



Power Requirements

+5 volts:	4.3 A

Ordering Information

Model 2961-Z2A	VME Host Adapter for the Grand Interconnect

Associated Products

Model 3972-Z1X	CAMAC Crate Controller, for use with the Grand Interconnect
Model V160-xxx2	VXI Slot-0 Controller, for use with the Grand Interconnect
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

INTRODUCTION

The 2961 VME Interconnect Highway Driver (IHD) is a double-width, 6U VME card which provides the interface between a VME bus and the Interconnect Highway. The Interconnect Highway is a fiber-optic highway which allows serial access of up to 126 slave nodes. These slave nodes may be any combination of CAMAC Crate Controllers or VXI Chassis Controllers. The maximum distance between fiber-optic nodes is 2 kilometers. The highway runs at 125 Megabits-per-second using 12-bits per word, 8 of which are information bits and 4 of control. Therefore, the maximum bandwidth of data transfers on the highway is 10 Megabytes-per-second.

The 2961 executes highway operations provided by a list of instructions preloaded into a 32K x 32 list memory. This memory is loaded prior to initiating operations by programmed I/O transfers. Once list processing is initiated, data transfers to/from the 2961 may occur using either DMA or programmed transfers. DMA operations to/from the 2961 are executed using a 32-bit word format.

The list processing element on the 2961 is composed of a hardware sequencer and a TMS320C25 Digital Signal Processor (DSP). The hardware sequencer is responsible for initially interrogating list elements and executing the data transfer instructions. If the sequencer encounters a non-data transfer instruction, it enables the DSP. The DSP then executes the non-data transfer instructions and returns control to the hardware sequencer.

The list may contain such instructions as Single Operations, Block Transfer Operations, Single Inline Write Operations, Generate Host Interrupts, Trigger operations. These instructions allow for a very versatile list. The list instruction can be directed at either CAMAC or VXI chassis. The format of the instruction varies depending on the type of slave node being accessed.

A Demand FIFO is provided to retain up to 2048 demand messages. These messages are generated by both CAMAC and VXI slaves. For CAMAC slaves, these messages are typically sourced when a CAMAC Look-At-Me (LAM) occurs. VXI slaves generate these messages in response to interrupt requests in the chassis. When these messages are received by the 2961, the chassis address and a demand identification byte are stored in the FIFO. Optionally, the receipt of a demand message may assert an interrupt request to the VME bus.

INSTALLATION

The 2961 can be installed in any non-slot 0 slots (2961 requires 2 slots) in the VME/VXI Chassis. The 2961 can be configured with straps A & B (located on the main board) to reside in A32, A24, A16 address space. Address selection is made with 3 8-position dip switches also located on the main board, a switch positioned to the right selects that address. SW1 position 1-8 selects addresses 31-24 respectively, SW2 position 1-8 selects addresses 23-16, and SW3 position 1-8 selects addresses 15-8.

The 2961 can be selected using straps C & D (located on the main board) to use any of the for Bus Requests. The 2961 uses the Bus Request signal to become a VME master for DMA operations.

WHEN SELECTING A BASE ADDRESS, CARE SHOULD BE TAKEN TO AVOID ADDRESSES ASSIGNED TO OTHER DEVICES

The daughter-board (top board) must be removed to get at the straps and the address switches. This is accomplished by removing the screws in the lower left corner and the upper right corner of the daughter-board), then carefully pry the boards apart.

Straps A & B are located towards the upper right side of the main board. To configure the 2961 for:

A32 address space - Strap A in top position, Strap B in bottom position.

Address Switches SW1, SW2, SW3 used for address selection.

A24 address space - Strap A in top position, Strap B in top position Address Switches SW2, SW3 used for address selection.

A16 address space - Strap A in bottom position and Strap B in bottom position.

Address Switche SW3 used for address selection

Strap C is located towards the middle of the board on the left and strap D is towards the lower left corner.

To configure the 2961 for:

Bus Request Level 3 - Strap C to the right, Strap D to the left
Bus Request Level 2 - Strap C to the right, Strap D to the right
Bus Request Level 1 - Strap C to the left, Strap D to the left
Bus Request Level 0 - Strap C to the left, Strap D to the right

The 2961 leaves the factory strapped for A32 (strap A in top position, strap B in bottom position) address space, Bus Request Level 2 (strap C to the right and D to the right, and set for a base address of 0x20000000) (SW1 position 3 to right all other switches to the left).

Front-Panel Indicators

Error Code 4 - Error Code 1 These four LEDs indicate the priority encoded result of the last attempt to execute a Command List. The Table below shows a list of error codes.

CODE	ERROR SOURCE
F	RESERVED
E	NO SYNC
D	PARITY ERROR
C	ADDRESS NOT RECOGNIZED
B	TIMEOUT
A	VXI TIMEOUT
9	VME TMO
8	N GREATER THAN 23
7	Q-REPEAT TIMEOUT
6	NO-X
5	NO-Q
4	ILLEGAL COMMAND
3	REMOTE PARITY ERROR
2	RESERVED
1	RESERVED
0	NO ERROR

See the Control Status Register for an explanation of each error code.

No Sync — Normally off, this led is lit when the 2961 highway driver is not receiving a synchronization message from the highway. The No Sync led may be lit during large block transfer operations since sync messages are not inserted during large data transfers. The loss of Sync during a large data transfer does not cause an error condition.

List Busy — This LED is lit when the 2961 is executing a list.

Highway Busy — When lit, this LED indicates when there are messages being sent out on the highway.

Demand Pending — This LED when lit, indicates that the 2961 has at least one demand message in the demand FIFO.

Reset Switch

Access to a recessed Reset switch is provided on the front panel. When this switch is depressed the 2961 will reset to an initial state.

2961 OPERATIONAL REGISTERS

The 2961 VME Interconnect Highway Driver (IHD) uses 16 I/O addressable registers to control and monitor operations. The base address of these registers may reside in either A16, A24 or A32 address space. Please refer to the Installation section of this manual for additional information.

All data transfers to/from the 2961 must be executed using 32-bit data transfers. The 2961 does not accommodate byte or shortword accesses. The 2961 responds to VME transfers with the following Address Modifier codes shown below.

A32 Configuration:	0F, 0E, 0D, 0B, 0A, and 09 Hex
A24 Configuration:	3F, 3E, 3D, 3B, 3A, and 39 Hex
A16 Configuration:	2D and 29 Hex

The following diagram shows the operational registers along with their offsets from the base address. Note that all registers offsets shown are in hexadecimal.

Address	**Register Description**	**Mnemonic**
Base + 0	Control/Status Register	CSR
Base + 4	Burst Count / Address Modifier	BAM
Base + 8	Interrupt Control/Status	ICSR
Base + C	Timer Control Register	TCR
Base + 10	FIFO Data Register	FDR
Base + 14	Command Memory Address Register	CMA
Base + 18	Command Memory Data Register	CMD
Base + 1C	List Transfer Count Register	LTCR
Base + 20	Total Transfer Count Register	TTCR
Base + 24	Memory Address Register	MAR
Base + 28	Buffer End Address Register	BEA
Base + 2C	Buffer Interval Counter	BIC
Base + 30	Multi-Buffer Control Register	MBMCT
Base + 34	Demand FIFO Register	DFR
Base + 38	Reset Interface	RSTIFC
Base + 3C	Communication I/O	COMIO

Appendix A contains a composite register layout chart for the 2961.

Control/Status Register (Offset 0x00)

The Control/Status Register (CSR) is a write-read register located at an offset of 0x00 from the selected base address. This register is used to control and monitor various operations occurring within the 2961 and on the Interconnect Highway. Since this register contains read-only, write-only, and write/read bits, two bit patterns are shown. Those bits that are shown as zero for the write layout must be set to zero when writing to this register.

Control/Status Register(CSR)

Write Operations:

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	BEA ENA	DMA DIR	DMA ENA	SUSP	GO

Read Operations:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR CD3	ERR CD2	ERR CD1	ERR CD0	VXI TMO	QTMO	N>23	ILLG CMD	RMT PER	NO SYNC	0	PAR ERR	TMO	ADNR	NOX	NOQ

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VME TMO	BUS ERR	LIST INT	DMD OFLO	DMD PND	0	XMT FULL	RCV DAV	DONE	0	0	BEA ENA	DMA DIR	DMA ENA	SUSP	0

Bit(s)	**Mnemonic**	**Description**
<31:28>	ERRCD3-0	ERROR CODE3 through 0 are read-only bits which encode the source of a 2961 error. The following chart shows the error code hex pattern along with the source of the error. The error sources are encoded by a priority encoder resulting in the highest priority error being displayed in the case where multiple errors are encountered. Individual bits in the CSR may be read to determine other error sources that may have occurred along with the highest priority error encoded.

Error Code	Error Source
F	Reserved.
E	No Sync.
D	PARITY ERROR. The 2961 highway receiver detected a parity error on the incoming message.
C	ADDRESS_NOT_RECOGNIZED. An addressed highway command message was transmitted by the 2961 and then received by the 2961 indicating that the addressed slave did not accept the message.
B	TIMEOUT. A highway command message was transmitted by the 2961 and a reply message was not received within the preselected timeout period.

A VXI TIMEOUT. A VXI bus timeout occurred in the VXI chassis addressed during the last highway operation.

9 VME TIMEOUT. A VMEbus timeout/error occurred when the 2961 tried a direct memory access.

8 N GREATER THAN 23. N>23 indicates that a Q-Scan operation in a CAMAC chassis resulted in the station number incrementing beyond 23.

7 Q-REPEAT TIMEOUT. This error code is generated when a Q-Repeat operation in the addressed CAMAC chassis did not receive a CAMAC Q-response of one within the 200 millisecond timeout period.

6 NO-X. A highway operation to an addressed CAMAC chassis resulted in a CAMAC X-response of zero.

5 NO-Q. A highway operation to an addressed CAMAC chassis resulted in a CAMAC Q-response of zero.

4 ILLEGAL COMMAND. This error code is generated when an attempt is made at accessing a non-existent location within a CAMAC or VXI controller.

3 REMOTE PARITY ERROR. This code is generated when an addressed slave receives a parity error.

2 Reserved.

1 Reserved.

0 NO ERROR. There are no errors to report from the 2961.

Bit(s)	**Mnemonic**	**Description**
<27>	VXI TMO	VXI/VME TIMEOUT is a read-only bit that is set when an operation in an addressed VXI/VME chassis resulted in a bus timeout.
<26>	QTMO	Q-REPEAT TIMEOUT is a read-only bit that is set when a Q-Repeat operation in an addressed CAMAC chassis failed to receive a CAMAC Q-response of one within the 200 millisecond timeout period.
<25>	N>23	N GREATER THAN 23 is a read-only bit that is set when a Q-Scan operation in an addressed CAMAC chassis terminated due to the CAMAC station number incrementing beyond 23.
<24>	ILLG CMD	ILLEGAL COMMAND is a read-only bit that is set when an attempt is mode to access a non-existent address location within a CAMAC or VXI controller.
<23>	RMT PER	REMOTE PARITY ERROR is a read-only bit which is set when an addressed slave receives a parity error.
<22>	NO SYNC	NO SYNC is a read-only bit that is set when the 2961 highway driver is not receiving a synchronization message from the highway. When no list processing operations are occurring, the 2961 sends a synchronization message around the highway,

expecting to see it returned. The NO-SYNC LED may be lit during large block transfer operations since the sync message is not inserted during large data transfers. A loss of the SYNC signal during large data transfers does not set any NO SYNC error conditions.

<21> N/U This bit is not used and read as a zero.

<20> PAR ERR PARITY ERROR is a read-only bit which is set when the 2961 receives a message that contains a parity error.

<19> TMO TIMEOUT is a read-only bit that is set when the 2961 transmits a command message to a slave and the slave does not respond within the timeout period.

<18> ADNR ADDRESS_NOT_RECOGNIZED is a read-only bit that is set when the 2961 transmits a command message to a slave and receives the message back indicating that the addressed slave did not respond.

<17> NOX NO-X is a read-only bit which is set when an addressed CAMAC chassis operation resulted with a CAMAC X-response of zero.

<16> NOQ NO-Q is a read-only bit which is set when an addressed CAMAC chassis operation resulted with a CAMAC Q-response of zero.

<15> VME TMO VME TIMEOUT is a read only bit that is set when the 2961 executes a Direct Memory Access (DMA) transfer and does not receive an acknowledge from memory within 200 microseconds.

<14> BUS ERR BUS ERROR is a read only bit that is set when the 2961 executes a Direct Memory Access (DMA) transfer and receives the BERR (BUS ERROR) signal from the addressed memory.

<13> LIST INT LIST INTERRUPT is a read-only bit that is set when the list processing DSP asserts an interrupt to the host computer. This bit also appears in the Interrupt Control/Status Register.

<12> DMD OFLO DEMAND OVERFLOW is a read-only bit that is set when the 2961 has 2048 demands pending in the Demand FIFO Register and a subsequent demand message is received. This bit may be cleared by a write operation to the Clear Demand FIFO Register address.

<11> DMD PND DEMAND PENDING is a read-only bit which is set when at least one demand message is contained in the Demand FIFO Register.

<10> N/U This bit is not used and read as a zero.

<9> XMT FULL TRANSMIT FULL is a read-only bit that reflects the status of the highway transmitter write data FIFO. The full flag is used to determine if the FIFO may accept additional write data words when executing programmed I/O highway operations. As long as this bit is zero, the FIFO may be written.

<8> RCV DAV RECEIVE DATA AVAILABLE is a read-only bit which is set to a one as long as highway read data is contained in the read data FIFO. This bit is used for programmed I/O read operations for determining data availability.

<7> DONE DONE is a read-only bit which indicates when the 2961 is not executing any list operations. Once a list operation is initiated, the DONE bit is set to a zero until the operation is completed.

<6:5> N/U This bit is not used and read as a zero.

<4> BEA ENA BUFFER END ADDRESS ENABLE is a write/read bit used to control the automatic reloading of the Memory Address Register once a predetermined number of DMA operations have occurred. Setting this bit to a one enables reloading and a zero disables reloading. Refer to the Buffer End Address Register section of this manual for additional information.

<3> DMA DIR DMA DIRECTION is a write/read bit that specifies the direction of DMA transfers. A direction of zero specifies DMA transfers from host memory to the 2961 (write operations). A one specifies DMA transfers from the 2961 to host memory (read operations).

<2> DMA ENA DMA ENABLE is a write/read bit that is used to enable/disable DMA activity. Setting this bit to a one enables DMA operations to occur when the list processor requests data transfers. A zero allows for programmed I/O transfers and disables DMA operations.

<1> SUSP SUSPEND is a write/read bit used to suspend list processing operations in the 2961. After SUSPEND is set to a one, the host must wait for the DONE bit to be set before using the 2961 for other operations. After the DONE bit has been set, the SUSPEND bit may then be written back to a zero.

<0> GO GO is a write-only bit that is used to initiate list processing operations at the address specified in the List Memory Address Register.

Burst Count/Address Modifier Register (Offset 0x04)

The Burst Count/Address Modifier Register (BAM) is a write/read register located at an offset of 4 from the selected base address. This register is used to specify the VME Address Modifier that the 2961 uses during Direct Memory Address (DMA) transfers. The BAM also contains bits used to specify the maximum number of transfers that the 2961 may execute before it must rearbitrate for the VME bus. An enable bit is provided to enable/disable the burst transfer feature.

The 2961 can execute Block Mode DMA transfers when a Block Mode Address Modifier specification is loaded into the Address Modifier bit locations. During Block Mode transfers, the 2961 only supplies the address for a VME DMA transfer every 64 longwords. Devices that respond to Block Mode Address Modifiers must increment their internal addresses for each transfer since the 2961 does not supply each address. The following diagram shows the bit layout for the Burst Count/Address Modifier Register.

Burst Count/Address Modifier Register (BAM)

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
AM5	AM4	AM3	AM2	AM1	AM0	0	BCT ENA	BCT 7	BCT 6	BCT 5	BCT 4	BCT 3	BCT 2	BCT 1	BCT 0

Bit(s)	**Mnemonic**	**Description**
<31:16>	N/U	These bits are not used and read as zeros.
<15:10>	AM5 - AM0	ADDRESS MODIFIER 5 through 0 are write/read bits which specify the VME Address Modifier that the 2961 uses during DMA transfers. The following chart shows the most commonly used Address Modifiers.

Note: Short refers to A16 addressing, Standard refers to A24 addressing and Extended refers to A32 addressing.

3F	Standard supervisory block transfer
3E	Standard supervisory program access
3D	Standard supervisory data access
3B	Standard nonpriviledged block transfer
3A	Standard nonpriviledged program access
39	Standard nonpriviledged data access
2D	Short supervisory access
29	Short nonpriviledged access
0F	Extended supervisory block transfer
0E	Extended supervisory program access
0D	Extended supervisory data access
0B	Extended nonpriviledged block transfer
0A	Extended nonpriviledged program access
09	Extended nonpriviledged data access

Bit(s)	**Mnemonic**	**Description**
<8>	BCT ENA	BURST COUNT ENABLE is a write/read bit used to enable/disable the burst mode transfer feature of the 2961. Setting this bit to a one enables burst transfers and disables burst transfers when the bit is set to a zero.
<7:0>	BCT7-BCT0	BURST COUNT 7 through 0 are write/read bits which specify the maximum number of DMA transfers that the 2961 may execute before rearbitrating for the VME bus. These 8 bits provide a burst count range of 1 through 256. A count of 256 is obtained when these bits are all set to zero.

Interrupt Control/Status Register (OFFSET 0x08)

The Interrupt Configuration Register (ICSR) is a write/read register located at an offset of 8 from the selected base address. This register is used to enable/disable Interrupt Sources, specify the VME Interrupt Request Level at which the 2961 interrupts the host, and select the Interrupt Vector. The 2961 has four sources of interrupts as follows:

1.) List Processing Done
2.) List Interrupt (Generate Host Interrupt)
3.) Demand
4.) Multi-Buffer Interrupt

The ICSR contains individual enables and status bits for each of the four sources along with a global interrupt enable bit and vector selection bits. After an interrupt is acknowledged by the host, the interrupt sources are automatically cleared. The interrupt source(s) that generated the request are sent out in the least significant 4 bit location of the interrupt vector along with the 4 vector selection bits during the interrupt acknowledge bus cycle. The following shows the vector that the 2961 asserts during the interrupt acknowledge cycle. Note that at least one bit in this field is set during an acknowledge cycle.

Interrupt Vector

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	VEC 3	VEC 2	VEC 1	VEC 0	MBM INT	DMD INT	LST INT	DON INT

To allow an interrupt status bit to be enabled in the ICSR, the corresponding enable bit must be set to a one. Interrupt sources are disabled by setting the bit to a zero. In the case where polling is used instead of interrupt generation, the interrupt source bits may be cleared by writing to the ICSR with the data set equal to the source bit to be cleared. The following diagram shows the bit layout of the Interrupt Control/Status Register.

Interrupt 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
VEC 3	VEC 2	VEC 1	VEC 0	MBM INT	DMD INT	LST INT	DON INT	INT ENA	MBM IE	DMD IE	LST IE	DON IE	IRQ 2	IRQ 1	IRQ 0

Bit(s)	**Mnemonic**	**Description**
<31:16>	N/U	These bits are not used and read as zero.
<15:12>	VEC3-VEC0	VECTOR SELECTION 3-0 are write/read bits which allow the user to further define the vector that the 2961 puts out during the interrupt acknowledge cycle. Vector = Interrupt Sources (Vector bits3-0) + Vector Selection (Vector bits7-4), See Vector Layout above.
<11>	MBM INT	MULTIBUFFER MEMORY INTERRUPT SOURCE is a write/read bit that is set when the Buffer Interval Counter overflows and increments the multibuffer flags. A write of this bit with data set to one clears the source. The MBM IE bit in this register must be set to a one to enable this source.
<10>	DMD INT	DEMAND INTERRUPT SOURCE is a write/read bit that is set whenever a demand message is received by the 2961. A write of this bit with data set to one clears the source. The DMD IE bit in this register must be set to a one to enable this interrupt source.
<9>	LST INT	LIST INTERRUPT SOURCE is a write/read bit that is set when the list processor executes a Generate Host Interrupt instruction. A write of this bit with data set to one clears the source. The LST IE bit in this register must be set to a one to enable this interrupt source.
<8>	DON INT	DONE INTERRUPT SOURCE is a write/read bit that is set when the 2961 completes execution of a list. A write of this bit with data set to a one clears the source. The DON IE bit in this register must be set to a one to enable this source.
<7>	INT ENA	INTERRUPT REQUEST ENABLE is a write/read bit that is used to enable/disable the generation of a VME interrupt request. Setting this bit to a one enables VME interrupts and a zero disables the request. An interrupt request is asserted as long as the INT ENA is set to a one and any of the four interrupt sources are true.
<6>	MBM IE	MULTIBUFFER MEMORY INTERRUPT SOURCE ENABLE is awrite/read bit that enables/disables the assertion of an interrupt

source when the Buffer Interval Counter overflows during multibuffer operations. Setting this bit to a one enables this interrupt source and disabled with a zero.

<5> DMD IE DEMAND INTERRUPT SOURCE ENABLE is a write/read bit that enables/disables the generation of an interrupt source when the 2961 receives a demand message from the highway. Setting this bit to a one enables the interrupt source and a zero disables the source.

<4> LST IE LIST INTERRUPT SOURCE ENABLE is a write/read bit used to enable/disable the assertion of an interrupt source when the list processor executes a Generate Host Interrupt instruction. Setting this bit to a one enables the interrupt source and a zero disables the source.

<3> DON IE DONE INTERRUPT SOURCE ENABLE is a write/read bit used to enable/disable the generation of an interrupt source when a list processing operation completes. Setting this bit to a one enables the interrupt source and a zero disables the source.

<2:0> IRQ2-IRQ0 INTERRUPT REQUEST LEVEL 2 through 0 are write/read bits which specify the Interrupt Request Level that the 2961 should use when an interrupt request is made. The following chart shows the interrupt request levels obtained from the IRQ bit combinations.

IRQ2	IRQ1	IRQ0	Interrupt Request Level
0	0	0	Not Used
0	0	1	IRQ Level 1
0	1	0	IRQ Level 2
0	1	1	IRQ Level 3
1	0	0	IRQ Level 4
1	0	1	IRQ Level 5
1	1	0	IRQ Level 6
1	1	1	IRQ Level 7

Timer Control Register (Offset 0x0C)

The Timer Control Register (TCR) is a write/read register located at an offset of 0C hex from the selected base address. This register is used to specify the frequency at which list execution is initiated during timer initiated list processing operations.

The timer frequency can range from 250 Kilohertz to .059 hertz. This range yields 'tic' rates (periods) from 40 microseconds to 16.777 seconds in 1 microsecond increments. This timer rate specification does not refer to the interval at which each element in the list is executed but the rate at which the entire list is initiated. This automatic mode is enabled with bit 24 in this register.

The 2961 provides for an external clock input that controls the rate at which the list is started. This input is routed through the external clock LEMO connector mounted on the front panel of the 2961. The clock period may range from 40 microseconds to DC.

If the timer rate selected for list execution is faster than the time it takes to execute the entire list, the clock transitions that occur while the list is executing are ignored. The following diagram shows the bit layout of the Timer Control Register.

Timer Control Register (TCR)

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

Bit(s)	**Mnemonic**	**Description**
<31:26>	N/U	These bits are not used and read as zeros.
<25>	CLK SRC	CLOCK SOURCE is a write/read bit that select the source of the timing signal that initiates list execution when the TIM ENA bit is set to a one. Setting the CLK SRC bit to a one selects the external clock input and a zero selects the internal timer.
<24>	TIM ENA	TIMER ENABLE is a write/read bit that is used to enable/disable timer initiated list processing operations. Setting this bit to a one enables timer initiated operations and a zero disables timer operations.
<23:0>	TCR23-TCR0	TIMER CONTROL RATE23 through 0 are write/read bits that are used to select the rate at which list processing operations are initiated when the timer operations are enabled. The least significant bit of this field corresponds to a 1 microsecond interval. The minimum period specification is 40 microseconds and the maximum is 16.777 seconds.

FIFO Data Register (Offset 0x10)

The FIFO Data Register (FDR) is a write/read register located at an offset of 10 Hex from the selected base address. This register is used for transferring data to/from the highway when

DMA (Direct Memory Access) operations are not used. When DMA operations are used to write/read data from the highway, the DMA data uses the FIFO data paths without any software intervention.

If it is necessary to transfer data to/from the highway without using DMA, programmed I/O transfers may be used to move the data to/from the data FIFO's. When programmed I/O is used to transfer the data, two flag bits located in the Control/Status Register (CSR) must be used. In the case of executing highway write operations, the user code must ensure that the write data FIFO does not overflow. Therefore, the Transmit Full (XMT FUL) bit in the CSR must be checked before a write operation to the FIFO Data Register is executed. For read operations, the user code must wait for the Receive Data Available (RCV DAV) in the CSR to be asserted before any data is retrieved from the FIFO Data Register.

The DFR contains only 16-bits of data for either write or read data. Therefore, to write/read one 32-bit word requires two accesses to this register. The first word of two for 32-bit transfers must be the lower 16-bits first followed by the high 16-bits. Please refer to the Executing Programmed Transfers section of this manual for additional information. The following diagram shows the bit pattern for the FIFO Data Register.

FIFO Data Register (FDR)

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
FDR 15	FDR 14	FDR 13	FDR 12	FDR 11	FDR 10	FDR 9	FDR 8	FDR 7	FDR 6	FDR 5	FDR 4	FDR 3	FDR 2	FDR 1	FDR 0

Bit(s)	**Mnemonic**	**Description**
<31:16>	N/U	These bits are not used and read as zeros.
<15:0>	FDR15-FDR0	FIFO DATA15 through 0 are write/read bits which are used to transfer data to/from the highway during programmed I/O (non-DMA) transfers. Note that a read from this register does not return the last data written through programmed I/O.

Command Memory Address Register (Offset 0x14)

The Command Memory Address Register (CMA) is a write/read register located at an offset of 14 Hex from the selected base address. This register is used for several purposes. The CMA is used to specify the start address for list processing, specify the memory address at which subsequent write/read operations to the Command Memory Data Register occur, and to read the last address that the list processor accessed during a list processing operation.

The Command Memory Data Register (CMD) is a 32K X 32 memory that is used to hold the list processing instructions to be executed. The CMD is loaded by executing programmed I/O write operations to at offset 18 Hex. Prior to executing the initial write to the memory, the CMA must first be loaded with the first address location to access. The valid address range is 0 to 7FFF

Hex. After the CMA is loaded, the first CMD data word may then be written. After a write operation to the CMD, the CMA is automatically incremented to the next address location. This allows the CMD to be loaded without having to write the CMA for every CMD write operation.

If it is necessary to read the CMD, the CMA must first be loaded with the initial address to be accessed. After a word of data is read from the CMD, the CMA is automatically incremented to the next address location.

A bit in this register is also provided which initiates a single list processing operation. Setting this bit causes the same operation as setting the GO bit in the Control/Status Register. The format of the Command Memory Address Register is shown in the following diagram.

Command Memory Address Register (CMA)

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
LST GO	CMA 14	CMA 13	CMA 12	CMA 11	CMA 10	CMA 9	CMA 8	CMA 7	CMA 6	CMA 5	CMA 4	CMA 3	CMA 2	CMA 1	CMA 0

Bit(s)	**Mnemonic**	**Description**
<31:16>	N/U	These bits are not used and read as zeros.
<15>	LST GO	LIST GO is a write-only bit which is set to a one to initiate a list processing operation. Setting this bit has the same effect as setting the GO bit in the CSR. This bit is not latched.
<14:0>	CMA14- CMA0	COMMAND MEMORY ADDRESS14 through 0 are used for specifying the initial address for CMD write/read operations, the initial address for list execution, and for determining where the list processor halted after list processing.

Command Memory Data Register (Offset 0x18)

The Command Memory Data Register (CMD) is a write/read register located at an offset of 18 Hex from the selected base address. This register is used to load the instructions for the list processor. The format of these instructions/commands may be found in the Command Memory Instructions section of this manual.

The Command Memory is a 32K X 32 memory which is accessed by programmed I/O. Before initial words may be written to the CMD, the Command Memory Address Register (CMA) must be loaded. After the CMA is loaded with the first address of the CMD to access, the CMD may then be written. After a write operation to the CMD, the CMA is automatically incremented to the next sequential address location. This eliminates the need to reload the CMA for every access to the CMD.

If necessary, the CMD may be read to verify its contents. As with write operations to the CMD, the CMA must be loaded prior to the initial access to the CMD. After a read operation is executed to the CMD, the CMA is automatically incremented. The format of the Command Memory Data Register is shown in the following diagram.

Command Memory Data Register (CMD)

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

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

Bit(s)	**Mnemonic**	**Description**
<31:0>	CMD31-CMD0	COMMAND MEMORY DATA31 through 0 are write/read bits used to load/read the 32K X 32 Command Memory.

List Transfer Count Register (Offset 0x1C)

The List Transfer Count Register (LTCR) is a read-only register located at an offset of 1C Hex from the selected base address. This register returns the two's complement of the number of transfers remaining during a block transfer mode of operation. When the list processor encounters a block transfer instruction, it loads the 32-bit List Transfer Count Register with the count specified in the instruction. This value is the two's complement of the number of transfers that are to occur for the block transfer operation. When this counter is incremented to zero, the block transfer operation terminates and the next instruction in the list is interpreted. If an error occurs during the block operation, the LTCR may then be read to determine the number of transfers remaining.

The List Transfer Count reflects the number of transfers without regard to the data word size.

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

List Transfer Count Register (LTCR)

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

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

Bit(s)	**Mnemonic**	**Description**
<31:0>	LTC31-LTC0	LIST TRANSFER COUNT31 through 0 are read-only bits which reflect the number of transfers remaining after a block transfer operation concludes. This data is in two's complement format.

Total Transfer Count Register (Offset 0x20)

The Total Transfer Count Register (TTCR) is a write/read register located at an offset of 20 Hex from the selected base address. This register is used to specify the maximum number of transfers that are to occur for one iteration of the entire list. Before a list operation is executed where data transfers to/from the highway occur using Direct Memory Access (DMA), the Total Transfer Count Register must be loaded with the two's complement of the maximum number of transfers. In case that the list terminates prematurely, the TTCR may be read to determine the number of transfers remaining to be executed.

Along with write access to the TTCR from the host by programmed I/O, the TTCR may also be loaded by using a list instruction. When the list processor encounters this instruction, the contents of the count specification are loaded into the TTCR.

The TTCR is incremented once for every DMA write or read access to VME. When the counter increments to zero, the transfer is considered complete. The following diagram shows the bit layout for the Total Transfer Count Register.

Total Transfer Count Register (TTCR)

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

Bit(s)	**Mnemonic**	**Description**
<31:0>	TTC31-TTC0	TOTAL TRANSFER COUNT31 through 0 are write/read bits which specify the maximum number of DMA transfers to execute for an iteration of the list. A read of this register returns the two's complement of the number of transfer remaining.

Memory Address Register (Offset 0x24)

The Memory Address Register (MAR) is a write/read register located at an offset of 24 Hex from the selected base address. This register is used to specify the initial Direct Memory Access (DMA) address. Before any DMA operations are executed by the 2961, the MAR must be loaded with the first address to be accessed during DMA write or read operations. After the 2961 executes the first DMA operation, the MAR is automatically incremented to the next sequential longword address.

When the Buffer End Address (BEA ENA) bit in the CSR is enabled, the MAR may automatically reload itself after the Buffer End Address (counter) expires. The value that is reloaded is contained in the MAR. This feature allows the 2961 to DMA data to a circular buffer.

Along with write access to the MAR from the host by programmed I/O, the MAR may also be loaded by using a list instruction. When the list processor encounters this instruction, the contents of the count specification are loaded into the MAR.

DMA operations to/from the 2961 are always 32-bits in length. Therefore, the MAR specification must be on a longword boundary. The following diagram shows the bit pattern for the Memory Address Register.

Memory Address Register (MAR)

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

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

Bit(s)	Mnemonic	Description
<31:0>	MAR31-MAR2	MEMORY ADDRESS31 through 2 are write/read bits MAR2 used to specify the initial DMA Memory Address.
<1:0>	N/U	These bits are not used and read as zeros.

Buffer End Address Register (Offset 0x28)

The Buffer End Address Register (BEA) is a write/read register located at an offset of 28 Hex from the selected base address. This register is used to specify the end address count at which the Memory Address Register (MAR) is reloaded, creating a circular buffer. As DMA operations are executed by the 2961, the MAR is incremented to the next sequential longword address and the BEA is decremented. When the BEA count is exhausted, the 2961 reloads the Memory Address Register with the initial value it contained when the MAR was loaded with programmed I/O and the Buffer End Address counter is also reloaded. The following diagram shows the bit pattern for the Buffer End Address Register.

Buffer End Address Register (BEA)

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

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

Bit(s)	Mnemonic	Description
<31:0>	BEA31- BEA0	BUFFER END ADDRESS 31 through 0 are write/read bits used to specify the number of DMA transfers that the 2961 executes before reloading the Memory Address Register. The circular buffer action is enabled using the BEA ENA bit in the CSR.

Buffer Interval Counter (Offset 0x2C)

The Buffer Interval Counter (BIC) is a write/read register located at an offset of 2C Hex from the selected base address. This counter is used to define the number of DMA transfers that the 2961 executes before a multibuffer flag bit and an interrupt is generated. The BIC allows the 2961 to use host computer memory to simulate a multibuffer memory. When using this mode, the 2961 is continuously acquiring data and transferring the read data to memory via DMA. When a predetermined number of DMA transfers have occurred, a multibuffer flag is set and optionally an interrupt generated. When the host sees the flag set, it may then read the buffer of data received. As additional words are transferred to memory, subsequent multibuffer flags are set.

When the Buffer End Address counter expires, the Memory Address Register is reloaded, creating a circular buffer. This data acquisition sequence continues until the 2961 is disabled.

Refer to the Multibuffered Data Acquisition section of this manual for additional information. The following diagram shows the bit pattern for the Buffer Interval Counter.

Buffer Interval Counter Register (BIC)

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

Bit(s)	Mnemonic	Description
<31:0>	BIC31-BIC0	BUFFER INTERVAL COUNTER 31 through 0 are write/read bits used to specify the number of DMA transfers that the 2961 executes before a multibuffer flag is set during multibuffered data acquisition.

Multibuffer Memory Control Register (Offset 0x30)

The Multibuffer Memory Control Register (MBMCT) is a write/read register located at an offset of 30 Hex from the selected base address. This register is used to enable the multibuffer flags and to monitor the status of the flags.

Once multibuffer activity has been enabled, the multibuffer flags in this register are set as the Buffer Interval Counter (BIC) expires. The BIC is decremented as the 2961 executes each DMA operation. When the BIC is exhausted, the next sequential flag bit is set. After the fourth flag is set, the next expiration of the BIC causes the first flag to be set. If the host computer has not cleared a flag, via programmed I/O, before the 2961 loops around to set it again, the flag overflow bit is set. Refer to the Multibuffered Data Acquisition section of this manual for additional information. The following diagram shows the bit pattern for the Multibuffer Control Register.

Multibuffer Memory Control Register (MBMCT)

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	MBM ENA	FLG OFL	FLG 3	FLG 2	FLG 1	FLG 0

Bit(s)	**Mnemonic**	**Description**
<31:6>	N/U	These bits are not used and read as zeros.
<5>	MBM ENA	MULTIBUFFER ENABLE is a write/read bit used to enable/disable the generation of the flag bits that correspond to the expiration of the Buffer Interval Counter (BIC). Setting this bit to a one enables the flags and disabled by a zero.
<4>	FLG OFL	FLAG OVERFLOW is a read/clear bit which is used to signify an overflow condition. This bit is set whenever the 2961 needs to set a multibuffer flag and the flag is already set, indicating that a buffer of data is being overwritten before the host could read the buffer and clear the flag. A write operation with this bit set to a one causes the overflow condition to be cleared.
<3:0>	FLG3-FLG0	MULTIBUFFER FLAG3 through 0 are read/clear bits that are sequentially set as the Buffer Interval Counter expires during multibuffer data acquisition. A write operation to these bits with any bit set to a one causes the flag to be cleared.

Demand FIFO Register (Offset 0x34)

The Demand FIFO Register (DFR) is a write (to clear)/read register located at an offset of 34 Hex from the selected base address. This register is loaded with demand message data received from the highway. When a slave chassis required service, it may generate an asynchronous demand message on the highway. When the 2961 receives these demands, they are placed in a 2K X 16 FIFO. Optionally, the 2961 may assert a VME interrupt when the 2961 receives a demand message. The FIFO data contains the node address of the chassis that generated the demand message along with an 8-bit identifier. Please refer to the individual slave module manuals for details on the sources of the identifiers.

As long as there is at least one demand message contained in the Demand FIFO Register the DEMAND PENDING (DMD PND) bit in the CSR is set to a one. If the DFR is full (2048 demands) and an additional demand message is received from the highway, the DEMAND OVERFLOW (DMD OFL) bit in the CSR is set to a one. The Demand FIFO Register and the DEMAND OVERFLOW bit are cleared by power-up, a write to the Reset Interface address, and by a write to the Reset Demand FIFO address.

A write operation to this register with any data pattern clears the Demand FIFO and the DEMAND OVERFLOW bit. The following diagram shows the bit pattern for the Demand FIFO Register Read.

Demand FIFO Register (DFR)

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	NDA 64	NDA 32	NDA 16	NDA 8	NDA 4	NDA 2	NDA 1	DID 7	DID 6	DID 5	DID 4	DID 3	DID 2	DID 1	DID 0

Bit(s)	**Mnemonic**	**Description**
<31:15>	N/U	These bits are not used and read as zeros.
<14:8>	NDA64-NDA1	NODE ADDRESS64 through 1 are read-only bits which indicates the chassis' node address that generated the demand message.
<7:0>	DID7-DID0	DEMAND IDENTIFIER7 through 0 are read-only bits which indicate the chassis' internal demand source.

Reset Interface (Offset 0x38)

The Reset Interface (RSTIFC) is a write-only address location located at an offset of 38 Hex from the selected base address. A write operation to this address with any data pattern causes the 2961 to be reset to an initial state.

Communication I/O (Offset 0x3c)

The Communication I/O Register (COMIO) is a write/read register located at an offset of 3C Hex from the selected base address. This register provides a general purpose communication interface between the DSP and the host CPU. The details of this communication scheme is beyond the scope of this manual. The following diagram shows the bit pattern for the Communication I/O register.

Communication I/O 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
COM 15	COM 14	COM 13	COM 12	COM 11	COM 10	COM 9	COM 8	COM 7	COM 6	COM 5	COM 4	COM 3	COM 2	COM 1	COM 0

Bit(s)	**Mnemonic**	**Description**
<31:16>	N/U	These bits are not used and read as zeros.
<15:0>	COM15-COM0	These bits are for communication between the DSP and the host CPU.

PROGRAMMED TRANSFERS

Programmed transfers are used to setup and initiate highway operations. Once these operations are started, data transfer to/from the Data FIFO Registers may occur under programmed transfer or by Direct Memory Access (DMA). DMA transfers have a significant speed advantage over programmed transfers. The DMA transfers to the 2961 occur without any software overhead once the transfer is setup.

When using programmed transfers, the software must examine the FIFO flags in order to determine whether to transfer data to/from the Data FIFO Resister (FIFO). These flags are located in the Control/Status Register of the Highway Operational Registers. Before a write operation to the 2961 FIFOs may occur, the software must read the Control/Status Register (CSR) to determine if there is room in the FIFO for another piece of write data. If the TRANSMIT FIFO FULL (XMT FUL) bit in the Control/Status Register is set the software must suspend the write operation until the bit is negated. When read commands are executed, the software must check the Control/Status Register for the RECEIVE DATA AVAILABLE (RCV DAV) bit before the data may be retrieved. If this bit is not set, the software must loop on reading this registers until the bit is asserted. These procedures must be followed when executing programmed transfer operations to the FIFO while list processing is in progress.

The following is a basic example of how to execute a programmed transfer list operation. For this example, the direction of data transfer is from the 2961 to a highway slave device.

1.) Set the Command Memory Address Register (CMA) to zero.
2.) Load the Command Memory Data Register (CMD) with the instructions required to execute the write operation. Make sure that the last element in the list is a HALT instruction.
3.) Set the Command Memory Address Register (CMA) to zero.
4.) Set the GO bit in the Control/Status Register (CSR).
5.) Read the Control/Status Register (CSR) and loop on this step until the TRANSMIT FIFO FULL (XMT FUL) bit is false.

6.) Write the FIFO Data Register (FDR) with the data to be transmitted. Return to step 5 until all the transmit data words have been loaded into the 2961.
* 7.) Wait for the DONE bit in the Control/Status Register (CSR) to be asserted, indicating that the operation is complete.
8.) Read the Control/Status Register (CSR) to check for any errors.

The following example shows the steps required to execute a programmed transfer operation that transfers data from a highway slave device to the 2961.

1.) Set the Command Memory Address Register (CMA) to zero.
2.) Load the Command Memory Data Register (CMD) with the instructions required to execute the write operation. Make sure that the last element in the list is a HALT instruction.
3.) Set the Command Memory Address Register (CMA) to zero.
4.) Set the GO bit in the Control/Status Register (CSR).
5.) Loop on reading the Control/Status Register (CSR) until RECEIVE DATA AVAILABLE (RCV DAV) bit is set.
6.) Read the data from the FIFO Data Register (FDR) and return to step 5 until all data words have been read.
* 7.) Wait for the DONE bit in the Control/Status Register (CSR) to be asserted indicating that the operation is complete.
8.) Read the Control/Status Register (CSR) to check for any errors.

* When using the 2961 with the Done Interrupt enabled, the steps that include waiting for DONE may be removed since the assertion of the DONE bit generates an interrupt.

DMA TRANSFERS

Direct Memory Access (DMA) transfers provide the most efficient mechanism of transferring data to/from the 2961 during list processing operations. Minimal programming is required to setup the DMA mechanism.

Before a DMA operation begins, the 2961 must have the Memory Address Register (MAR) loaded with the initial host memory address to access during the operation. After the 2961 accesses the specified location, the memory address register is incremented to the next longword address. Note that the address specification must be longword aligned. As the DMA operations progresses, the Memory Address Register is incremented until the DMA operation is complete.

The maximum number of transfers that the 2961 is to executed must be loaded in the Total Transfer Count Register (TTCR) prior to enabling DMA. This register is loaded with the two's complement of the number of transfers to occur during the DMA operation. This value must take into account all data transfer instructions in the list, excluding the Single Inline Write operation. As each data word is transferred to/from host memory, the transfer count is incremented. When the count is incremented to zero, the DMA operation is complete.

The BUFFER END ADDRESS ENABLE (BEA ENA) bit in the Control/Status Register is used to establish a circular buffer in host memory for repetative operations. When the buffer end address is enabled, the Memory Address Register and Buffer End Address are reloaded with the initial data after the Buffer End Address expires. The Memory Address and Total Transfer

Count Registers may also be loaded by the list processor by inserting the appropriate instructions in the list. The DMA direction may also be changed by list instructions.

COMMAND MEMORY INSTRUCTION FORMATS

The 2961 contains a 32K x 32 word memory used to hold the list processing instructions to be executed by the hardware list processor or the DSP. When list processing is initiated, the hardware list processor examines the header instruction of the first list instruction. If the header indicates that the instruction is a data transfer instruction to a CAMAC or VXI/VME chassis, the hardware processor forms the necessary command to transmit on the highway. After the highway transaction is complete, the Command Memory Address is incremented and the next instruction interpreted.

If the hardware list processor encounters a "special" instruction, one that does not require a highway operation, the DSP is enabled. The DSP then executes the "special" instruction and continues processing until a HALT instruction is found or another data transfer operation is requested.

The lower 16-bits of the first longword of a command instruction specifies the type of the instruction. This 16-bit word is referred to as the instruction header. The following diagram shows the format of the instruction header.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CM 1	CM 0	NDA 64	NDA 32	NDA 16	NDA 8	NDA 4	NDA 2	NDA 1	TM 2	TM 1	AM 2	AM 1	WS 2	WS 1	AD

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

CM1	CM0	Instruction Type
0	0	CAMAC Instruction
0	1	VXI/VME Instruction
1	0	Special Instruction
1	1	Reserved

If the CM1 bit is a zero, indicating that highway data transfers are to occur, bits 13 through 7 specify the node address at which the addressed command is directed. For special instructions, these bits may take on other definitions. The bits 13 through 7 for data transfer instructions allow a node address value from one to 126. Even though node address 0 is valid, it is reserved for future expansion.

The definitions of the remaining bits in the instruction header vary depending on the type of command. The next two sections fully define bits 6 through 0 of the instruction header.

Subsequent words in a command instruction are dependant on the command type. For example, a CAMAC header is followed by the Station Number, Function Code, and Subaddress whereas VXI/VME headers are followed by the 32-bit VME address.

The next two sections of this manual describe the various CAMAC and VXI/VME instructions.

CAMAC Instructions

For all CAMAC instructions, the instruction header has bits 15 and 14 set to zeros. Bits 13 through 7 are used to specify the node address of the CAMAC chassis to access during the operation. Other bits in this word define the Transfer Mode, Access Mode, and Data Word Size. The following diagram shows the instruction header for CAMAC instructions.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	NDA 64	NDA 32	NDA 16	NDA 8	NDA 4	NDA 2	NDA 1	TM 2	TM 1	AM 2	AM 1	WS 2	WS 1	AD

Bits 15 and 14 of the CAMAC instruction header must be set to zeros.

Bits 13 through 7 are used to specify the node address of the CAMAC chassis to access.

Bits 6 and 5 are the Transfer Mode bits and specify the CAMAC transfer protocol as follows:

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

The Single Operation Transfer Mode simply transfers one data word to/from the CAMAC chassis for the specified CAMAC command. These operations may include write, read and control operations.

Block Transfer Operations move blocks of data to/from a CAMAC chassis for each block instruction. The number of data words to transfer during Block Transfer operations are found in additional words accompanying this instruction. These operations may include either CAMAC write or read operations.

Single Inline Write Operations are single transfer CAMAC write operations that have the CAMAC write data embedded in the list. These commands are useful for initializing modules and also allow CAMAC write operations to occur in a read command list.

The Access Mode bits, AM2 and AM1, specify the access method to be used during the requested transfer. The Access Modes provided are as follows.

AM2	AM1	Access Mode
0	0	Q-Stop Mode
0	1	Q-Ignore Mode
1	0	Q-Repeat Mode
1	1	Q-Scan Mode

The Access Modes for CAMAC provide a mechanism for controlling the data transfer base on the CAMAC Q-response received after every Dataway cycle. These modes include stopping when a Q=0 response is found, repeating a CAMAC command until a Q=1 response is found, or scanning through a CAMAC chassis. Please refer to the CAMAC Access Modes section of this manual for additional information.

The Word Size bits, WS2 and WS1, are used to specify the size of the CAMAC data word accessed in the CAMAC chassis. The following chart shows the available CAMAC data word size selections.

WS2	WS1	Data Word Size
0	0	32-Bits
0	1	24-Bits
1	0	16-Bits
1	1	8-Bits

Even though 32-bit transfers are not supported by CAMAC, this data word size was included for compatibility with VXI/VME data word sizes. If 32-bit data word sizes are selected, they are treated as 24-bit transfers.

The last bit in the instruction header, ABORT DISABLE, is used to enable or disable the termination of an operation when an X-response of zero occurs. Setting this bit to a one enables the termination due to a X=0 response from a CAMAC module and a zero enables the X=0 termination.

CAMAC NAF Word

The second 16-bit word for a CAMAC instruction contains the CAMAC Station Number (N), Subaddress (A), and the Function Code (F). This second 16-bit word is the high 16-bits of the first 32-bit Command Memory Data word. The following diagram shows the format of the second instruction word containing the CAMAC NAF specification.

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

Bits 31 and 30 are not used and must be set to zero.

The Station Number 16 through 1 bits are used to specify the CAMAC Station Number to be used for the addressed operation. This allows for a CAMAC Station Number specification in the range of 0 through 31. Valid CAMAC Station Numbers that specify actual slots within the chassis are 1 through 23. Station Number 30 is a pseudo-address and is used to access the internal registers of the 3972. Station Numbers 24 through 29 and 0 are reserved. Please refer to the 3972 CAMAC Crate Controller for additional information on internal controller registers.

The Subaddress 8 through 1 bits are used to specify the CAMAC Subaddress to be accessed during the addressed CAMAC operation. Valid Subaddresses range from 0 to 15.

The Function Code 16 through 1 bits are used to specify the CAMAC Function Code to be used 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
0	1	CAMAC Control *
1	0	CAMAC Write
1	1	CAMAC Control *

* CAMAC Control operations are dataless operations. A Dataway cycle is executed for these commands but no data is transferred.

CAMAC Single Transfer Instructions

The Header Word and CAMAC NAF specification are common for all CAMAC instructions. This information occupies an entire 32-bit Command Memory Data word. For CAMAC Single Operations, this is the only data required. For each Single Transfer instruction encountered in the list, only one data word is transferred. The direction of the transfer is based on the CAMAC Function Code specified for the command. This instruction format may also be used to execute CAMAC control operations. To transfer multiple data words to/from a CAMAC chassis with one instruction, the CAMAC Block Transfer instruction must be used. The following diagram shows the composite format for the CAMAC Single Operation instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	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	NDA 64	NDA 32	NDA 16	NDA 8	NDA 4	NDA 2	NDA 1	0	0	AM 2	AM 1	WS 2	WS 1	AD

CAMAC Block Transfer Instructions

CAMAC Block Transfer instructions allow multiple data words to be transferred to/from a CAMAC chassis with one instruction. This provides an efficient method for transferring a block of data to/from a particular CAMAC NAF.

Associated with each Block Transfer instruction is a Transfer Count. The Transfer Count, which is 32-bits is length, is the two's complement of the maximum number of CAMAC words to transfer during 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 finds a Block Transfer instruction in the Command Memory, it loads the List Transfer Count Register on the 2961 with the Transfer Count data found in the list. The List Transfer Count Register is then incremented as each data word is transferred to/from the CAMAC chassis. The Block Transfer operation continues until the List Transfer Count is exhausted or an error occurs. If the transfer terminates prematurely, the List Transfer Count Register may then be read to determine the number of data words remaining to transfer.

The format of the CAMAC Block Transfer instruction is shown in the following diagram.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	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	NDA 64	NDA 32	NDA 16	NDA 8	NDA 4	NDA 2	NDA 1	0	1	AM 2	AM 1	WS 2	WS 1	AD

2nd Longword:

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

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

CAMAC Single Inline Write Instruction

The 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 to execute a CAMAC write operation in a read list.

The format of the CAMAC Single Inline Write instruction is shown in the following diagram.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	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	NDA 64	NDA 32	NDA 16	NDA 8	NDA 4	NDA 2	NDA 1	1	0	AM 2	AM 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	W 23	W 22	W 21	W 20	W 19	W 18	W 17	W 16

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

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

VXI/VME Instructions

For all VXI/VME instructions, the instruction header has bits 15 set to a zero and bit14 set to a one. Bits 13 through 7 are used to specify the node address of the VXI/VME chassis to access during the operation. Other bits in this word define the Transfer Mode, Access Mode, and Data Word Size. The following diagram shows the instruction header for VXI/VME instructions.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	1	NDA 64	NDA 32	NDA 16	NDA 8	NDA 4	NDA 2	NDA 1	TM 2	TM 1	AM 2	AM 1	WS 2	WS 1	AD

Bits 15 must be set to a zero and bit 14 to a one.

Bits 13 through 7 are used to specify the node address of the VXI/VME chassis to access.

Bits 6 and 5 are the Transfer Mode bits and specify the VXI/VME transfer protocol as follows:

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

The Single Operation Transfer Mode simply transfers one data word to/from the VXI/VME chassis.for the specified VXI/VME command. These operations may include write or read operations.

Block Transfer Operations move blocks of data to/from a VXI/VME chassis for each block instruction. The number of data words to transfer during Block Transfer operations are found in additional words accompanying this instruction. These operations may include either VXI/VME write or read operations.

Single Inline Write Operations are single transfer VXI/VME write operations that have the write data embedded in the list. These commands are useful for initializing modules and also allow VXI/VME write operations to occur in a read command list.

The Access Mode bits, AM2 and AM1, specify the access method to be used during the requested transfer. This mode concerns the addressing of the VXI/VME chassis. After the initial VXI/VME transfer is executed during block transfers, the address may be either incremented or left the same for subsequent transfers of the block. The Access Modes provided are as follows.

AM2	AM1	Access Mode
0	0	Increment Address
0	1	Reserved
1	0	Address Unchanged
1	1	Reserved

The Word Size bits, WS2 and WS1, are used to specify the size of the VXI/VME data word accessed in the addressed chassis. The following chart shows the available VXI/VME data word size selections.

WS2	WS1	Data Word Size
0	0	32-Bits
0	1	Reserved
1	0	16-Bits
1	1	8-Bits

The last bit in the instruction header, ABORT DISABLE, is used to enable or disable the termination of an operation when an a VXI Timeout is encountered. Setting this bit to a one allows an operation to run to completion without regard to the bus timeouts. Setting this bit to a zero causes an operation to terminate when a bus error occurs.

VXI/VME Address

The second 16-bit word for VXI/VME instructions contains the VME Address Modifier, a bit indicating the direction of the transfer (write or read), and a bit that is set to execute operations within the chassis controller. This second 16-bit word is the high 16-bits of the first 32-bit Command Memory Data word. The following diagram shows the second instruction word for VXI/VME instructions.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT	DIR	0	0	0	0	0	0	0	0	AMD 5	AMD 4	AMD 3	AMD 2	AMD 1	AMD 0

The INTERNAL bit, bit 31, is used to indicate whether the addressed command is to be executed internal to the chassis controller or external. Setting this bit to a one specifies that the operation is executed internal to the chassis controller. For operations to occur on the VXI/VME backplane, this bit must be set to a zero.

The DIRECTION bit, bit 30, is used to define the direction of the transfer. When DIRECTION is set to zero, the direction of the transfer is from the 2961 to the VXI/VME chassis controller (write operations). This bit is set to a one for transferring data from the addressed VXI/VME chassis to the 2961 (read operations).

Bits 29 through 22 are not used and must be set to zeros.

The ADDRESS MODIFIER bits, AMD5 through AMD0, are used to specify the VME address modifier that the addressed chassis controller is to use for the VXI/VME bus cycle. The following chart is a subset of the VME address modifiers.

Address Modifier Data (Hex)	Function
3F	Standard (A24) Supervisory Block Transfer
3E	Standard (A24) Supervisory Program Access
3D	Standard (A24) Supervisory Data Access
3B	Standard (A24) Nonprivleged Block Transfer
3A	Standard (A24) Nonprivleged Program Access
39	Standard (A24) Nonprivleged Data Access
2D	Short (A16) Supervisory Access
29	Short (A16) Nonprivleged Access
0F	Extended (A32) Supervisory Block Transfer
0E	Extended (A32) Supervisory Program Access
0D	Extended (A32) Supervisory Data Access
0B	Extended (A32) Nonprivleged Block Transfer
0A	Extended (A32) Nonprivleged Program Access
09	Extended (A32) Nonprivleged Data Access

The next longword (32-bits) of any VXI/VME instruction is the physical address. The following diagram shows the bit pattern for the 2nd longword of a VXI/VME instruction.

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

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

VXI/VME Single Transfer Instructions

The Header Word, Address Modifier, and Address specification are common for all VXI/VME instructions. This information occupies two 32-bit Command Memory Data words. For VXI/VME Single Operations, this is the only data required. For each Single Transfer instruction encountered in the list, only one data word is transferred. The direction of the transfer is controlled by the DIRECTION bit. To transfer multiple data words to/from a VXI/VME chassis with one instruction, the VXI/VME Block Transfer instruction must be used.

The following diagram shows the composite format for the VXI/VME Single Operation instruction.

1st Longword

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT	DIR	0	0	0	0	0	0	0	0	AMD 5	AMD 4	AMD 3	AMD 2	AMD 1	AMD 0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	1	NDA 64	NDA 32	NDA 16	NDA 8	NDA 4	NDA 2	NDA 1	0	0	AM 2	AM 1	WS 2	WS 1	AD

2nd Longword:

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

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

VXI/VME Block Transfer Instructions

VXI/VME Block Transfer instructions allow multiple data words to be transferred to/from a VXI/VME chassis with one instruction. This provides an efficient method for transferring a block of data to/from a particular module(s).

Associated with each Block Transfer instruction is a Transfer Count. The Transfer Count, which is 32-bits is length, is the two's complement of the maximum number of VXI/VME words to transfer during the block operation. This transfer count specifies the number of data words to transfer, regardless of the Data Word Size.

When the list processor finds a Block Transfer instruction in the Command Memory, it loads the List Transfer Count Register on the 2961 with the Transfer Count data found in the list. The List Transfer Count Register is then incremented as each data word is transferred to/from the VXI/VME chassis. The Block Transfer operation continues until the List Transfer Count is exhausted or an error occurs. If the transfer terminates prematurely, the List Transfer Count Register may then be read to determine the number of data words remaining to transfer. The format of the VXI/VME Block Transfer instruction is shown in the following diagram.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT	DIR	0	0	0	0	0	0	0	0	AMD 5	AMD 4	AMD 3	AMD 2	AMD 1	AMD 0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	1	NDA 64	NDA 32	NDA 16	NDA 8	NDA 4	NDA 2	NDA 1	0	1	AM 2	AM 1	WS 2	WS 1	AD

2nd Longword:

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

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

3rd Longword:

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

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

VXI/VME Single Inline Write Instruction

The Single Inline Write instruction allows a predetermined data word to be placed in the list and written to the specified VXI/VME address. Most often, this type of instruction is used for module initialization or as a mechanism to execute a VXI/VME write operation in a read list. The format of the VXI/VME Single Inline Write instruction is shown in the following diagram.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INT	DIR	0	0	0	0	0	0	0	0	AMD 5	AMD 4	AMD 3	AMD 2	AMD 1	AMD 0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	1	NDA 64	NDA 32	NDA 16	NDA 8	NDA 4	NDA 2	NDA 1	1	0	AM 2	AM 1	WS 2	WS 1	AD

2nd Longword:

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

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

3rd Longword:

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

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

Special Instructions

There are several special instructions implemented by the 2961. These instructions are not handled by the hardware list processor. When the hardware list processing mechanism encounters any special instructions, it sends a signal to the DSP. The DSP then examines the special instruction and executes the operation.

The only required special instruction that must be used is the HALT instruction. The HALT must be the last list instruction loaded following a valid list. The following chart shows the special instructions implemented by the 2961. The hex data shown corresponds to the first 16-bit word of the special instruction (header).

Special Instruction (hex)	Function
8000	Halt
8040	Addressed Slave Trigger
8041	Broadcast Trigger
8043	Generate Host Interrupt
8070	Load Memory Address Register
8071	Load Transfer Count Register
8072	Set DMA Direction (Reads)
8073	Clear DMA Direction (Writes)
8100	Write Reply FIFO Short
8101	Write Reply FIFO Long

Halt Instruction

The Halt instruction has a value of 8000 hex 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

Addressed Slave Trigger Instruction

The Addressed Slave Trigger instruction has an opcode of 8040 hex and is used to initiate a trigger at an addressed slave. The data sent to the addressed slave is contained in the list instruction. The addressed slave takes the write data and strobes it into the Trigger Source Register. Refer to the individual slave device manuals for further information on the Trigger Source Register.

The following is the format of the Addressed Slave Trigger 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	NDA 64	NDA 32	NDA 16	NDA 8	NDA 4	NDA 2	NDA 1

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	0	0

NODE ADDRESS 64 through 1 specify the node address to be accessed.

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
TD 15	TD 14	TD 13	TD 12	TD 11	TD 10	TD 9	TD 8	TD 7	TD 6	TD 5	TD 4	TD 3	TD 2	TD 1	TD 0

TRIGGER DATA 15 through 0 is the data to be written into the addressed slave Trigger Source Register.

Broadcast Trigger Instruction

The Broadcast Trigger Instruction has an opcode of 8041 hex and is used to generate a Broadcast Trigger Message on the highway. This message is received by all slave devices on the highway. Once a slave device receives this message, it takes the data preloaded in the Broadcast Trigger Mask Register and applies the data to the Trigger Source Register. Any bit that is set to a one in the Broadcast Trigger Mask Register causes the corresponding local trigger to be asserted. Please refer to the individual slave devices operating manual for additional information.

The following shows the format of the Broadcast Trigger 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	0	0	1	0	0	0	0	0	1

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	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Generate Host Interrupt Instruction

The Generate Host Interrupt Instruction has an opcode of 8043 hex and is used to generate an interrupt to host computer. In order for this instruction to generate an interrupt to the VME bus, it must be first enabled in the Interrupt Control/Status Register. If polling is desired, the LIST INTERRUPT bit in the Control/Status Register can be used. The following shows the format for the Generate Host Interrupt 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

Load Memory Address Register Instruction

The Load Memory Address Register instruction has an opcode of 8070 hex and is followed by a longword that contains the data to be loaded. This instruction loads the Memory Address Register used for Direct Memory Accesses (DMA) through the list processor. When the DSP encounters this instruction, it makes sure that all the current DMA activity is complete before loading the data into the Memory Address Register. The following shows the format for the Load Memory Address Register 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	0	0	1	1	1	0	0	0	0

2nd Longword:

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

MEMORY ADDRESS 31 through 2 is the data to be loaded into the Memory Address Register.

Load Total Transfer Count Register Instruction

The Load Total Transfer Count Register instruction has an opcode of 8071 hex and is followed by a longword that contains the data to be loaded. This instruction loads the Total Transfer Count Register used for Direct Memory Accesses (DMA) through the list processor. When the

DSP encounters this instruction, it makes sure that all the current DMA activity is complete before loading the data into the Total Transfer Count Register. The following shows the format for the Load Total Transfer Count Register 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	0	0	1	1	1	0	0	0	1

2nd Longword:

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

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

TOTAL TRANSFER COUNT 31 through 0 is the data to be loaded into the Total Transfer Count Register.

Set DMA Direction Instruction

The Set DMA Direction instruction has an opcode of 8072 hex and is used to set the DMA DIRECTION in the Control/Status Register to a one. Setting this bit to a one sets the direction of DMA transfer from the 2961 to computer memory (reads). When the DSP encounters this instruction, it makes sure that all the current DMA activity is complete before setting the direction bit. The following shows the format for the Set DMA Direction 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	1	1	0	0	1	0

Clear DMA Direction Instruction

The Clear DMA Direction instruction has an opcode of 8073 hex and is used to clear the DMA DIRECTION in the Control/Status Register. Clearing this bit to a zero sets the direction of DMA transfer from the computer memory to the 2961 (writes). When the DSP encounters this instruction, it makes sure that all the current DMA activity is complete before clearing the direction bit. The following shows the format for the Clear DMA Direction 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	1	1	0	0	1	1

Write Reply FIFO Short Instruction

The Write Reply FIFO Short instruction has an opcode of 8100 hex and is used to write 16-bits of data to the Reply FIFO on the 2961. The second longword associated with this command contains the data to write to the FIFO. The Reply FIFO holds all the data received from the highway during read operations on the highway. If desired, the list processor may be instructed to insert a data pattern into the read data path. This can be used to tag data when a branch instruction is used or to realign a read data buffer to longwords. Since the 2961 forms 32-bit read data words before transferring the data to the output FIFO, it may be necessary to include a Write Reply FIFO Short instruction in a list to make sure longword alignment is maintained. For example, if a list operation resulted in an odd number of 16-bit words being transferred, the user must include a Write Reply FIFO Short instruction to ensure that the last 16-bit word is not "stuck" in the 2961. The following diagram shows the format for the Write Reply FIFO Short 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	1	0	0	0	0	0	0	0	0

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

Write Reply FIFO Long Instruction

The Write Reply FIFO Long instruction has an opcode of 8101 hex and is used to write 32-bits of data to the Reply FIFO on the 2961. The second longword associated with this command contains the data to write to the FIFO. The Reply FIFO holds all the data received from the highway during read operations on the highway. If desired, the list processor may be instructed to insert a data pattern into the read data path. This can be used to tag data when a branch instruction is used. The following diagram shows the format for the Write Reply FIFO Long 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	1	0	0	0	0	0	0	0	1

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

CAMAC ACCESS MODES

There are four CAMAC Access Modes available for accessing CAMAC chassis. These modes include Q-Ignore, Q-Stop, Q-Repeat and Q-Scan. When a CAMAC chassis is addressed, the two Access Mode bits, AM2 and AM1, are passed to the chassis controller. The Q-modes operations are actually performed in the chassis controller, not the highway driver.

The following chart shows the various Access Modes based on the binary combinations of the AM bits.

Access Mode 2	Access Mode 1	Access Mode
0	0	Q-Stop
0	1	Q-Ignore
1	0	Q-Repeat
1	1	Q-Scan

The following sections detail the four access modes.

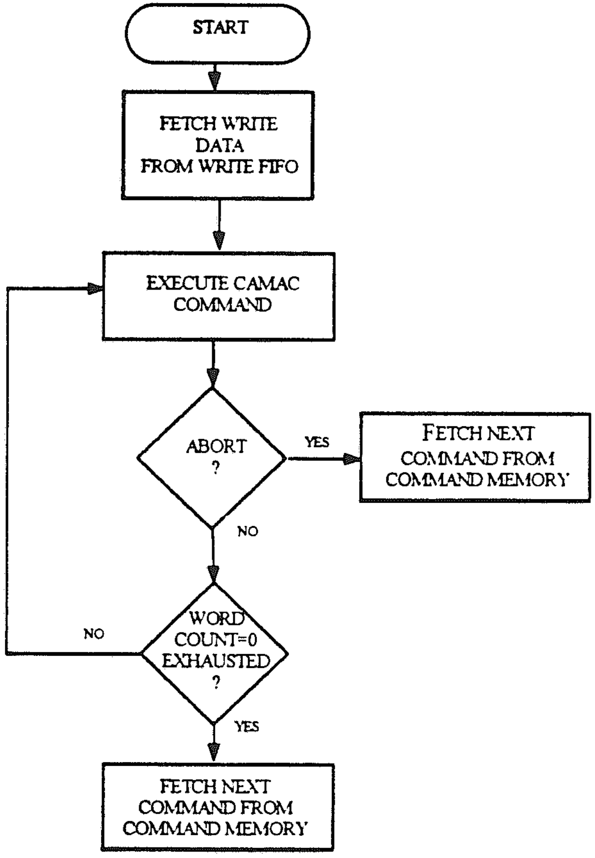
CAMAC Q-Stop Access Mode

The CAMAC Q-Stop Access Mode is selected by setting both the access mode bits to zeros. During Q-Stop operations, the CAMAC 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 NAF without actually knowing the amount of room available within the module. Therefore, the transfer count specification is set large enough to write the entire buffer. If the entire buffer is not sent, due to a NO-Q condition, the transfer count registers on the 2961 and 3972 may be evaluated to determine the number of transfers that did not occur.

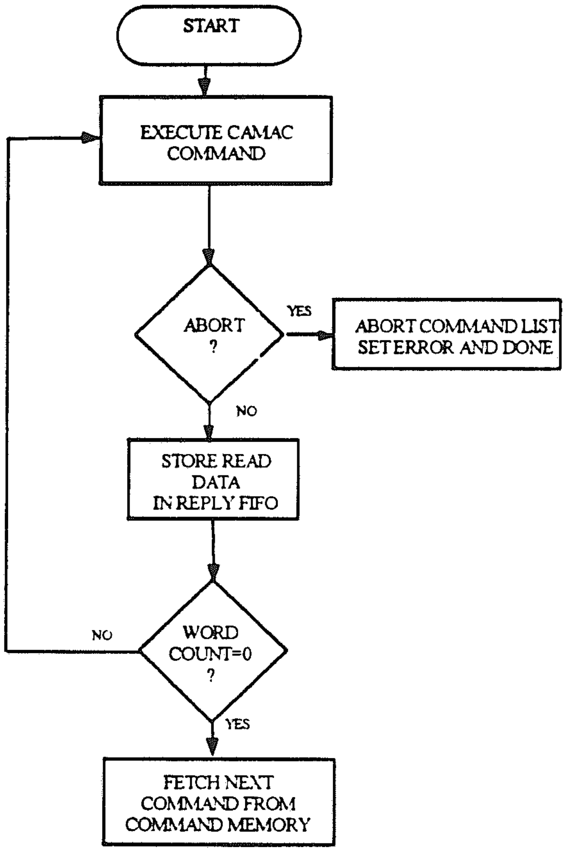
The transfer will also terminate if an error is encountered. The definition of an error for this mode is as follows:

$$\begin{aligned} \text{ERROR} &= \text{NO-Q} \\ &+ \text{NO-X} * \text{AD} \end{aligned}$$

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


START
FETCH WRITE DATA FROM WRITE FIFO
EXECUTE CAMAC COMMAND
ABORT ?
YES
FETCH NEXT COMMAND FROM COMMAND MEMORY
NO
WORD COUNT=0 EXHAUSTED ?
NO
YES
FETCH NEXT COMMAND FROM COMMAND MEMORY

IGNORE - Q WRITE COMMANDS


START
EXECUTE CAMAC COMMAND
ABORT ?
YES
ABORT COMMAND LIST SET ERROR AND DONE
NO
STORE READ DATA IN REPLY FIFO
WORD COUNT=0 ?
NO
YES
FETCH NEXT COMMAND FROM COMMAND MEMORY

IGNORE - Q READ COMMANDS

CAMAC Q-Ignore Access Mode

The CAMAC Q-Ignore Access Mode is selected by setting the AM2 bit to zero and the AM1 bit to a one. During Q-Ignore operations, the CAMAC command (NAF) specified is repeated for each data word until the transfer count is exhausted. This mode does not examine the CAMAC Q-response from the previous transfer to determine if the operation should be terminated. The transfer will terminate if an error is encountered during the operation. The definition of an error for this mode is as follows:

ERROR = NO-X * AD

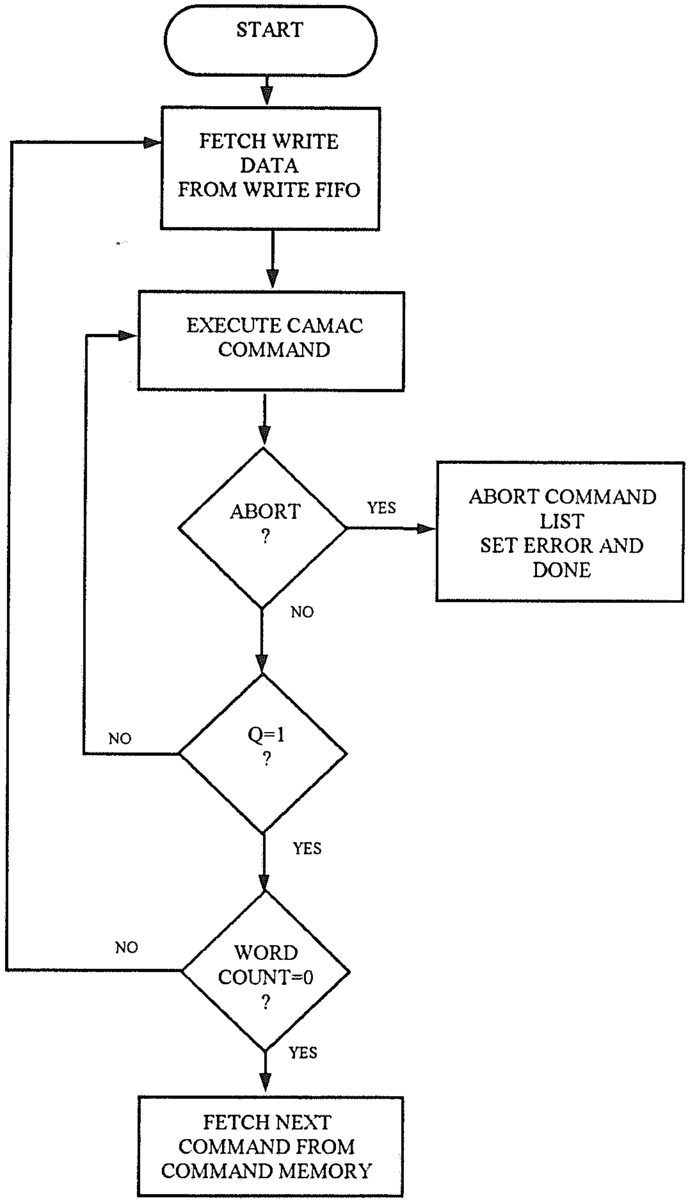
The previous flow diagram shows the sequences the CAMAC Chassis Controller uses for executing Q-Ignore write and read transfers.

CAMAC Q-Repeat Access Mode

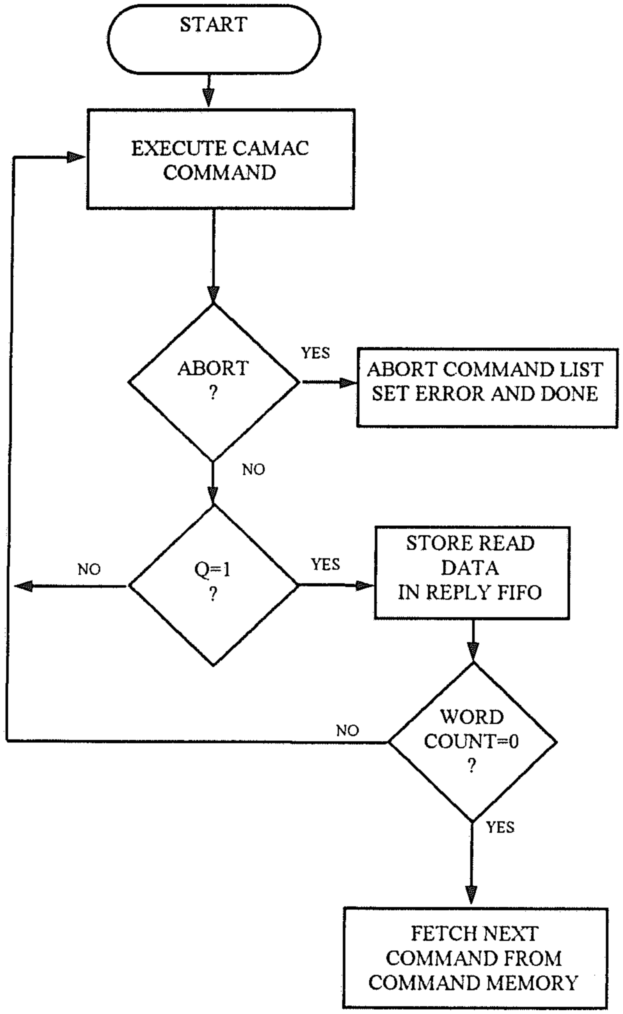
The CAMAC Q-Repeat Access Mode is selected by setting the AM2 bit to a one and the AM1 bit to a zero. During Q-Repeat operations, the CAMAC command (NAF) specified is repeated for each data word until a CAMAC Q-response of one is returned. A Q-response of one either causes new write data to be fetched or read data to be stored. The command is repeated for each data word until the transfer count is exhausted. If a Q-response of one is not obtained within the 200 millisecond timeout period, the transfer terminates with a Q-Repeat Timeout. The following describes an error when executing commands in the Q-repeat mode.

ERROR = Q-REPEAT TIMEOUT
+ NO-X * AD

The following simplified flow chart shows the sequences the CAMAC Chassis Controller used for executing Q-Repeat write and read transfers.


START
FETCH WRITE DATA FROM WRITE FIFO
EXECUTE CAMAC COMMAND
ABORT ?
YES
ABORT COMMAND LIST SET ERROR AND DONE
NO
Q=1 ?
NO
YES
WORD COUNT=0 ?
NO
YES
FETCH NEXT COMMAND FROM COMMAND MEMORY

Q-REPEAT WRITE COMMANDS


START
EXECUTE CAMAC COMMAND
ABORT ?
YES
ABORT COMMAND LIST SET ERROR AND DONE
NO
Q=1 ?
NO
YES
STORE READ DATA IN REPLY FIFO
WORD COUNT=0 ?
NO
YES
FETCH NEXT COMMAND FROM COMMAND MEMORY

Q -REPEAT READ COMMANDS

CAMAC Q-Scan Access Mode

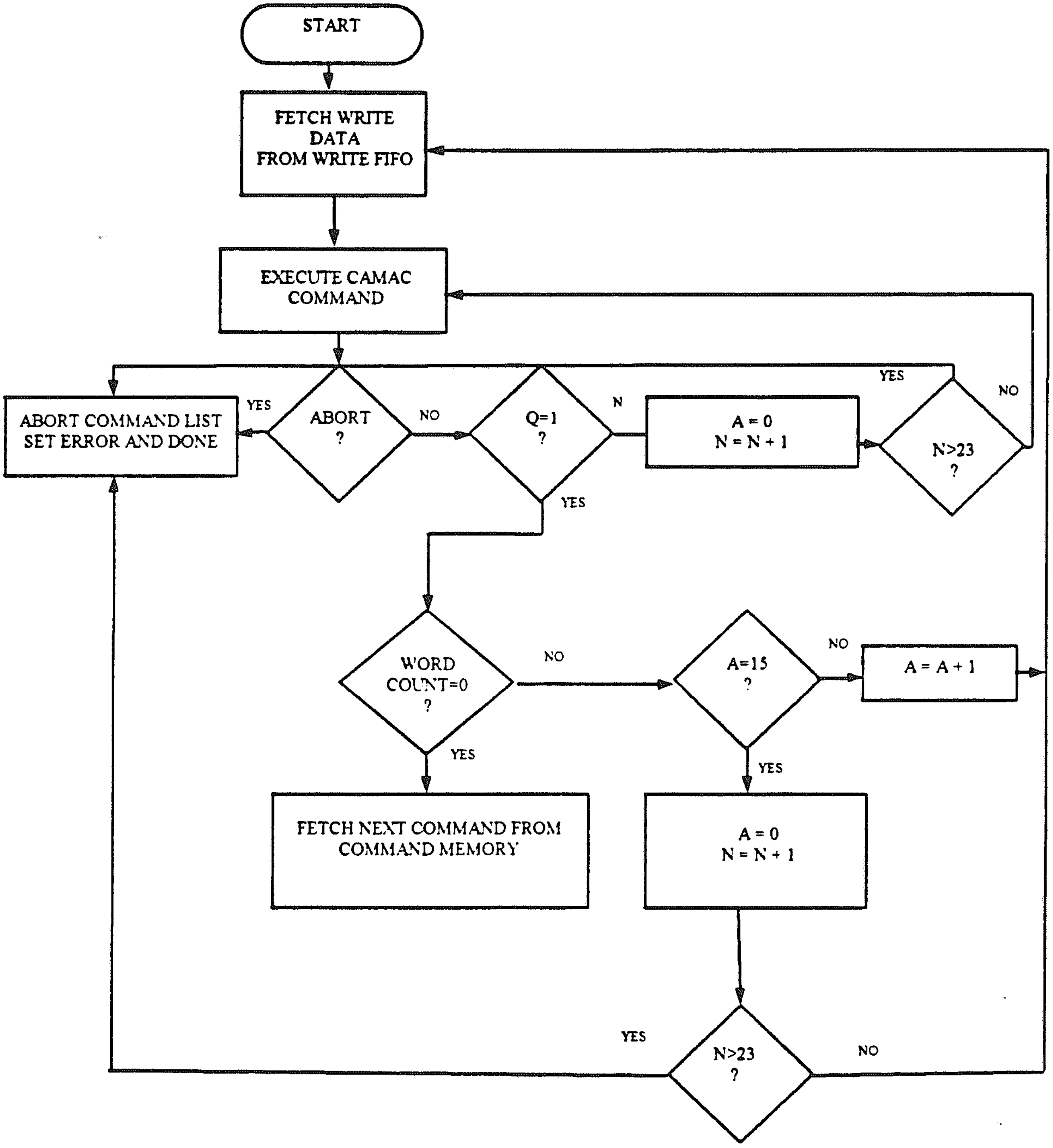
The CAMAC Q-Scan Access Mode is selected by setting the AM2 and AM1 bits to a one. During Q-Scan operations, the CAMAC Chassis Controller uses the Q-response from the previously executed CAMAC command to determine the Station Number (N) and Subaddress (A) for the next operation. A Q-response of zero indicates that the last valid Subaddress of the current Station Number has been accessed. The CAMAC controller responds to the Q=0 response by incrementing the Station Number and resetting the Subaddress to zero. A Q=1 response indicates that the last CAMAC command was executed to a valid address. The controller responds to the Q=1 by either storing the read data or retrieving the next write data word. After the Q=1 is received, the controller updates the CAMAC address as follows: the Subaddress is incremented or, if the Subaddress was 15, it is reset to zero and the Station Number is incremented.

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

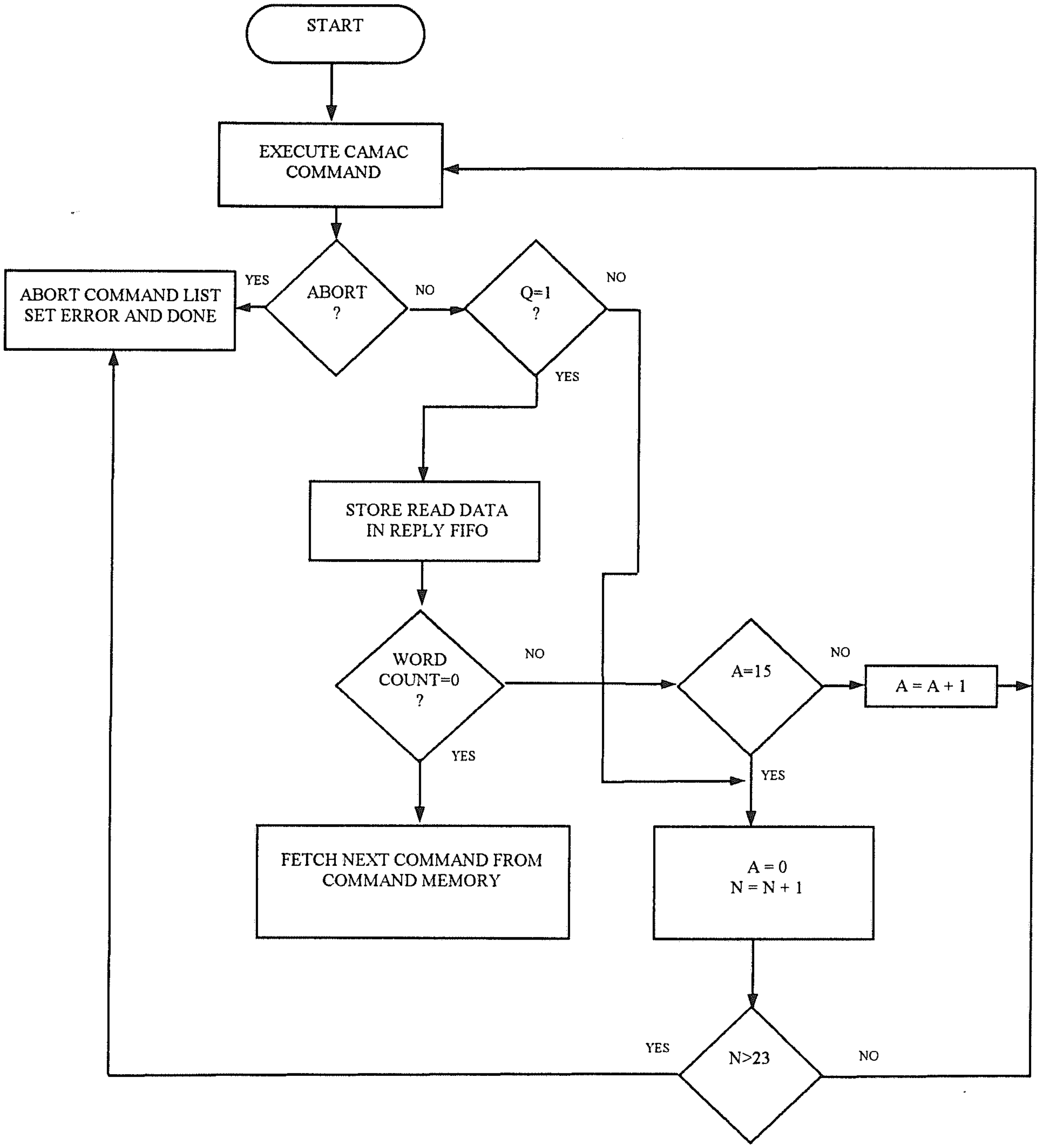
$$\text{ERROR} = N > 23$$

The following diagrams show the basic flow for Q-Scan write and read operations.

Q-Scan Write Operations



Q-Scan Read Operations



CAMAC List Processing Example

As an example, assume it is desired to read a two channel analog-to-digital converter (ADC) located in chassis number 3 slot 6. 1024 samples of the 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 application example, an F(17) A(0) with data = 1 is first executed to select the desired channel. An F(26) A(0) is then executed to enable conversions. A Standard CAMAC Block Transfer operation is executed in the Q-Repeat mode to read the 1024 converted samples. After the samples are read, an F(24) A(0) command is executed to disable the conversions. These steps are then repeated for channel number 2.

For this example, the following commands are executed :

Single Inline CAMAC Write	Node(3) N(6) F(17) A(0) DATA(1)
Single Inline CAMAC Write	Node(3) N(6) F(26) A(0) DATA (not required for Control)
CAMAC Block Transfer	Node(3) N(6) F(2) A(0) WORD COUNT(1024)
Single Inline CAMAC Write	Node(3) N(6) F(24) A(0) DATA(not required for Control)
Single Inline CAMAC Write	Node(3) N(6) F(17) A(0) DATA(2)
Single Inline CAMAC Write	Node(3) N(6) F(26) A(0) DATA (not required for Control)
CAMAC Block Transfer	Node(3) N(6) F(2) A(0) WORD COUNT(1024)
Single Inline CAMAC Write	Node(3) N(6) F(24) A(0) DATA(not required for Control)

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

Hex Instruction Data	List Entry Contents
0C1101C2	N(6) A(0) F(17) Single Inline Write Node (3) Q-Ignore 24-Bit
00000001	DATA(1)
0C1A01C2	N(6) A(0) F(26) Single Inline Write Node (3) Q-Ignore 24-Bit
00000000	DATA (0)
0C0201B2	N(6) A(0) F(2) Block Transfer Node(3) Q-Repeat 24-Bit
FFFFFC00	Transfer Count (-1024)
0C1801C2	N(6) A(0) F(24) Single Inline Write Node (3) Q-Ignore 24-Bit
00000000	DATA (0)
0C1101C2	N(6) A(0) F(17) Single Inline Write Node(3) Q-Ignore 24-Bit

Hex Instruction Data	List Entry Contents
00000002	DATA (2)
0C1A01C2	N(6) A(0) F(26) Single Inline Write Node(3) Q-Ignore 24-Bit
00000000	DATA (0)
0C0201B2	N(6) A(0) F(2) Block Transfer Node(3) Q-Repeat 24-Bit
FFFFFC00	Transfer Count (-1024)
0C1801C2	N(6) A(0) F(24) Single Inline Write Node(3) Q-Ignore 24-Bit
00000000	DATA (0)
00008000	HALT

VXI LIST PROCESSING EXAMPLE

As an example, assume it is desired to setup, enable, and read a block of 20000 32-bits words from a VXI module. For this example, the following parameters exist:

Node Address = 16
Logical Address = 2
A32 Device
Readout address for data is at base address for A32 space

For this example, two Single VXI Inline Write operations are executed to setup the modules Offset Register in A16 address space and then enable the module by setting the A24/A32 enable bit in the Status/Control Register. The module will be set to occupy address 30000000 in A32 address space. After setup, a VXI Block Transfer instruction is executed to read the data.

With a Logical Address of 2, the modules' Status/Control Register is located at C084 hex and the Offset Register is at address C086 hex. Both of these registers are located in A16 address space and may be accessed with an address modifier of 2D hex.

For this example, the following instructions are executed.

Single Inline VXI Write	Address(C086)	Address Modifier(2D)	Data(3000)
Single Inline VXI Write	Address (C084)	Address Modifier (2D)	Data (8000)
Block VXI Read	Address (30000000)	Address Modifier (D)	Transfer Count (-20000)

After the list elements are formed, the actual list is as follows:

Hex Instruction Data	List Entry Contents
002D4840	Address Modifier (2D) VXI Single Inline Write 32-Bit
0000C086	Address (C086)
00003000	DATA(00003000)
002D4840	Address Modifier (2D) VXI Single Inline Write 32-Bit
0000C084	Address (0000C084)
00008000	DATA(00008000)
400D4820	Read, Address Modifier (D) VXI Block Transfer Increment Address 32-Bit
30000000	Address (30000000)
FFFFB1E0	Transfer Count (-20000)
00008000	HALT

INTERRUPTS

The 2961 may generate interrupts to the VME bus from any of four sources. These sources include the assertion of the DONE, the presence of a Demand Message in the Demand FIFO Register, the list processor encountering the Generate Host Interrupt instruction, or by the Buffer Interval Counter overflowing during multibuffer read operations. These sources are individually enabled along with a global interrupt enable bit in the Interrupt Control/Status Register. The request level at which the 2961 interrupts the host computer and Vector asserted during the Inerrupt acknowledge cycle is selected through the Interrupt Control/Status Register.

The DONE interrupt source is generated when a list processing operation is completed and the DONE, bit 7, in the Control/Status Register is set to a one. The DONE INTERRUPT ENABLE, bit 3, in the Interrupt Control/Status Register must be set to a one in order to generate a VME interrupt.

If enabled, the receipt of a Demand Message from the highway can generate a VME interrupt. The interrupt is enabled in the Interrupt Control/Status Register using the DEMAND INTERRUPT ENABLE bit, bit 5. When this interrupt source is generated, the entire contents of the Demand FIFO Register should be read in order to allow additional interrupts to occur.

The list processor may also generate an interrupt to the VME bus. The Generate Host Interrupt instruction asserts the interrupt source and must be enabled using the LIST INTERRUPT ENABLE bit, bit 4, in the Interrupt Control/Status Register.

Another interrupt source from the 2961 is generated during multibuffer read operation. During multibuffer read operations, the Buffer Interval Counter is used to indicate when a segment of the read data buffer has been filled. When this occurs, an interrupt may be generated. This interrupt source is enabled using the MULTIBUFFER INTERRUPT ENABLE bit, bit 6, in the Interrupt Control/Status Register. Please refer to the Multibuffer Operations section of this manual for additional information.

The 2961 can generate a VME interrupt when the INTERRUPT REQUEST ENABLE (INT ENA) bit of the Interrupt Control/Status Register is set to a one. The request level at which the 2961 will interrupt the host computer is selected by INTERRUPT REQUEST LEVEL2-0 (IRQ2-IRQ0), bits 2-0 of the Interrupt Control/Status Register. The Vector the 2961 will put out during the Interrupt acknowledge cycle is selected by VECTOR SELECTION3-0 (VEC3-VEC0), bits 15-12 of the Interrupt Control/Status Register.

After an interrupt is acknowledged by the host, the interrupt sources are automatically cleared. The interrupt source(s) that generated the request are sent out in the least significant 4 bit location of the interrupt vector along with the 4 vector selection bits during the interrupt acknowledge bus cycle.

The following shows the vector that the 2961 asserts during the interrupt acknowledge cycle. Note that at least one bit in this field is set during an acknowledge cycle.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	VEC 3	VEC 2	VEC 1	VEC 0	MBM INT	DMD INT	LST INT	DON INT

To allow an interrupt status bit to be enabled in the Interrupt Control/Status Register, the corresponding enable bit must be set to a one. Interrupt sources are disabled by setting the bit to a zero. In the case where polling is used instead of interrupt generation, the interrupt source bits may be cleared by writing to the ICSR with the data set equal to the source bit to be cleared.

DEMANDS

Demands are asynchronous messages received from devices on the interconnect highway. These messages provide a means of informing the 2961 that the device sourcing the message requires attention. The Demand Message contains the chassis address of the node requesting service along with an 8-bit Demand Identification byte. For the definition of the Demand Identification bits, refer to the manual for the slave device.

After a Demand Message is received by the 2961, it is stored in a 2048 word FIFO. As long as there is at least one demand in the FIFO, the DEMAND PENDING bit in the Control/Status Register is set to a one. After all demand words are read from the FIFO, the bit is reset to a zero.

If the Demand FIFO is full and another Demand Message is received by the 2961, the message is lost and the DEMAND OVERFLOW bit in the Control/Status Register is set to a one. After the overflow occurs, the DEMAND FIFO OVERFLOW bit can be cleared by executing a write to the Demand FIFO Reset Register address.

MULTIBUFFER OPERATIONS

The 2961 may be configured to execute highway read operations automatically and notify the host computer when a buffer a data is available. As the host is reading data from one buffer segment, the 2961 may be filling another. This is referred to as multibuffering.

The highway operations can be executed automatically by enabling the on-board timer through the Timer Control Register. This register controls the frequency at which the list is executed. This rate is referred to as the 'tic' rate. After selecting the frequency in this register, the timer must be enabled.

When multibuffer operations are executed, the read data received from the highway is transferred to the host computer memory via Direct Memory Access (DMA). The DMA operation is initially set up through programmed I/O transfer to the 2961. This establishes the initial DMA address as well as the total number of VME transfers to executed before reloading the memory address and buffer end address registers. The reloading of the memory address and buffer end address registers is enable with the BUFFER END ADDRESS ENABLE (BEA ENA) bit in the Control/Status Registers. When the reloading is enabled, the host computer memory is used as a circular buffer. When the entire DMA buffer has been filled, the 2961 reloads its memory address and buffer end address registers.

Before the multibuffer operation is initiated, the Buffer Interval Counter must be loaded with the count indicating the number of VME transfers that are to occur before setting a multibuffer flag. As an example, assume that the buffer end address (transfer count) specification for the 2961 is 1000. This creates a circular buffer in the host memory of 1000 words. The buffer is to be split into four equal segments. Therefore, the Buffer Interval Counter (BIC) is loaded with 1000/4. Every time the BIC is exhausted, a multibuffer flag bit is set. The flag bits are incrementally set until the transfer count expires and the internal pointer is set to the first flag bit.

When the host sees a flag bit on, it reads out the corresponding segment of data from the buffer and clears the appropriate multibuffer flag by executing a write to the Multibuffer Flag Register. The data for this write must correspond with the flag to clear.

If a flag is set and readout of the buffer segment is not complete by the time the 2961 loops around to the buffer segment again, the FLAG OVERFLOW bit in the Multibuffer Control Register. This is typically caused by the memory buffer being read out too slow.

EXTERNAL CLOCK INPUT

If the clock rate selections on the 2961 do not meet an application requirement for timer initiated list processing, an external clock source may be connected to the 2961. This input is fed into the 2961 through the LEMO connector located on the front panel. This input provided to the 2961 must be TTL compatible. The list is triggered by the high to low transition of the input, and must have a minimum pulse width of 300 nanoseconds.

To enable the external clock input, the Timer Control Register must have both the CLOCK SOURCE and TIMER ENABLE bits set to a one. Refer to the Timer Control Register section of this manual for additional information.

APPENDIX A - 2961 Composite Register Layout

Control/Status Register(CSR) 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR CD3	ERR CD2	ERR CD1	ERR CD0	VXI TMO	QTMO	N>23	ILLG CMD	RMT PER	NO SYNC	0	PAR ERR	TMO	ADNR	NOX	NOQ
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
VME TMO	BUS ERR	LIST INT	DMD OFLO	DMD PND	0	XMT FULL	RCV DAV	DONE	0	0	BEA ENA	DMA DIR	DMA ENA	SUSP	0

Burst Count / Address Modifier Register (BAM) 0x04

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
AM5	AM4	AM3	AM2	AM1	AM0	0	BCT ENA	BCT 7	BCT 6	BCT 5	BCT 4	BCT 3	BCT 2	BCT 1	BCT 0

Interrupt Control/Status Register (ICSR) 0x08

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
VEC 3	VEC 2	VEC 1	VEC 0	MBM INT	DMD INT	LST INT	DON INT	INT ENA	MBM IE	DMD IE	LST IE	DON IE	IRQ 2	IRQ 1	IRQ 0

Timer Control Register (TCR) 0x0C

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

FIFO Data Register (FDR) 0x10

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
FDR 15	FDR 14	FDR 13	FDR 12	FDR 11	FDR 10	FDR 9	FDR 8	FDR 7	FDR 6	FDR 5	FDR 4	FDR 3	FDR 2	FDR 1	FDR 0

Command Memory Address Register (CMA) 0x14

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
LST GO	CMA 14	CMA 13	CMA 12	CMA 11	CMA 10	CMA 9	CMA 8	CMA 7	CMA 6	CMA 5	CMA 4	CMA 3	CMA 2	CMA 1	CMA 0

Command Memory Data Register (CMD) 0x18

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

List Transfer Count Register (LTCR) 0x1C

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

Total Transfer Count Register (TTCR) 0x20

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

Memory Address Register (MAR) 0x24

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

Buffer End Address Register (BEA) 0x28

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

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

Buffer Interval Counter Register (BIC) 0x2C

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

Multibuffer Memory Control Register (MBMCT) 0x30

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	MBM ENA	FLG OFL	FLG 3	FLG 2	FLG 1	FLG 0

Demand FIFO Register (DFR) 0x34

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	NDA 64	NDA 32	NDA 16	NDA 8	NDA 4	NDA 2	NDA 1	DID 7	DID 6	DID 5	DID 4	DID 3	DID 2	DID 1	DID 0

Reset Interface (RSTIFC) 0x38

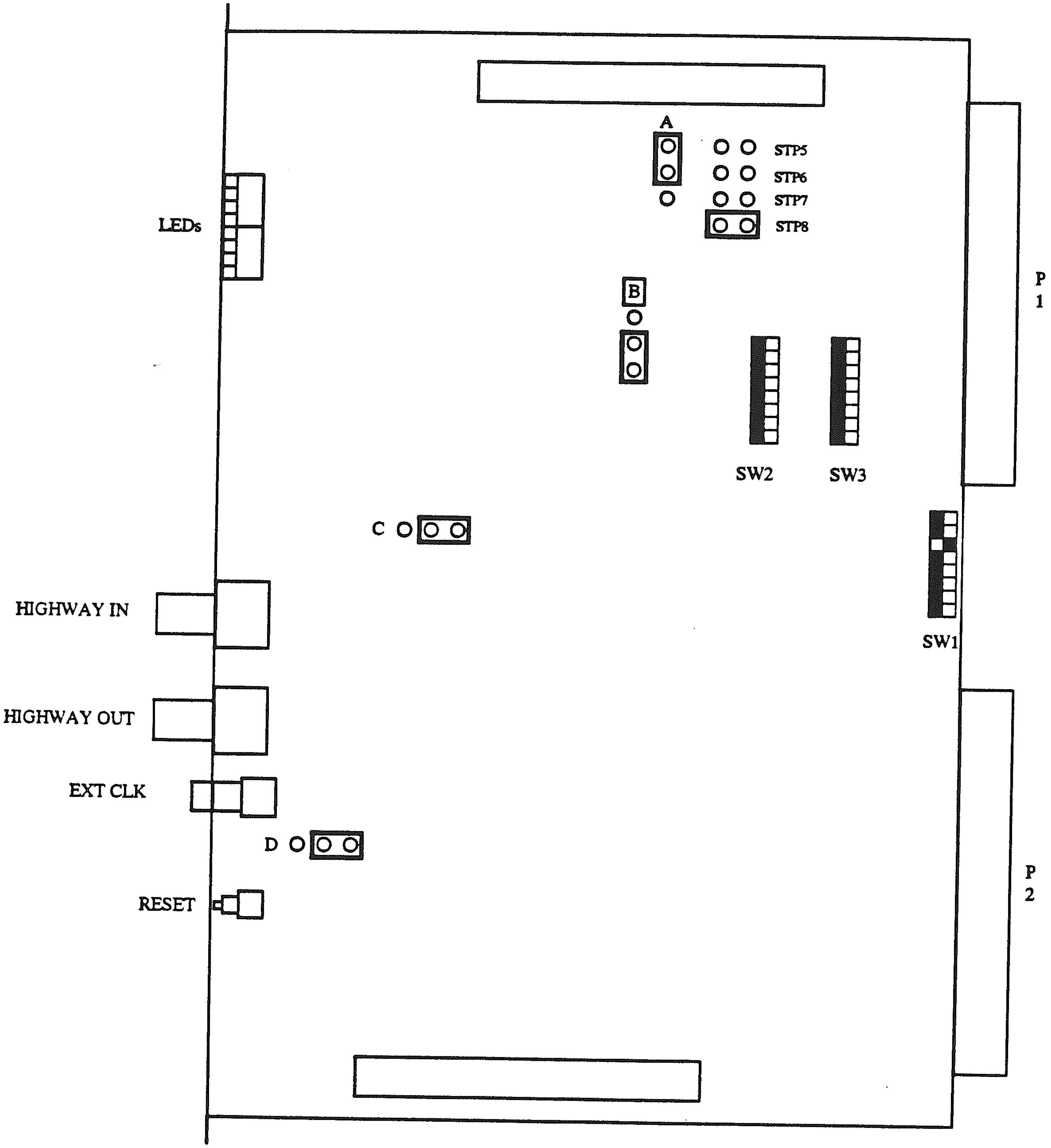
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

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

Communication I/O Register (COMIO) 0x3C

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
COM 15	COM 14	COM 13	COM 12	COM 11	COM 10	COM 9	COM 8	COM 7	COM 6	COM 5	COM 4	COM 3	COM 2	COM 1	COM 0



VME105B Default Strap Locations

2961 Main Card Strap and Switch Locations

WARRANTY

KineticSystems Company, LLC warrants its standard hardware products to be free of defects in workmanship and materials for a period of one year from the date of shipment to the original end user. Software products manufactured by KineticSystems are warranted to conform to the Software Product Description (SPD) applicable at the time of purchase for a period of ninety days from the date of shipment to the original end user. Products purchased for resale by KineticSystems carry the original equipment manufacturer's warranty.

KineticSystems will, at its option, either repair or replace products that prove to be defective in materials or workmanship during the warranty period.

Transportation charges for shipping products to KineticSystems shall be prepaid by the purchaser, while charges for returning the repaired warranty product to the purchaser, if located in the United States, shall be paid by KineticSystems. Return shipment will be made by UPS, where available, unless the purchaser requests a premium method of shipment at their expense. The selected carrier shall not be construed to be the agent of KineticSystems, nor will KineticSystems assume any liability in connection with the services provided by the carrier.

The product warranty may vary outside the United States and does not include shipping, customs clearance, or any other charges. Consult your local authorized representative or reseller for more information regarding specific warranty coverage and shipping details.

PRODUCT SPECIFICATIONS AND DESCRIPTIONS IN THIS DOCUMENT SUBJECT TO CHANGE WITHOUT NOTICE.

KINETICSYSTEMS SPECIFICALLY MAKES NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR ANY OTHER WARRANTY EITHER EXPRESSED OR IMPLIED, EXCEPT AS IS EXPRESSLY SET FORTH HEREIN. PRODUCT FAILURES CREATED BY UNAUTHORIZED MODIFICATIONS, PRODUCT MISUSE, OR IMPROPER INSTALLATION ARE NOT COVERED BY THIS WARRANTY.

THE WARRANTIES PROVIIDED HEREIN ARE THE PURCHASER'S SOLE AND EXCLUSIVE REMEDIES ON ANY CLAIM OF ANY KIND FOR ANY LOSS OR DAMAGE ARISING OUT OF, CONNECTED WITH, OR RESULTING FROM THE USE, PERFORMANCE OR BREACH THEREOF, OR FROM THE DESIGN, MANUFACTURE, SALE, DELIVERY, RESALE, OR REPAIR OR USE OF ANY PRODUCTS COVERED OR FURNISHED BY KINETICSYSTEMS INCLUDING BUT NOT LIMITED TO ANY CLAIM OF NEGLIGENCE OR OTHER TORTIOUS BREACH, SHALL BE THE REPAIR OR REPLACEMENT, FOB FACTORY, AS KINETICSYSTEMS MAY ELECT, OF THE PRODUCT OR PART THEREOF GIVING RISE TO SUCH CLAIM, EXCEPT THAT KINETICSYSTEMS' LIABILITY FOR SUCH REPAIR OR REPLACEMENT SHALL IN NO EVENT EXCEED THE CONTRACT PRICE ALLOCABLE TO THE PRODUCTS OR PART THEROF WHICH GIVES RISE TO THE CLAIM. IN NO EVENT SHALL KINETICSYSTEMS BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOSS OF PROFITS.

Products will not be accepted for credit or exchange without the prior written approval of KineticSystems. If it is necessary to return a product for repair, replacement or exchange, a Return Authorization (RA) Number must first be obtained from the Repair Service Center prior to shipping the product to KineticSystems. The following steps should be taken before returning any product:

1. Contact KineticSystems and discuss the problem with a Technical Service Engineer.
2. Obtain a Return Authorization (RA) Number.
3. Initiate a purchase order for the estimated repair charge if the product is out of warranty.
4. Include a description of the problem and your technical contact person with the product.
5. Ship the product prepaid with the RA Number marked on the outside of the package to:

KineticSystems Company, LLC
Repair Service Center
900 North State Street
Lockport, IL 60441

Telephone: (815) 838-0005
Facsimile: (815) 838-4424
Email: tech-serv@kscorp.com