# Model V160-xyz3

Interconnect Slot-0 Controller

## User's Manual

August 27, 2004

# CONTENTS

## LIST OF FIGURES



## LIST OF TABLES

# Interconnect VXI Slot-0 Controller

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

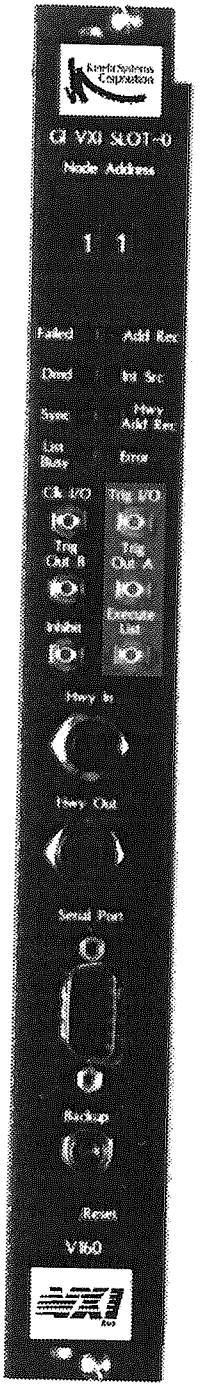
# V160

### Features

- **Provides a VXIbus interface for the Interconnect Highway**
- **Up to 126 controllers per Interconnect network**
- **Provides full VXI Slot-0 functionality**
- **Provides full throughput with a 2 km maximum distance between fiber-optic nodes**
- **Provides up to 10 Mbyte/s highway throughput**
- **Includes a high-speed command processor**
- **Supports buffer memory options**
- **TTL Trigger Expansion**

### Typical Applications

- **Jet aircraft engine test cells**
- **Rocket engine test cells**
- **Missile testing**
- **Wind tunnel data acquisition and control**
- **Aerospace testing**
- **Nuclear accelerator control and monitoring**
- **High-performance ATE**
- **Multi-mainframe VXIbus systems**
- **Distributed VXIbus systems**
- **Systems requiring galvanic isolation**




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

The Grand Interconnect connects multiple I/O chassis to a host computer, providing a high-throughput, deterministic data acquisition and control system. A complete Grand Interconnect system includes an Interconnect Host Adapter (IHA), up to 126 I/O nodes that may include VXI Slot-0 controllers and/or CAMAC (IEEE 583) crate controllers, and a fiber-optic highway. The system supports distances between nodes up to 2 km (6560 ft).

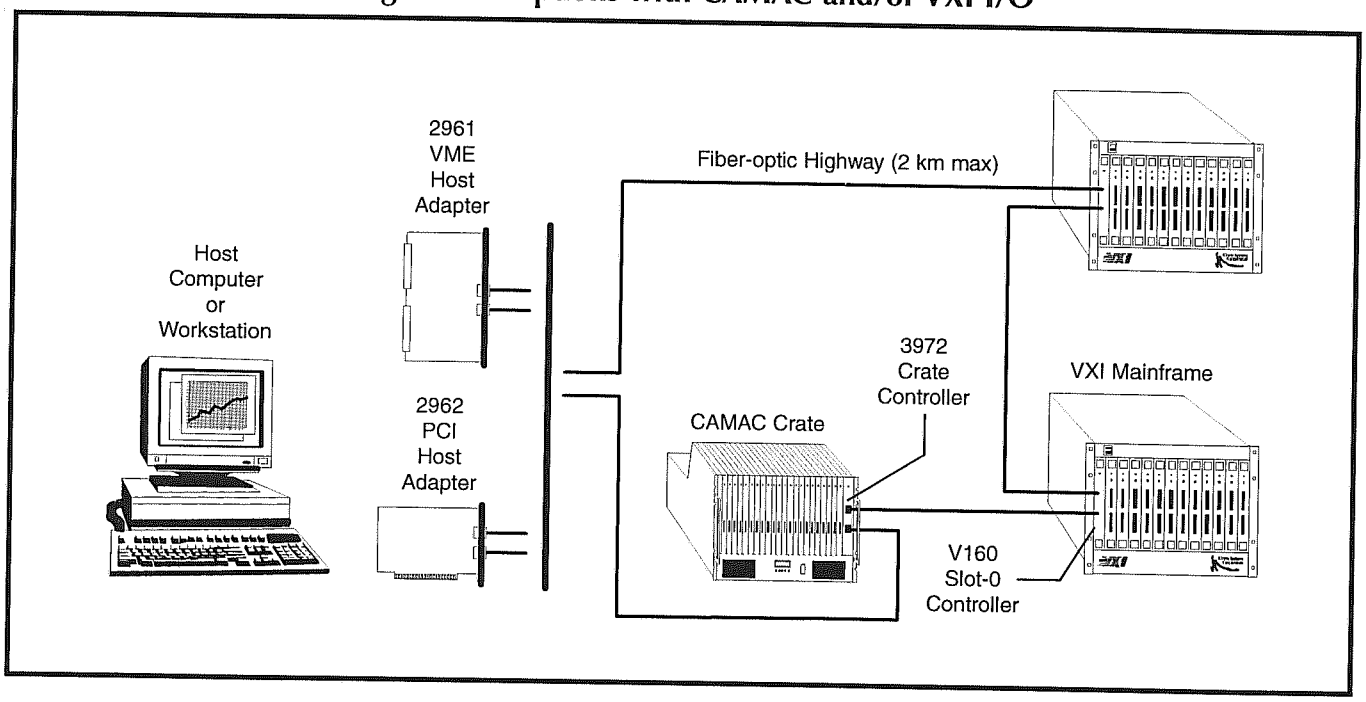
The V160 is a single-width, C-size, VXIbus controller that interfaces the Interconnect Highway to a VXI chassis. It is a slave device on the highway and receives its commands from an Interconnect Host Adapter. This module meets all the requirements of a VXIbus Slot-0 controller, including a MODID register for geographic addressing, a "CLK10" 10 MHz source, TTL/ECL trigger functions, and VXI interrupt handling. Interrupts generated in the VXI chassis are acknowledged by the V160, causing a demand message to be transmitted over the highway to notify the host.

The V160 communicates with the VXIbus using A32, A24, or A16 addressing modes as well as D32, D16, or D08 (EO) data transfers. All access via VXIbus is performed using register-based commands. This module includes a 16 MHz clock driver, a data transfer bus priority arbiter, and an interrupt handler.

List processing is included to allow the V160 to execute a sequence of preloaded write or read commands to selected VXI module registers. The list can be triggered by a software command from the Interconnect Host Adapter (IHA); an internal, crystal-controlled, programmable counter/timer; an externally supplied trigger source; a preselected interrupt; or a TTL/ECL trigger line. During setup the list is downloaded over the highway. Once execution begins, data received from VXIbus read operations is stored in a 2 kbyte FIFO memory for transmission to the host computer. For applications that require more than 2 kbytes of storage for timer-initiated lists, a 4 Mbyte buffer memory option (V160-wB22) is available. For VXIbus write operations, data may be embedded in the list or extracted from a FIFO memory.

The Interconnect Host Adapter also contains a list processor. Its list memory can be used for all I/O transfers. With this approach, both the I/O register addresses and the data must be transmitted over the highway which reduces overall throughput. However, a high-performance list operation for a multi-chassis system can be obtained by using an instruction in the IHA's list processor to transfer a block of read or write data for all operations in a chassis; then the list processor in each V160 can select the sequence of I/O registers to be read or written. This causes only data to be transmitted over the highway, resulting in maximum I/O throughput. The full 10 Mbyte/s data throughput can be achieved by the V160, limited only by the IHA and VXI module transfer capabilities. The I/O list can be driven by a programmable timer with rates ranging from less than 0.06 Hz to 500 kHz.

## Grand Interconnect Configuration Options with CAMAC and/or VXI I/O




| Item | Specification | | |
|---|---|---|---|
| Connectors | | | |
| Fiber Optic (Hwy In and Hwy Out) | ST-Type Connectors | | |
| Serial Port (Test interface) | 9P "D" | | |
| Clock I/O | SMB (V160-Axyz) | | |
| Trigger I/O | SMB (V160-Axyz) | | |
| Trigger Out (A and B) | SMB (V160-Axyz) | | |
| Inhibit | SMB (V160-Axyz) | | |
| Execute List | SMB (V160-Axyz) | | |
| Trigger Expansion | 26 Position High Density Connector (V160-Bxyz) | | |
| Backup | 5.54 mm Power Connector | | |
| Power Requirements | **+5 V** | **-5.2 V** | **-2 V** |
| AA1z | 6.7 A | 250 mA | 250 mA |
| AByz | 8.5 A | 250 mA | 250 mA |
| BA1z | 7.0 A | 250 mA | 250 mA |
| BByz | 8.8 A | 250 mA | 250 mA |
| Environmental and Mechanical | | | |
| Temperature range | | | |
| Operational | 0°C to 50°C | | |
| Storage | -25°C to +75°C | | |
| Relative humidity | 0 to 85%, non-condensing, to 40°C | | |
| Cooling requirements | 10 CFM | | |
| Dimensions | 340 mm x 233.35 mm x 30.48 mm (C-size VXIbus) | | |
| Front-panel potential | Chassis ground | | |

## Ordering Information

Model V160-AA13 Interconnect VXI Slot-0 Controller
Model V160-AB13 Interconnect VXI Slot-0 Controller with 1 Mbyte Memory
Model V160-AB23 Interconnect VXI Slot-0 Controller with 4 Mbyte Memory
Model V160-BA13 Interconnect VXI Slot-0 Controller with Trigger Line Interface
Model V160-BB13 Interconnect VXI Slot-0 Controller with 1 Mbyte Memory and Trigger Line Interface
Model V160-BB23 Interconnect VXI Slot-0 Controller with 4 Mbyte Memory and Trigger Line Interface

Model V160-0001 4 Mbyte Memory Factory Upgrade
Model V160-0002 1 Mbyte Memory Factory Upgrade

## Related Products

| | |
|---|---|
| Model 2961 | VME Interconnect Host Adapter |
| Model 2962 | PCI Interconnect Host Adapter |
| Model 3972 | Interconnect CAMAC Crate Controller |
| Model 5802-Lxyz | Cable—50 µm Fiber-optic |
| Model 5802-Nxyz | Cable—62.5 µm Fiber-optic |
| Model 5856-Bxyz | Cable—9S "D" to Unterminated |
| Model 5919-Z1A | Connector—SMB Cable-type |
| Model 5930-Z1A | Connector—9S "D" |


Phone: (815) 838-0005 • FAX: (815) 838-4424 • E-mail: mkt-info@kscorp.com • Web: http://www.kscorp.com

## UNPACKING AND INSTALLATION

The Model V160-xyz3 is shipped in an anti-static bag within a Styrofoam packing container. Carefully remove the module from its anti-static bag and prepare to set the various options to conform to the operating environment.

### LOGICAL ADDRESS SWITCHES

The V160-xyz3 is a Slot 0 capable device. It is shipped from the factory configured for Slot 0 operation with its Logical Address Switches statically configured for Logical Address 0. If the V160-xyz3 is to operate as the Slot 0 device, then its Logical Address must be set for Logical Address 0. If it is not, these switches must be set for Logical Address 1 to 255. Logical Address 255 can be shared by multiple devices in a system that supports Dynamic Configuration. If a non-Slot 0 V160-xyz3 is to be used in a system that does not support Dynamic Configuration, or in a system where static configuration of the module is desired, the Logical Address must be manually established. This is accomplished by manipulating the eight rocker switches located under the access hole in the module's right-side ground shield. (Refer to Figure 1.)

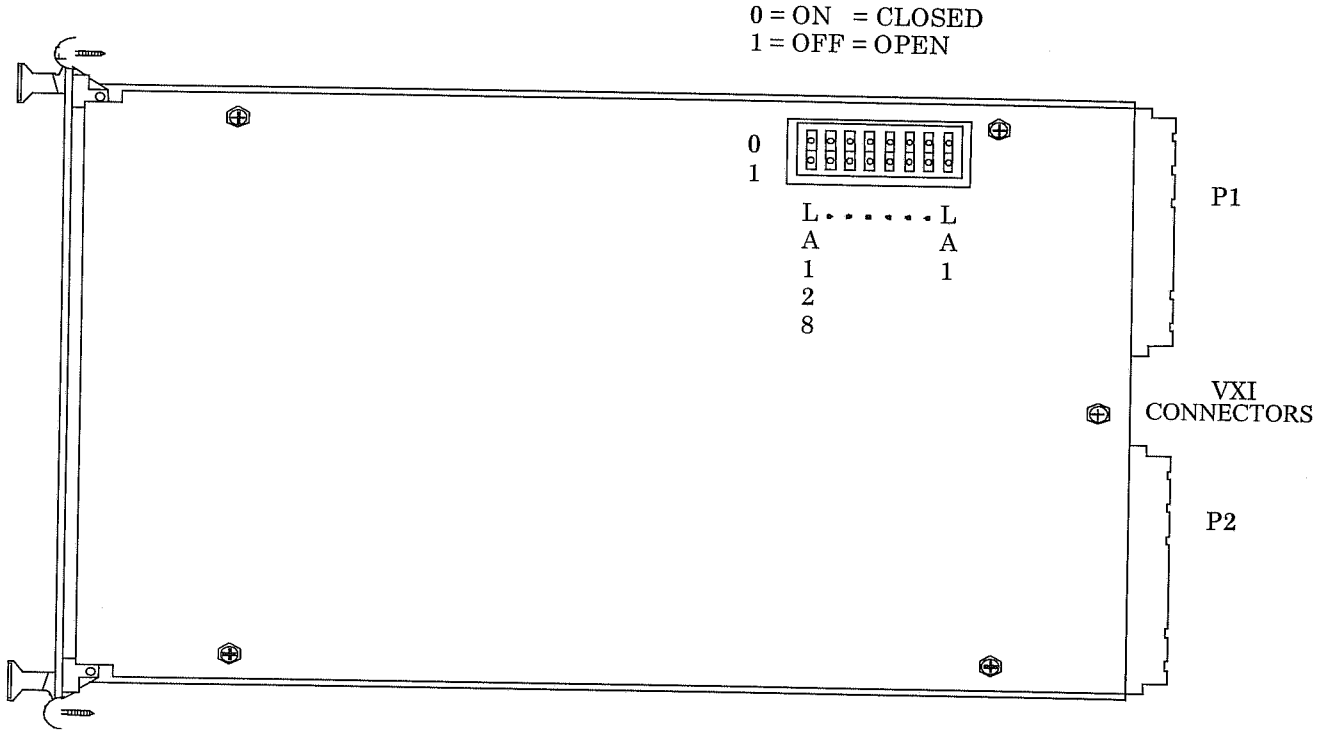



**Figure - 1 V160-xyz3 Logical Address Switches**

These switches represent a binary combination of numbers that range from zero to 255. Use a scribe or other appropriate instrument to set the Logical Address to the desired value. A switch pushed down on the OPEN side is ON (1). For a Logical Address of zero, all switches should be pushed down on the closed (top as viewed in Figure 1) side.

The Logical Address is used to define the A16 base address of the V160-xyz3's VXIbus defined configuration registers. Further explanation on calculating this base address can be found in the section on VXIbus Configuration Registers.

## MODULE INSERTION

**Depending on the version of the V160 there are either straps loaded or switches loaded for the configuration of the device.**

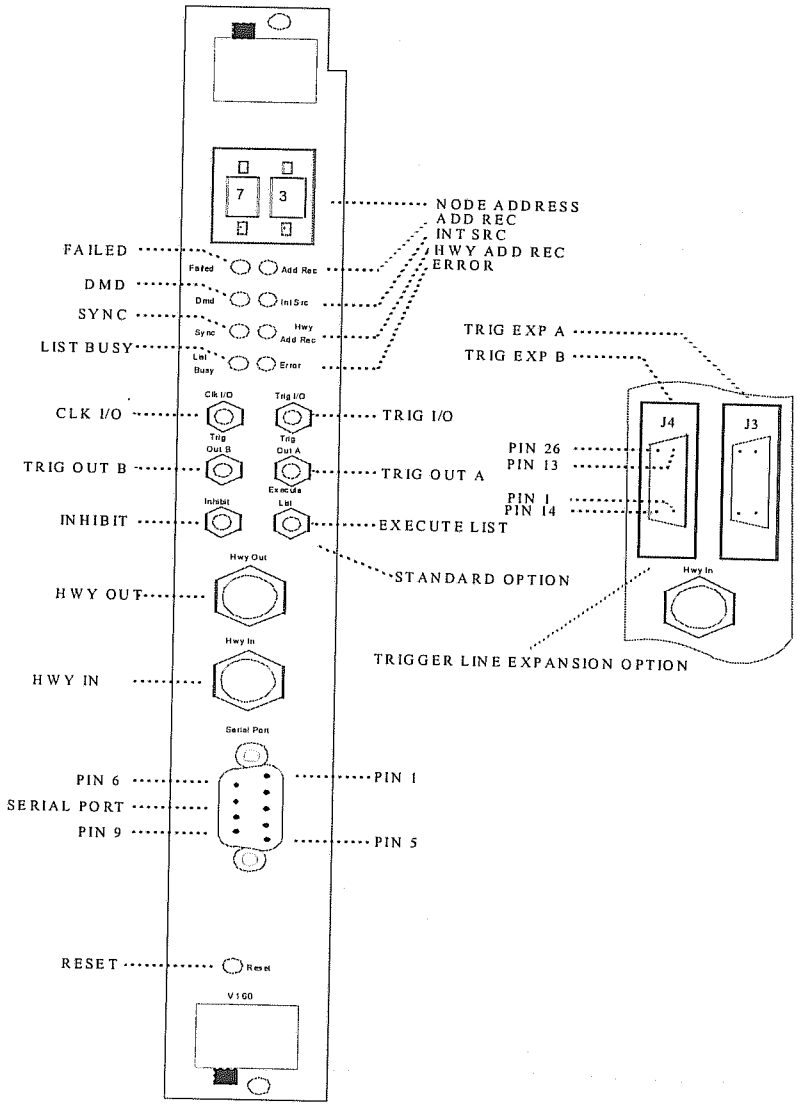
The V160-xyz3 is a C-sized, single width VXIbus module. This module is shipped from the factory configured for Slot 0 operation. Installing a Slot 0 configured V160-xyz3 in any other slot (other than Slot 0) can result in damage to it and other components in the system. Refer to the section on V160-xyz3 Slot 0 Configuration for more information.

**CAUTION:** **TURN OFF MAINFRAME POWER BEFORE INSERTING OR REMOVING MODULE.**

**WARNING:** **REMEMBER TO REMOVE THE INTERRUPT ACKNOWLEDGE AND BUS GRANT DAISY CHAIN JUMPERS BEFORE INSERTING MODULE.**

The VXIbus backplane must be properly configured before inserting a module and applying power. The Bus Grant and Interrupt Acknowledge daisy chain jumpers should be installed in any unoccupied slot.

**Figure 2 - V160-xyz3 Front Panel Layout**



## FRONT PANEL INFORMATION

This section describes the connectors, status LEDs, etc. that appear on the front panel and their functions.

### NODE ADDRESS SWITCH

The Node Address Switch establishes the V160-xyz3's highway node address. This switch contains two digits in hexadecimal format. Each slave on the highway must have a unique node address, ranging from

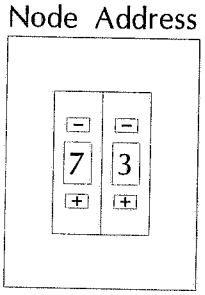



**Figure 3 - V160 Node Address Switch**

1 ($01_{16}$) to 127 ($7F_{16}$). Node address 0 ($00_{16}$) is reserved and should not be used.

Figure 3 shows the node address switch set to a value of $73_{16}$ (115). Notice the most significant digit is on the left and the least is on the right.

LEDS

The V160-xyz3 has eight LEDs located on the front panel to display status information. All LEDs are one-shot extended. Table 1 lists each LED and defines when they are active (lighted).

**Table 1. Status LED Definitions**

| LED Mnemonic (Color) | Description |
|---|---|
| Failed (R) | Failed - Onboard DSP is executing or has failed its self-test. |
| Add Rec (G) | Address Recognized - Transfer to V160's Configuration Registers was successful. |
| Dmd (R) | Demand - A demand message has been placed in the Demand FIFO. |
| Int Src (R) | Interrupt Source - V160 is generating an interrupt source signal. |
| Sync (G) | Synchronization - Indicates valid highway activity or synchronization message. |
| Hwy Add Rec (G) | Highway Address Recognized - V160 recognizes that it is the addressed node. |
| List Busy (G) | List Busy - List Processor is executing instructions from the List Memory. |
| Error (R) | Error - A parity error has occurred on the highway. |

## OPTION SPECIFIC CONNECTORS

The V160-xyz3 has a set of front panel connectors used for clocks and triggers, which differ depending on the option. The V160-Ayz3 has six SMB connectors, while the V160-Byz3 has two 26-pin SCSI II type connectors.

## SMB CONNECTOR OPTION

All V160-xyz3 options have front panel mounted SMBs. Table 2 describes the function of each.

**Table 2. SMB Connector Descriptions**

| Mnemonic | Description |
|---|---|
| Clk I/O | 10 MHz system clock - Can be configured with straps for input or output. |
| Trig I/O | Trigger Input/Output - Configured with straps to route a TTL Trigger Line in/out. |
| Trig Out B | Internal Trigger Output B signal - See Internal Register description. |
| Trig Out A | Internal Trigger Output A signal - See Internal Register description. |
| Inhibit | Reflects the state of Inhibit bit in the Control/Status Register (Internal Registers). |
| Execute List | Input trigger that may be used to start list execution. |

Each of these signals appears on the center pin of the SMB connector. The shell of these connectors is connected to module ground.

## TRIGGER EXPANSION CONNECTOR OPTION

V160-Byz3 options have two 26-position SCSI II type socket connectors that replace the SMB connectors and contains some additional signals. This series of options buffers and routes the VXIbus TTL Trigger to these front panel connectors.

**Table 3. V160-xyz3 Trigger Expansion Pinouts**

| Trigger Expansion B - J4 | | | Trigger Expansion A - J3 | |
|---|---|---|---|---|
| Pin Description | Pin Number(s) | | Pin Description | Pin Number(s) |
| TTL Trigger 0 I/O | 1 | | TTL Trigger 0 I/O | 1 |
| TTL Trigger 1 I/O | 2 | | TTL Trigger 1 I/O | 2 |
| TTL Trigger 2 I/O | 3 | | TTL Trigger 2 I/O | 3 |
| TTL Trigger 3 I/O | 4 | | TTL Trigger 3 I/O | 4 |
| TTL Trigger 4 I/O | 5 | | TTL Trigger 4 I/O | 5 |
| TTL Trigger 5 I/O | 6 | | TTL Trigger 5 I/O | 6 |
| TTL Trigger 6 I/O | 7 | | TTL Trigger 6 I/O | 7 |
| TTL Trigger 7 I/O | 8 | | TTL Trigger 7 I/O | 8 |
| CLK10 Input | 9 | | CLK10 Output | 9 |

| Trigger Expansion B - J4 | |
|---|---|
| **Pin Description** | **Pin Number(s)** |
| *No Connection | 10 |
| *No Connection | 11 |
| Execute List Input | 12 |
| *No Connection | 13 |
| Ground | 14-26 |

| Trigger Expansion A - J3 | |
|---|---|
| **Pin Description** | **Pin Number(s)** |
| Trigger A Output | 10 |
| Trigger B Output | 11 |
| Execute List Output | 12 |
| Inhibit Output | 13 |
| Ground | 14-26 |

* No internal connection is made.

Expansion connector part number - AMP 749611-1.
Mating 26 position male cable connector - Honda PCS-XE26M ( #28 or #30 AWG discrete wire ).
Backshell for mating connector - Honda PCS-XE26LA or PCS-XE26LKA.

## FIBEROPTIC

The V160-xyz3's two front panel mounted ST type connectors serve as the input and output for the fiberoptic highway. The connector labeled Hwy In is the highway input. This input should be connected to the highway output connector of the driver or another slave. Conversely, the Hwy Out is the output connector and should be connected to a driver or slave highway input connector. Fiberoptic highway connections should be made using 62.5/100µm fiberoptic cables.

## SERIAL PORT

The serial port connector is a 9-position subminiature D type connector with pins. It is used as a general purpose serial interface to the V160-xyz3. For more information on the serial port, see the section on Serial Port Functions. The connector's pinout is shown in Table 4. Any standard 9-position subminiature D type socket connector may be used as a mating connector.

**Table 4. Serial Port Connector Pinout**

| Pin Number | Pin Description |
|---|---|
| 2 | Receive (RXD*) |
| 3 | Transmit (TXD*) |
| 5 | Ground (GND) |
| 1, 4, 6, 9 | No internal connection. |

## VXIBUS CONNECTORS

The V160-xyz3 is a C-sized VXIbus module and uses both the P1 and P2 connectors. Tables 5 and 6 list the connections made.

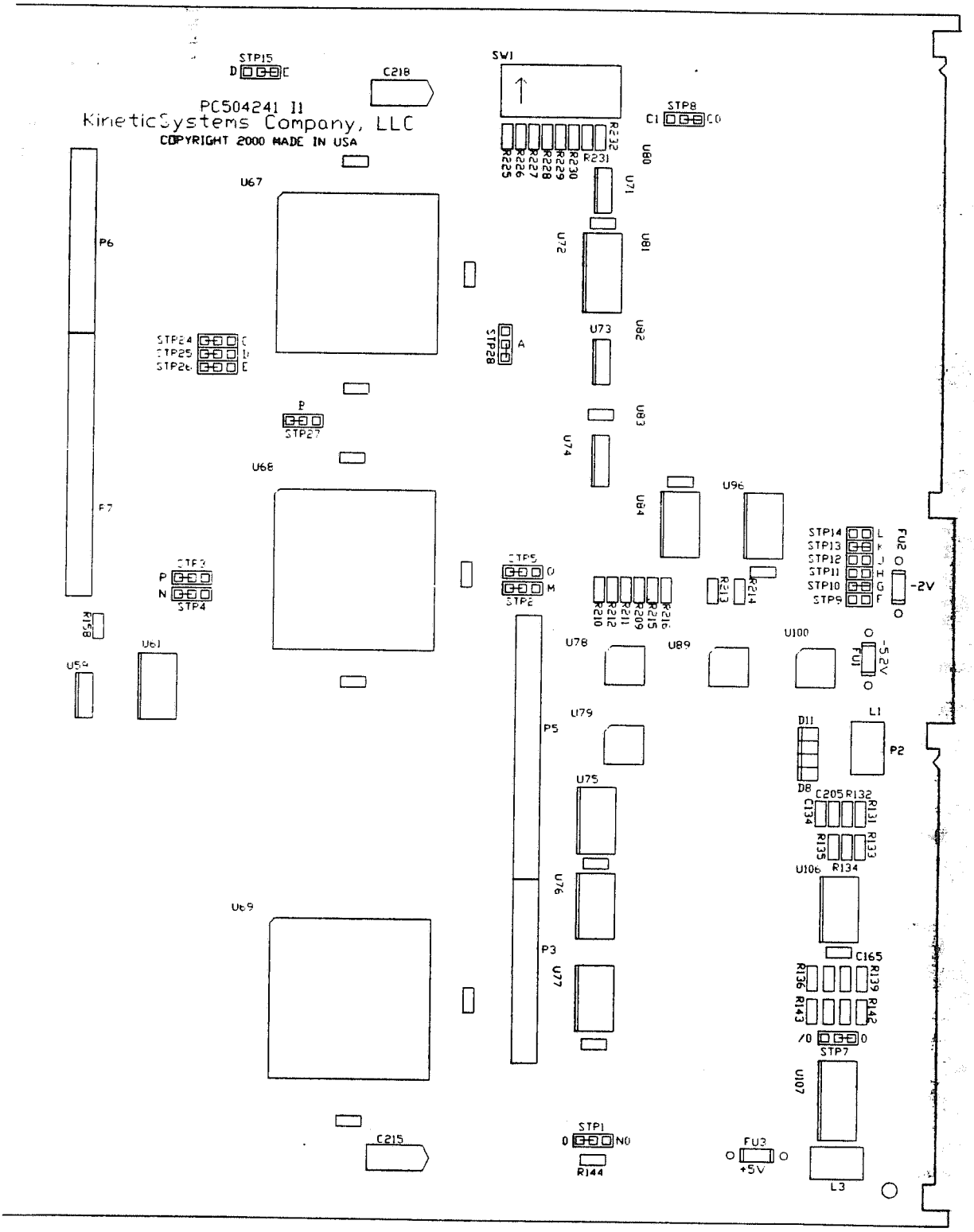
**Table 5. VXIbus P1 Connections**

| Pin | Row A | Row B | Row C |
|---|---|---|---|
| 1 | D00 | BBSY* | D08 |
| 2 | D01 | No Connection | D09 |
| 3 | D02 | No Connection | D10 |
| 4 | D03 | BG0IN* | D11 |
| 5 | D04 | BG0OUT* | D12 |
| 6 | D05 | BG1IN* | D13 |
| 7 | D06 | BG1OUT* | D14 |
| 8 | D07 | BG2IN* | D15 |
| 9 | GND | BG2OUT* | GND |
| 10 | SYSCLK | BG3IN* | SYSFAIL* |
| 11 | GND | BG3OUT* | BERR* |
| 12 | DS1* | BR0* | SYSRESET* |
| 13 | DS0* | BR1* | LWORD* |
| 14 | WRITE* | BR2* | AM5 |
| 15 | GND | BR3* | A23 |
| 16 | DTACK* | AM0 | A22 |
| 17 | GND | AM1 | A21 |
| 18 | AS* | AM2 | A20 |
| 19 | GND | AM3 | A19 |
| 20 | IACK* | GND | A18 |
| 21 | IACKIN* | No Connection | A17 |
| 22 | IACKOUT* | No Connection | A16 |
| 23 | AM4 | GND | A15 |
| 24 | A07 | IRQ7* | A14 |
| 25 | A06 | IRQ6* | A13 |
| 26 | A05 | IRQ5* | A12 |
| 27 | A04 | IRQ4* | A11 |
| 28 | A03 | IRQ3* | A10 |
| 29 | A02 | IRQ2* | A09 |
| 30 | A01 | IRQ1* | A08 |
| 31 | No Connection | No Connection | No Connection |
| 32 | +5 V | +5 V | +5V |

**Table 6. VXIbus P2 Connections**

| Pin | Row A | Row B | Row C |
|---|---|---|---|
| 1 | ECLTRG0 | +5 V | CLK10+ |
| 2 | -2 V | GND | CLK10- |
| 3 | ECLTRG1 | No Connection | GND |
| 4 | GND | A24 | -5.2 V |
| 5 | MODID12 | A25 | LBUSC00 |
| 6 | MODID11 | A26 | LBUSC01 |
| 7 | -5.2 V | A27 | GND |
| 8 | MODID10 | A28 | LBUSC02 |
| 9 | MODID09 | A29 | LBUSC03 |
| 10 | GND | A30 | GND |
| 11 | MODID08 | A31 | LBUSC04 |
| 12 | MODID07 | GND | LBUSC05 |
| 13 | -5.2 V | +5 V | -2 V |
| 14 | MODID06 | D16 | LBUSC06 |
| 15 | MODID05 | D17 | LBUSC07 |
| 16 | GND | D18 | GND |
| 17 | MODID04 | D19 | LBUSC08 |
| 18 | MODID03 | D20 | LBUSC09 |
| 19 | -5.2 V | D21 | -5.2 V |
| 20 | MODID02 | D22 | LBUSC10 |
| 21 | MODID01 | D23 | LBUSC11 |
| 22 | GND | GND | GND |
| 23 | TTLTRG0* | D24 | TTLTRG1* |
| 24 | TTLTRG2* | D25 | TTLTRG3* |
| 25 | +5 V | D26 | GND |
| 26 | TTLTRG4* | D27 | TTLTRG5* |
| 27 | TTLTRG6* | D28 | TTLTRG7* |
| 28 | GND | D29 | GND |
| 29 | No Connection | D30 | No Connection |
| 30 | MODID00 | D31 | GND |
| 31 | GND | GND | No Connection |
| 32 | No Connection | +5 V | No Connection |

**Figure 4 - V160 -XY3 Strapping Information**



## STRAPPING INFORMATION

Figure 4 shows the factory-configured positions for straps A-E and STP 1,8, the TTL Trigger Line and CLK 10 selections, and the SYSRESE. The configuration is as follows:

1.) V160-xyz3 is a Slot 0 device.
2.) V160-xyz3 is an Interrupt Handler.
3.) Bus Request Level is set to 3.
4.) Backplane CLK 10 signal is an output on front panel Clk I/O.
5.) Positions 1 through 8 on SW1 are pushed down on top (set to zeros) so V160-xyz3 is at Logical Address zero.
6.) V160-xyz3 drives backplane Clk10+/-.
7.) TTL Trigger Line 0 is routed to front panel Trig I/O as an output.
8.) V160-xyz3 has pull-up enabled for Slot 0 MODID line.
9.) V160-xyz3 drives the backplane SYSRESET signal.

**Table 7. - Strap Configuration**

| Label | Reference Designator | Function |
|---|---|---|
| A | STP28 | * Arbiter Enable |
| B | STP27 | * Interrupt Handler |
| C | STP24 | * Slot 0 |
| D | STP25 | Bus Request Level |
| E | STP26 | |
| F | STP9 | * CLK10 +/- |
| G | STP10 | |
| H | STP11 | |
| J | STP12 | |
| K | STP13 | |
| L | STP14 | |
| M | STP2 | TTL Trigger Line |
| N | STP4 | |
| P | STP3 | |
| Q | STP5 | TTL Trigger I/O |
| /0 0 | STP7 | * Slot 0 MODID Line Resistor |

| Label | Reference Designator | Function |
|---|---|---|
| N0 0 | STP1 | V160-xyz3 MODID Line |
| CI C0 | STP8 | CLK10 I/O |
| D E | STP15 | * SYSRESET (System Reset Enable) |

| V160-xyz3 Strap Descriptions | |
|---|---|
| Arbiter Enable | Enables V160-xyz3 to function as the bus arbiter deciding which requester is granted control of the data bus. |
| Interrupt Handler | Enables V160-xyz3 to be an Interrupt Handler. |
| Slot 0 | Enables the V160-xyz3 to be a Slot 0 device. |
| Bus Request Level | Specifies the bus request level the V160-xyz3 will use when requesting the data bus. |
| CLK10 +/- | Configures routing for CLK10 +/- to P2 connector. |
| TTL Trigger Line | Specifies the TTL trigger line to be used in conjunction with the front panel Trigger I/O. |
| TTL Trigger I/O | Determines the direction of the front panel Trigger I/O. |
| Slot 0 MODID | Enables pull-up resistor connected to V160-xyz3 MODID line when used as a Slot 0 device. |
| V160-xyz3 MODID Line | Used for testing purposes only. Should remain in the factory-configured position to connect the MODID line to the MODID resistor. |
| CLK 10 I/O | Configures front panel CLK10 signal as an input or an output. |
| SYSRESET | Allows an onboard reset signal to be routed to *SYSRESET signal on backplane. |

## VXIBUS SLOT 0 CONFIGURATION

The V160-xyz3 is shipped from the factory configured for Slot 0 operation. If is to be installed in any other slot, is must be properly configured before installation.

**Warning:** If the V160-xyz3 is configured for Slot 0 operation, installation in any other slot may result in damage to the V160-xyz3, the Slot 0 device, and the VXIbus backplane.

A VXIbus Slot 0 device functions as the VMEbus System Controller. The System Controller is the VMEbus Arbiter for the data transfer bus. The controller also provides both the 16-MHz system clock (SYSCLK) and system reset (SYSRESET). The Slot 0 Device must also implement a MODID register to monitor and control the VXIbus MODID lines.

## V160-XYZ3 SLOT 0 CONFIGURATION

To configure theV160-xyz3 for Slot 0 operation,

1. Strap A (STP28) in down position - V160-xyz3 is bus arbiter.
2. Strap C (STP24) in left position - V160-xyz3 is Slot 0 device.
3. Strap D E (STP15) in right position - V160-xyz3 drives *SYSRESET.
4. Strap /0 0 (STP7) in right position - Enable Slot 0 MODID pull-up resistor.

To configure the V160-xyz3 for Non-Slot 0 operation,

1. Strap A (STP28) in up position - V160-xyz3 is not the bus arbiter.
2. Strap C (STP24) in right position - V160-xyz3 is not a Slot 0 device.
3. Strap D E (STP15) in left position - V160-xyz3 does not drive *SYSRESET.
4. Strap /0 0 (STP7) in left position - Disable Slot 0 MODID pull-up resistor.

## BUS REQUEST LEVEL

Straps D and E (STP25 and STP26, respectively) are used to specify the VMEbus Request Level the V160-xyz3 uses. Table 8 shows how these straps determine the request level.

**Table 8. - Bus Request Level Selections**

| Strap Position | | Bus Request Level |
|---|---|---|
| Strap D (BR1) | Strap E (BR0) | |
| Left | Left | Request Level 3 |
| Left | Right | Request Level 2 |
| Right | Left | Request Level 1 |
| Right | Right | Request Level 0 |

## INTERRUPT HANDLER

The Interrupt Handler feature may be defeated by means of a strap on the V160-xyz3. The default configuration is to have the Interrupt Handler enabled - Strap B in the left position. If the Interrupt Handler needs to be disabled, Strap B must be placed in the right position.

## TTL TRIGGER I/0

The V160-xyz3 can be configured to route the front panel Trigger I/O to a selected TTL Trigger line or take a selected TTL Trigger Line and route it to the front panel Trigger I/O. Straps M, N, and P are used to select a particular TTL Trigger Line. Strap Q is used to specify the direction of the TTL Trigger Line. When Strap Q is in the left position, the selected TTL Trigger Line is routed out to the front panel Trigger I/O signal. Conversely, the Strap Q in the right position allows the front panel Trigger I/O to source a trigger on the selected line. Table 9 shows how combinations of Straps M, N, and P select the TTL Trigger Line to be used. The left position of these straps is interpreted as a logical 0 and the right as a 1.

**Table 9. - I/O TTL Trigger Line Selections**

| STRAP POSITIONS | | | TTL TRIGGER LINE |
|---|---|---|---|
| M (TTL2) | P (TTL1) | N (TTL0) | |
| Left | Left | Left | 7 |
| Left | Left | Right | 6 |
| Left | Right | Left | 5 |

| STRAP POSITIONS | | | TTL TRIGGER LINE |
|---|---|---|---|
| M (TTL2) | P (TTL1) | N (TTL0) | |
| Left | Right | Right | 4 |
| Right | Left | Left | 3 |
| Right | Left | Right | 2 |
| Right | Right | Left | 1 |
| Right | Right | Right | 0 |

## CLK10 SOURCE

As the Slot 0 device, the V160-xyz3 must supply the VXIbus defined differential ECL CLK10 source. This signal may originate from an onboard 10MHz source or from the front panel CLK10 I/O signal. Straps F through L are used to determine if the V160-xyz3 supplies CLK10 and the if the source is from the onboard source or front panel CLK10 I/O signal. Only two shorting straps are supplied for strap locations F through L. When one of these straps is placed in location F, G, or H, the other strap must be in position J, K, or L, respectively. Table 10 defines the only valid combinations that may be used to configure straps F through L. If the front panel CLK10 I/O signal is used as the source, the CI-CO strap must be in the CI (left) position. If the front panel signal is not used as the CLK10 source, the CI-CO strap may be placed in the CO (right) position. This configuration allows the VXIbus CLK10 signal to appear as an output on the V160-xyz3's front panel CLK10 I/O signal, regardless of which slot the V160-xyz3 is installed.

**Table 10. CLK10 Configurations**

| Straps Installed in: | Function: |
|---|---|
| F and J | V160-xyz3 is source for VXIbus CLK10 from front panel CLK10 I/O signal. |
| G and K | V160-xyz3 is source for VXIbus CLK10 from onboard 10Mhz source. |
| H and L | V160-xyz3 is not the source for VXIbus CLK10. |

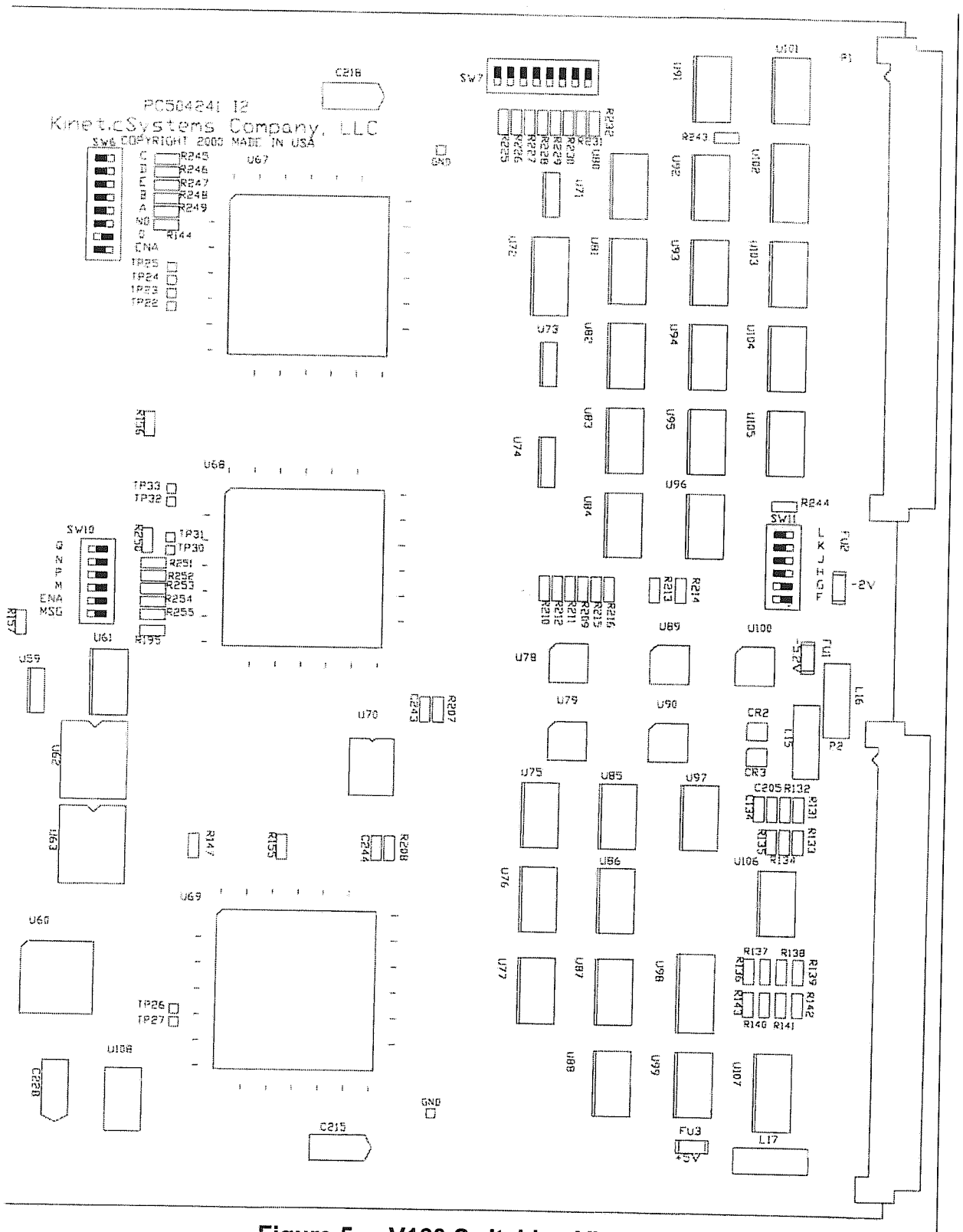



**Figure 5 - V160 Switching View**

The V160 is factory configured as shown.

1. Switch C Left
2. Switch D Left
3. Switch E Left
4. Switch A Left
5. Switch B Left
6. Switch 0 Right

1. V160 drives backplane CLK 10±.
2. TTL Trigger Line 0 is routed to front panel Trig I/O as an output.
3. V260 has pullup enabled for Slot 0 MODID line.

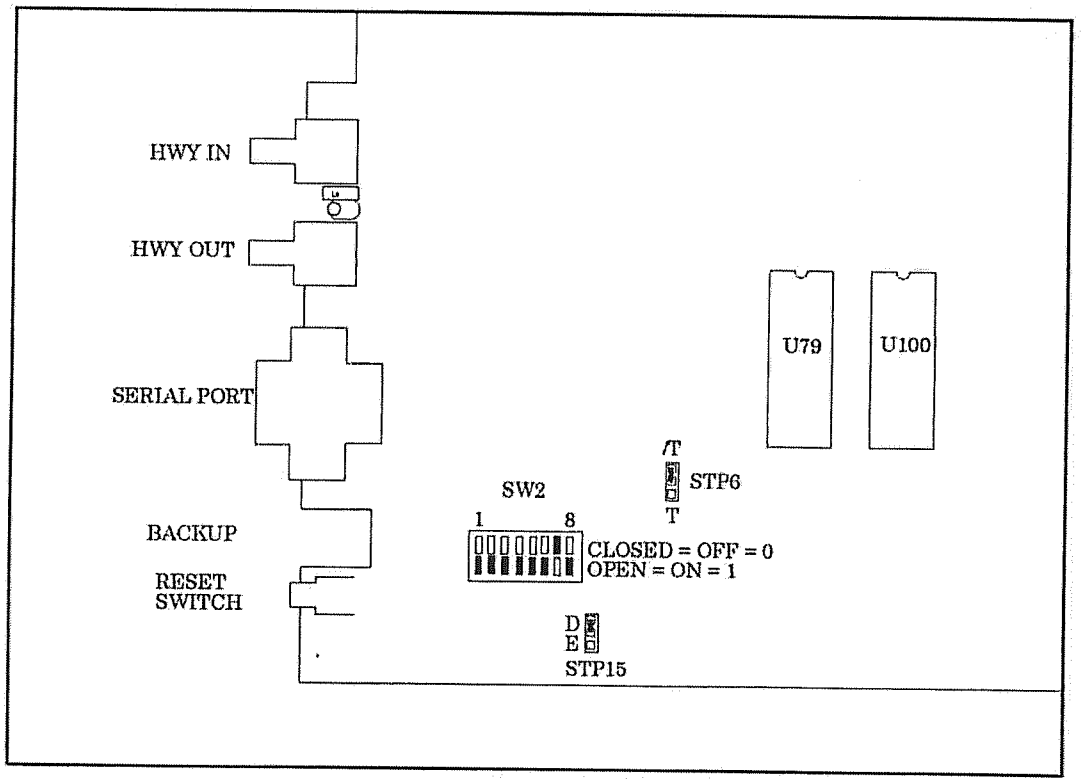



**Figure 6 Switch Setting 3**

Figure 6 shows the factory switch setting 3 for the SYSRESET and emulator switch. This configuration allows the V160 to drive the backplane SYSRESET signal.

*Model V160-xyz3*

Table 11 – Switch Configuration

| Label | Reference Designator | Function |
|---|---|---|
| A | SW6 POS 5 | *Arbiter Enable |
| B | SW6 POS 4 | *Interrupt Handler |
| C | SW6 POS 1 | *Slot 0 |
| D | SW6 POS 2 | Bus Request Level |
| E | SW6 POS 3 | |
| F | SW11 POS 1 | *CLK 10 ± |
| G | SW11 POS 2 | |
| H | SW11 POS 3 | |
| J | SW11 POS 4 | |
| K | SW11 POS 5 | |
| L | SW11 POS 6 | |
| M | SW10 POS 4 | TTL Trigger Line |
| N | SW10 POS 2 | |
| P | SW10 POS 3 | |
| Q | SW10 POS 1 | TTL Trigger I/O |
| /0 0 | SW6 POS 7 | *Slot 0 MODID Line Resistor |
| /0 0 | SW6 PO6 | V160 MODID Line<br>Factory Configuration – DO NOT MOVE |
| CI C0 | SW11 POS 5 | CLK 10 I/O |
| L | SW11 POS 6 | *SYSRESET (System Reset Enable) |
| ENA | SW10 POS 5 | Returns IRQ Level of Interrupter<br>Returns Logical Address of Interrupter |

## SWITCH DESCRIPTIONS

| | |
|---|---|
| Arbiter Enable | Enables V160 to function as the bus arbiter deciding which requester is granted control of the data bus. |
| Interrupt Handler | Enables V160 to be an Interrupt Handler. |
| Slot 0 | Enables the V160 to be a Slot 0 device. |
| Bus Request Level | Specifies the bus request level the V160 will use when requesting the data bus. |
| CLK 10± | Configures routing for CLK 10± to P2 connector. |
| TTL Trigger Line | Specifies the TTL trigger line to be used in conjunction with the front panel Trigger I/O. |
| TTL Trigger I/O | Determines the direction of the front panel Trigger I/O. |
| Slot 0 MODID | Enables pullup resistor connected to V160 MODID line when used as a Slot 0 device. |
| V160 MODID Line | Used for testing purposes only. Should remain in the factory configured position to connect the MODID line to the MODID resistor. |
| DSP Emulator | Allows use of emulator in place of DSP. This strap must remain in its factory configured position for proper operation. |
| CLK 10 I/O | Configures front panel CLK 10 signal as an input or an output. |
| SYSRESET | Allows an onboard reset signal to be routed to *SYSRESET signal on backplane. |

*Model V160-xyz3*
## V160 SLOT 0 CONFIGURATION

To configure the V160 for Slot 0 operation:

1. Switch 6 A — in left position – V160 is bus arbiter.
2. Switch 6 C — in left position – V160 is Slot 0 device.
3. Switch 11 L — in left position – V160 drives *SYSRESET.
4. Switch 6 0 — in right position – Enable Slot 0 MODID pullup resistor.

To configure the V160 for Non Slot 0 operation:

5. Switch 6 A — in right position – V160 is not the bus arbiter.
6. Switch 6 C — in right position – V160 is not a Slot 0 device.
7. Switch 11 L — in right position – V160 doe not drive *SYSRESET.
8. Switch 6 0 — in left position – Disable Slot 0 MODID pullup resistor.

## BUS REQUEST LEVEL

Switch 6 D and E are used to specify the VMEbus Request Level the V160 uses. The following table shows how these switches determine the request level.

**Table 12 - Bus Request Level Selection**

| Switch Position | | Bus Request Level |
|---|---|---|
| **Switch 6 D** | **Switch 6 E** | |
| Left | Left | Request Level 3 |
| Right | Left | Request Level 2 |
| Left | Right | Request Level 1 |
| Right | Right | Request Level 0 |

## INTERRUPT HANDLER

The Interrupt Handler feature may be defeated by means of a switch on the V160. The default configuration is to have the Interrupt Handler enabled – Switch 6 D in the left position. If the Interrupt Handler needs to be disabled, Switch 6 E must be placed in the right position.

## TTL TRIGGER I/O

The V160 can be configured to route the front panel Trigger I/O to a selected TTL Trigger line or take a selected TTL Trigger Line and route it to the front panel Trigger I/O. Switches D and E are used to select a particular TTL Trigger Line. Switch N0 0 is used to specify the direction of the TTL Trigger Line. When Switch 6 E is in the left position, the selected TTL Trigger Line is routed out to the front panel Trigger I/O signal. Conversely, the Switch 6 E in the right position allows the front panel Trigger I/O to source a trigger on the selected line. The following table shows how combinations of Switch 6 D and E select the TTL Trigger Line to be used. The left position of these switches is interpreted as a logical 0 and the right as a 1.

**Table 13 I/O TTL Trigger Line Selection**

| Switch 10 Position | | | TTL Trigger Line |
|---|---|---|---|
| M (TTL2) | N (TTL1) | P (TTL0) | |
| Left | Left | Left | 0 |
| Left | Left | Right | 1 |
| Left | Right | Left | 2 |
| Left | Right | Right | 3 |
| Right | Left | Left | 4 |
| Right | Left | Right | 5 |
| Right | Right | Left | 6 |
| Right | Right | Right | 7 |

## CLK 10 SOURCE

As the Slot 0 device, the V160 must supply the VXIbus defined differential ECL CLK 10 source. This signal may originate from an onboard 10MHz source or from the front panel CLK 10 I/O signal. Switch 11 F through K are used to determine if the V160 supplies CLK 10 and if the source is from the onboard source or front panel CLK 10 I/O signal. Only two shorting straps are supplied for strap locations F through L. When one of these straps is placed in location F, G, or H, the other switch must be in position J, IK, or L, respectively. The following table defines the only valid combinations that may be used to configure Switch 11 F through K. If the front panel CLK 10 I/O signal is used as the CI-CO switch must be in the CI (left) position. If the front panel signal is not used as the CLK 10 source, the CI-CO switch may be placed in the CO (right) position. This configuration allows the VXIbus CLK 10 signal to appear as an output on the V160's front panel CLK 10 I/O signal, regardless of which slot the V160 is installed.

**Table 14 CLK 10 Configurations**

| Switch 11 | Function |
|---|---|
| F and J | V160 is source for VXIbus CLK 10 from front panel CLK 10 I/O signal. |
| G and K | V160 is source for VXIbus CLK 10 from onboard 10 MHz source. |
| H and L | V160 is not the source for VXIbus CLK 10. |

# PROGRAMMING INFORMATION

## VMEBUS/VXIBUS ADDRESSING

Of the defined VXI*bus* Configuration Registers, The V160-xyz3 implements those required for extended register-based devices. The V160-xyz3 does not contain any registers located in A24 or A32 address space.

Access to the Configuration Registers for all VXI*bus* modules is available through the VMEbus short address space. The register addresses are located in the upper 16 kilobytes of the A16 address range ($C000_{16}$ to $FFFF_{16}$). The setting of the Logical Address switch, or the contents of the Logical Address Register (see below) are mapped into Address lines A6 through A13, thereby establishing a base address for the module somewhere in the range of$C000_{16}$ to $FFC0_{16}$.

## A16 BASE ADDRESS

The bit pattern for the A16 base address is shown below:

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A 15 | A 14 | A 13 | A 12 | A 11 | A 10 | A 09 | A 08 | A 07 | A 06 | A 05 | A 04 | A 03 | A 02 | A 01 | |
| 1 | 1 | LA 128 | LA 64 | LA 32 | LA 16 | LA 8 | LA 4 | LA 2 | LA 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 15 through 1 correspond to the VME address lines A15-A01.

Bits 15 and 14 are set to one (VXI defined).

Bits 13 through 6 are user selectable via the Logical Address switches and the ID/Logical Address register.

Bits 5 through zero are set to zero to indicate a block of 64 bytes.

## VXIBUS CONFIGURATION REGISTERS

Configuration Registers are required by the VXI*bus* specification so that the appropriate levels of system configuration can be accomplished. The Configuration Registers in the V160-xyz3 are offset from the base address. **Note: the V160-xyz3 only responds to these addresses if the Short Nonpriviledged Access ($29_{16}$) or Short Supervisory Access ($2D_{16}$) Address Modifier Codes are set for the VMEbus cycle.** Table 11 shows the applicable Configuration Registers present in the V160-xyz3, their offset from the base (Logical) address, and their Read/Write capabilities.

**Table 15. Configuration Registers Configuration (A16) Space**

| A16 Offset | Write/Read | Register Name |
|---|---|---|
| $00_{16}$ | Write/Read | ID/Logical Address Register |
| $02_{16}$ | Read Only | Device Type Register |
| $04_{16}$ | Write/Read | Status/Control Register |
| $06_{16}$ | N/A | Reserved |
| $08_{16}$ | Write/Read * | Attribute / MODID Register * |
| $0A_{16}$ | Read Only | Serial Number High Register |
| $0C_{16}$ | Read Only | Serial Number Low Register |
| $0E_{16}$ | Read Only | Version Number Register |
| $10_{16}$ - $19_{16}$ | Read Only | Reserved Registers |
| $1A_{16}$ | Read Only | Interrupt Status Register |
| $1C_{16}$ | Write/Read | Interrupt Control Register |
| $1E_{16}$ | Read Only | Subclass Register |
| $20_{16}$ | Read Only | Suffix Register High |
| $22_{16}$ | Read Only | Suffix Register Low |
| $24_{16}$ - $3F_{16}$ | N/A | Reserved |

* If the V160-xyz3 is configured as a Slot 0 Controller, the register at offset $08_{16}$ is the write/read MODID register. For non-Slot 0 configurations, the register at offset $08_{16}$ is the read-only Attribute register.

## ID/LOGICAL ADDRESS REGISTER (OFFSET $00_{16}$

The ID/Logical Address Register, which is located at offset $00_{16}$ from the logical base address, serves two functions, depending on the direction of the VME transfer. When executing a read operation to this register, the data returned indicates the Device Class, the Address Space requirements outside of A16 space, and the Manufacturer's Identification. A write operation to this register is only executed during a dynamic configuration sequence. During the configuration sequence, the Resource Manager assigns a logical address to the V160-xyz3 by writing the logical address into the lower 8-bits of this register. The format and bit assignments for this ID/Logical Address register are shown in the following diagram. Since this register has write-only and read-only bits, two bit patterns are shown.

On Read transactions:

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

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15, 14 | Device Class | This is an Extended Register Based device |
| 13,12 | Address Space | The V160-xyz3 does not require A24/A32 addressment space. |
| 11-0 | Manufacturer's ID | 3881 ($F29_{16}$) for KineticSystems. |

On Write transactions:

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | LA 128 | LA 64 | LA 32 | LA 16 | LA 08 | LA 04 | LA 02 | LA 01 |

For Write transfers to offset $00_{16}$ of the V160-xyz3, bits 15 through 08 are not used. A write to these bits has no effect on the V160-xyz3. In Dynamically Configured Systems (i.e., the Logical Address switches were set to a value of 255), bits 07 through 00 are written with the new Logical Address value. This write operation is typically executed by a Resource Manager.

## DEVICE TYPE REGISTER (OFFSET $02_{16}$)

The Device Type Register is a read-only register located at an offset of $02_{16}$ from the logical base address. This register contains the Model Code of the V160-xyz3. Since the V160-xyz3 is an A16-only device, the entire 16 bits of this register are used for the Model Code.

Model Codes for VXI Slot 0 devices must be in the range of $00_{16}$ to $FF_{16}$. Model Codes for VXI non-Slot 0 devices must be in the range of $100_{16}$ to $FFFF_{16}$. When the V160-xyz3 is configured for non-Slot 0

***Model V160-xyz3***
functionality, the Model Code returned when reading this register is $160_{16}$. To follow the VXI specification for Model Codes, the $100_{16}$ bit is set to zero when the V160-xyz3 is configured for Slot 0 operation.

V160-xyz3 Model Codes: $160_{16}$ for non-Slot 0 Operation
$060_{16}$ for Slot 0 Operation

The following diagram shows the bit pattern of the Device Type Register for both Slot 0 and non-Slot 0 configurations.

For Slot 0 configurations:

| 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 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

For non-Slot 0 configurations:

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

## STATUS/CONTROL REGISTER (OFFSET $04_{16}$

The Status/Control Register, which is located at an offset of $04_{16}$ from the logical base address, contains write-only, read-only, and write/read bits. The following describes the bits for write and read operations.

This bit pattern shows the register layout for read accesses to the Status/Control Register.

On Read transactions:

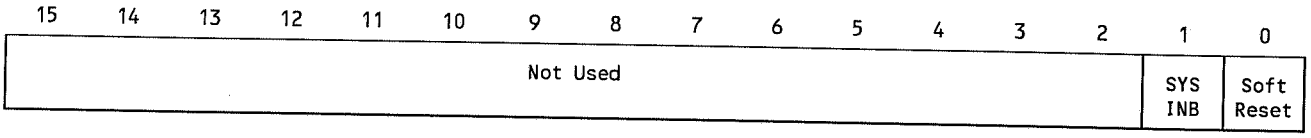
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | MODID* | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | Ready | Pass | SYSINB | Soft Reset |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15 | Not Used | This bit is not used and should be written as a zero. |
| 14 | MODID* | This bit is set to a one if the module is <u>not</u> selected with the MODID line on the VXI P2 connector. A zero indicates that the device is selected by a high state on the P2 MODID line. |

***Model V160-xyz3***

| | | |
|---|---|---|
| 13-4 | Not Used | These bits are not used and read as ones. |
| 3 | Ready | READY is a read-only bit that is set to a one indicating successful completion of register initialization. |
| 2 | Pass | Pass is a read-only bit that is set to a one after the V160-xyz3 has completed its power-on self-test without any errors. If errors occur, this bit is set to a zero and the SYSFAIL signal is asserted. |
| 1 | SYS INB | Reading this bit as a one indicates that the V160-xyz3 is may not drive the SYSFAIL* line. |
| 0 | Soft Reset | This write/read bit is used to reset the V160-xyz3. Reading this bit as a one indicates that the V160-xyz3 is currently in the RESET state. Setting this bit to a zero removes the V160-xyz3 from the RESET state. |

This bit pattern shows the register layout for write accesses to the Status/Control Register.




| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-2 | Not Used | These bits are not used for write operations. |
| 1 | SYS INB | SYSFAIL INHIBIT is a write-only bit that is set to a one to inhibit the V160-xyz3 from asserting the VXI SYSFAIL* signal. |
| 0 | Soft Reset | This write/read bit used to reset the V160-xyz3. Reading this bit as a one indicates that the V160-xyz3 is currently in the RESET state. Setting this bit to a zero removes the V160-xyz3 from the RESET state. |

## ATTRIBUTE REGISTER (OFFSET $08_{16}$

The Attribute Register is located at an offset of $08_{16}$ from the logical base address. This register is only available when the V160-xyz3 is configured as a non-Slot 0 device. When the V160-xyz3 is configured for Slot 0 operation, the Attribute Register is replaced with the MODID (Module Identification) Register. Refer to the MODID Register section of this manual for additional information.

The Attribute Register is a read-only register that contains information regarding the interrupt capabilities of the V160-xyz3.

The format of the Attribute Register is shown in the following diagram.

*Model V160-xyz3*

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | IR* | IH* | IS* |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-3 | Not Used | These bits are not used and read as ones. |
| 2 | IR* | The V160-xyz3 has interrupter control capabilities and returns this bit set to zero. |
| 1 | IH* | The V160-xyz3 has interrupt handling capabilities and returns this bit set to zero. |
| 0 | IS* | The V160-xyz3 has interrupt status capabilities and returns this bit set to zero. |

**MODID REGISTER (OFFSET $08_{16}$)**

The MODID Register is located at an offset of $08_{16}$ from the logical base address. This register is only available when the V160-xyz3 is configured as a Slot 0 device. When the V160-xyz3 is configured for non-Slot 0 operation, the MODID Register is replaced with the Attribute Register. Refer to the Attribute Register section of this manual for additional information.

The MODID Register is a write/read register used to control the MODID geographic addressing lines on the VXI P2 connector. Each of the 13 slots in the VXI chassis has an individual line that can be asserted and monitored through the MODID Register. Before any MODID lines can be asserted by the V160-xyz3, the Output Enable bit (bit 13) of the register must be set to a one. When the outputs are enabled, setting a MODID bit location to a one asserts the associated MODID signal.

The data read from the MODID bits in this register do not necessarily reflect the state of the bits that were written to this register. Instead, a read of this register returns the actual state of each MODID line.

The following diagram shows the bit pattern for the MODID Register.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | MODID Enable | MID 12 | MID 11 | MID 10 | MID 9 | MID 8 | MID 7 | MID 6 | MID 5 | MID 4 | MID 3 | MID 2 | MID 1 | MID 0 |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15,14 | Not Used | These bits are not used and read as ones. |

| | | |
|---|---|---|
| 13 | MODID Enable | OUTPUT ENABLE is a write/read bit used to enable or disable the V160-xyz3 from driving the MODID signals. Setting this bit to a one enables the drivers and a zero disables them. |
| 12-0 | MID12-0 | MODULE ID 12 through 0 are write/read bits used to assert and monitor the MODID signals. Writing a bit with a one will assert (as indicated by a high state) the corresponding slot's MODID line. |

## SERIAL NUMBER HIGH REGISTER (OFFSET $0A_{16}$

The Serial Number High Register, which is located at an offset of $0A_{16}$ from the logical base address, is used in combination with the Serial Number Low Register to define the serial number of the V160-xyz3 module. These registers are read-only and a write operation to these registers has no effect.

The format of the Serial Number High Register is shown in the following diagram.

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

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-0 | SN31-16 | SERIAL NUMBER 31 through 16 are read-only bits which represent the upper 16-bits of the 32-bit module serial number. These bits are used in conjunction with the bits SN15-SN00 in the Serial Number Low register. |

## SERIAL NUMBER LOW REGISTER (OFFSET $0C_{16}$

The Serial Number Low Register, which is located at an offset of $0C_{16}$ from the logical base address, is used in combination with the Serial Number High Register to define the serial number of the V160-xyz3 module. These registers are read-only and a write operation to these registers has no effect.

The format of the Serial Number Low Register is shown in the following diagram.

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

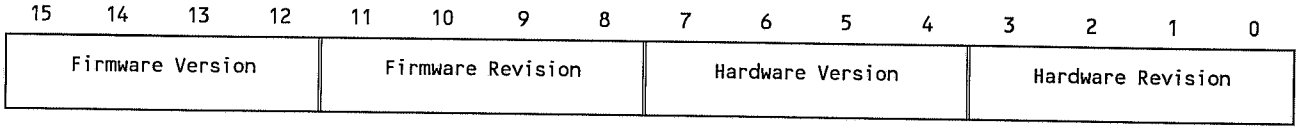
| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-0 | SN15-0 | SERIAL NUMBER15 through 0 are read-only bits which represent the lower 16-bits of the 32-bit module serial number. These bits are used |

in conjunction with the bits SN31-SN16 in the Serial Number High register.

## VERSION NUMBER REGISTER (OFFSET $0E_{16}$

The Version Number Register, which is located at an offset of $0E_{16}$ from the logical base address, is a read-only register that reflects the current revision level of the hardware and firmware residing on the V160-xyz3. All write operations to this register are ignored.

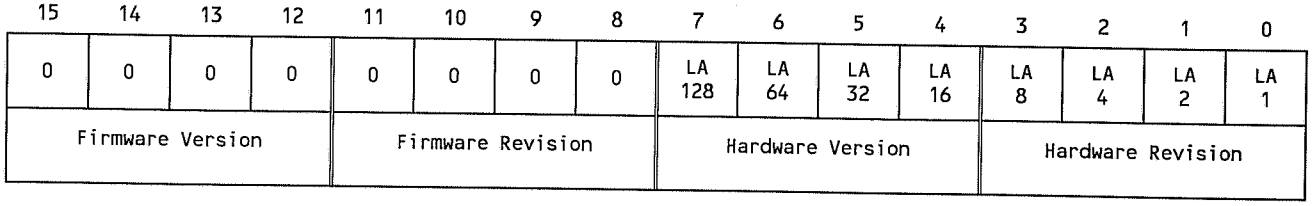
The following shows the bit layout of the Version Number Register.




| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-12 | Firmware Version | These bits reflect the module's firmware main revision level. |
| 11-8 | Firmware Revision | These bits reflect the module's firmware revision number. |
| 7-4 | Hardware Version | These bits reflect the module's firmware main revision number. |
| 3-0 | Hardware Revision | These bits reflect the module's firmware revision number. |

## INTERRUPT STATUS REGISTER (OFFSET $1A_{16}$

The Interrupt Status Register is a 16-bit read-only register located at an offset of $1A_{16}$ from the logical base address. This register is enabled onto the VMEbus during interrupt acknowledge cycles. The register contains the logical address of the V160-xyz3 in the lower 8-bits of the register and the upper 8-bits indicates the cause/status of the interrupt. Since the V160-xyz3 only has one interrupt source, the cause/status bits are always read as zero.
The format of the Interrupt Status Register is as follows:




| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-8 | Cause/Status | These bits are always returned as zeros. |
| 7-0 | LA128-LA1 | These bits return the current logical address of the V160-xyz3. |

## INTERRUPT CONTROL REGISTER (OFFSET $1C_{16}$

The Interrupt Control Register, which is located at offset $1C_{16}$ from the logical base address, is a write/read register used to configure the V160-xyz3 for interrupt generation. The Interrupt Request Level and Interrupt Enable bit are located in this register.

The format and description of the bits in the Interrupt Control Register are shown in the following diagram.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | IR EN* | 1 | IRQ S3 | IRQ S2 | IRQ S1 | 1 | 1 | 1 |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-8 | Not Used | These bits are not used and read as ones. |
| 7 | IREN* | INTERRUPT REQUEST ENABLE is a write/read bit used to enable/disable the generation of an interrupt by the V160-xyz3. Setting this bit to a zero enables the V160-xyz3 to interrupt and a one disables all interrupts. |
| 6 | Not Used | This bit is not used and read as a one. |
| 5-3 | IRQS3-IRQS1 | INTERRUPT REQUEST SELECT 3 through 1 are write/read bits used to select the desired Interrupt Request Level that the V160-xyz3 asserts when an interrupt is sourced. The following shows the Interrupt Request Level selections. |

| IRQS3 | IRQS2 | IRQS1 | Interrupt Request Level |
|---|---|---|---|
| 0 | 0 | 0 | IRQ7 |
| 0 | 0 | 1 | IRQ6 |
| 0 | 1 | 0 | IRQ5 |
| 0 | 1 | 1 | IRQ4 |
| 1 | 0 | 0 | IRQ3 |
| 1 | 0 | 1 | IRQ2 |
| 1 | 1 | 0 | IRQ1 |
| 1 | 1 | 1 | IRQ0 |

| | | |
|---|---|---|
| 2-0 | Not Used | These bits are not used and read as ones. |

## SUBCLASS REGISTER (OFFSET $1E_{16}$)

The Subclass Register, which is located at an offset of $1E_{16}$ from the logical base address, is a read-only register that indicates the subclass of the V160-xyz3. The V160-xyz3 is an Extended Register Based Device as the following pattern indicates.

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

| Bit(s) | Meaning |
|---|---|
| 15 | This bit is set to a one indicating that this is a VXIbus extended device. |
| 14-0 | These bits are set of $7FFE_{16}$ which indicates that the V160-xyz3 is an Extended Register Based Device. |

## SUFFIX HIGH REGISTER (OFFSET $20_{16}$)

The Suffix High register, which is located at an offset of $20_{16}$ from the logical base address, is a read-only register used in combination with the Suffix Low Register to determine the module model number suffix. The Suffix High Register contains the first two ASCII characters of the suffix and the Suffix Low Register contains the second two characters. The suffix shown is for the V160-xyz3-ZA11 module.

The format and bit assignments for the Suffix High Register are as follows:

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

This read only register contains the first two ACSII characters of the module's suffix ("ZA" = $5A41_{16}$).

## SUFFIX LOW REGISTER (OFFSET $22_{16}$)

The Suffix Low register, which is located at an offset of $22_{16}$ from the logical base address, is a read-only register used in combination with the Suffix High Register to determine the module model number suffix. The Suffix Low Register contains the last two ASCII characters of the suffix and the Suffix High Register contains the first two characters. The suffix shown is for the V160-xyz3-ZA11 module.

The format and bit assignments for the Suffix Low Register are as follows:

*Model V160-xyz3*

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

This read only register contains the last two ACSII characters of the module's suffix ("11" = $4141_{16}$).

## V160-XYZ3 INTERNAL REGISTER DESCRIPTION

All accesses to the V160-xyz3 internal registers must be 32-bit accesses, even if the register is less than 32-bits in width. Table 12 shows the address allocations of the various internal registers on the V160-xyz3.

**Table 16. Internal Registers**

| Address Location(Base 16) | Access Capabilities | Register Name |
|---|---|---|
| 00 | Write/Read | Control/Status Register (CSR) |
| 04 | Write Only | Burst Count Register (BCT) |
| 08 | Write Only | Delay Count Register (DCT) |
| 0C | Read Only | Total Transfer Count Register (TTCR) |
| 10 | Read Only | List Transfer Count Register (LTCR) |
| 14 | Read Only | Demand FIFO Register (DFR) |
| 18-1C | N/A | Reserved |
| 20 | Write Only | Trigger Source Register (TSR) |
| 24 | Write Only | Broadcast Trigger Mask (BTMSK) |
| 28-2C | N/A | Reserved |
| 30 | Write/Read | List Memory Address Register (LMA) |
| 34 | Write/Read | List Memory Data Register (LMD) |
| 38 | Write Only | Execute List (LGO) |
| 3C | Write Only | List Trigger Register (LTG) |
| 40 | Write/Read | List Write/Read Data Register (LWRD) |
| 44-4C | N/A | Reserved |
| 50 | Write Only | Timer Control Register (TCR) |
| 54 | Write Only | Timer Data Register (TDR) |

*Model V160-xyz3*

| Address Location(Base 16) | Access Capabilities | Register Name |
|---|---|---|
| 58-7C | N/A | Reserved |
| <80> | Write/Read | Multibuffer Memory Address (MBMA) |
| <84> | Read Only | Multibuffer Data Register (MBMD) |
| <88> | Write/Read | Multibuffer Control Register (MBMC) |
| <8C> | Write/Read | Buffer Interval Counter (BIC) |
| <90> | Write/Read | Buffer End Address (BEA) |

< > - Only applicable for models with buffer memory option.

## CONTROL/STATUS REGISTER (CSR) ($00_{16}$)

The Control/Status Register (CSR) is located at address $00_{16}$. This write/read register is used to control and monitor various operations within the V160-xyz3.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STS PAS | TMR ENA | LST BSY | DMD OFL | DMD CLR | DMD PND | DMD ENA | MBM D | INT DEN | MBM ENA | DLY ENA | BST ENA | INH | SYS FAL | RD SFL | SYS RST |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15 | STS PAS | The Self-Test Passed is a read-only bit is set to a one when the self-test executed by the V160-xyz3 passes, and set to a zero when it fails. |
| 14 | TMR ENA | Timer Enable is a write/read bit used to enable/disable the timer function on the V160-xyz3. The timer can be used to execute a list or source a trigger when the timer expires. |
| 13 | LST BSY | List Busy is a read-only bit that is set to a one as long as the V160-xyz3 is executing a list operation. |
| 12 | DMD OFL | Demand Overflow is a read-only bit that is set to a one when the Demand FIFO is full and a subsequent demand is received. This bit is cleared on power-up and by a write to the CSR with the DMD CLR (Demand Clear) bit set to a one. |
| 11 | DMD CLR | Demand Clear is a write-only bit which is used to clear the contents of the Demand FIFO and to clear the Demand Overflow (DMD OFL) bit once it is set. |
| 10 | DMD PND | Demand Pending is a read-only bit that is set to a one as long as a demand is stored in the Demand FIFO. This bit is cleared as long as no demands are present, after power-up, and after a write to the CSR with the DMD CLR bit set to a one. |
| 9 | DMD ENA | Demand Enable is a write/read bit which is used to enable/disable the generation |

| | | |
|---|---|---|
| | | of highway demand messages when the Demand FIFO contains data. |
| 8 | MBM DEN | Multibuffer Memory Demand Enable is a write/read bit used to enable or disable the storage of a demand entry in the Demand FIFO when the multibuffer memory reaches a programmable pointer. Setting this bit to a one enables the demand and a zero disables it. |
| 7 | INT DEN | Interrupt Demand Enable is a write/read bit used to enable or disable the storage of a demand entry in the Demand FIFO when a VXI interrupt occurs. Setting this bit to a one enables the demand and a zero disables it. |
| 6 | MBM ENA | Multibuffer Memory Enable is a write/read bit used to enable the multibuffer memory option for the V160-xyz3. Setting this bit to a one enables the memory and a zero disables it. |
| 5 | DLY ENA | Delay Enable is a write/read bit used to enable/disable the programmable delay associated with the receipt of broadcast trigger messages. Setting this bit to a zero disables the delay and a one enables it. |
| 4 | BST ENA | Burst Enable is a write/read bit used to enable/disable burst mode transfers on the VXI bus. Setting this bit to a one enables a DMA burst transfer of up to 256 transfers without relinquishing control of the VXI bus. Setting this bit to a zero disables this feature. |
| 3 | INH | Inhibit is a write/read bit used to source the front panel INHIBIT signal. |
| 2 | SYS FAL | SYSFAIL is a write/read bit used to control the VXI bus SYSFAIL signal. Setting this bit to a one asserts the signal, and unasserted when the bit is set to zero. |
| 1 | RD SFL | Read SYSFAIL is a read-only bit that reflects the current state of the VXI bus SYSFAIL signal. |
| 0 | SYS RST | System Reset is a write-only bit that is used to source the VXI bus SYSRESET signal. Setting this bit to a one asserts the SYSRESET signal. This bit is not latched. |

## BURST COUNT REGISTER (BCT) ($04_{16}$)

The Burst Count Register (BCT) is a write-only register located at address $04_{16}$. This register is used to specify the maximum number of transfers the V160-xyz3 may execute on the VXI bus without relinquishing control of the VXI bus. This specification allows for up to 256 transfers without re-negotiation. To enable burst transfers, the BST ENA bit in the Control/Status Register must be set to a one prior to DMA operations.

***Model V160-xyz3***

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | BCT 7 | BCT 6 | BCT 5 | BCT 4 | BCT 3 | BCT 2 | BCT 1 | BCT 0 |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-8 | Not Used | These bits are not used. |
| 7-0 | BCT7 -0 | Burst Count 7 through 0 specify the number of DMA transfers the V160-xyz3 may execute before relinquishing control of the VXI bus during a DMA operation. |

## DELAY COUNT REGISTER (DCT) ($08_{16}$)

The Delay Count Register is a write-only register located at address $08_{16}$ within the V160-xyz3. This register is used to specify a delay count associated with the receipt of a Broadcast Trigger command from the Interconnect Highway. Once the trigger command is received from the highway, the delay counter starts counting down until it underflows. When the count is exhausted, the contents of the Broadcast Trigger Register is used to source the corresponding signals. The delay count is enabled with the DLY ENA bit in the Control/Status Register. If the delay is not enabled in the CSR, the contents of the Broadcast Trigger Register are used immediately to source the corresponding signals.

The delay count specification is the number of 200 nanoseconds intervals to wait before sourcing the broadcast trigger data. This allows for a delay specification in the range of 200 nanoseconds to 13.1 milliseconds in 200 nanosecond increments.

There is a minimum delay time within the V160-xyz3 before the broadcast trigger data is sourced. This time is XX nanoseconds. Therefore, to calculate the time from receipt of the Broadcast Trigger message until the actual broadcast, data source is as follows:

Time = (xx nanoseconds) + (Delay Count X 200 nanoseconds)

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

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

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-0 | DCT15-0 | Delay Count 15 through 0 specify the amount of time to wait after receiving a Broadcast Trigger message until the Broadcast Trigger Mask Register data is sourced. This specification is the number of 200 nanosecond increments. |

## TOTAL TRANSFER COUNT REGISTER (TTCR) ($0C_{16}$)

The Total Transfer Count Register is a read-only register located at address $0C_{16}$ within the V160-xyz3.

***Model V160-xyz3***

This register is used to verify the number of valid transfers that have occurred during a requested operation. If an error occurs during a requested operation, this register may then be read to determine the actual number of valid transfers before the error occurred. This value is the two's complement of number of words remaining to be transferred. This may then be subtracted from the original value to determine the valid transfer count.

The TTCR is originally loaded with the transfer count value that is received from the command message. As DMA transfers occur, this counter is incremented and DMA operations continue until the TTCR is exhausted.

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

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TCR 15 | TCR 14 | TCR 13 | TCR 12 | TCR 11 | TCR 10 | TCR 9 | TCR 8 | TCR 7 | TCR 6 | TCR5 | TCR 4 | TCR 3 | TCR 2 | TCR 1 | TCR 0 |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 31-0 | TCR31-0 | Total Transfer Count 31 through 0 specify the total number of transfers remaining to be performed by the V160-xyz3. This data is in two's complement format. |

## LIST TRANSFER COUNT REGISTER (LTCR) ($10_{16}$)

The List Transfer Count Register is a read-only register located at address $10_{16}$ within the V160-xyz3. This register is used to verify the number of valid transfers that have occurred during a list block transfer operation. If an error occurs during a list block transfer operation, this register may be read to determine the actual number of valid transfers before the error occurred. This value is the two's complement of number of words remaining to be transferred. This may then be subtracted from the original value to determine the valid transfer count.

The LTCR is originally loaded with the transfer count value that is contained in the transfer count specification for the block instruction in the list. As DMA transfers occur, this counter is incremented and DMA operations continue until the LTCR is exhausted.

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

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LCR 15 | LCR 14 | LCR 13 | LCR 12 | LCR 11 | LCR 10 | LCR 9 | LCR 8 | LCR 7 | LCR 6 | LCR 5 | LCR 4 | LCR3 | LCR 2 | LCR 1 | LCR 0 |

*Model V160-xyz3*

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 31-0 | LCR31-0 | List Transfer Count 31 through 0 specify the number of transfers remaining to be performed by the V160-xyz3. This data is in two's complement format. |

**DEMAND FIFO REGISTER (DFR) ($14_{16}$)**

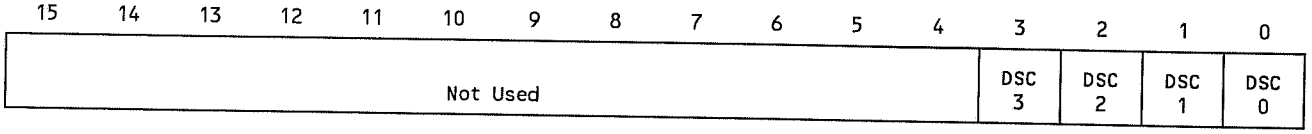
The Demand FIFO Register is a read-only register located at address $14_{16}$ in the V160-xyz3. This register is a FIFO capable of "holding" 2048 demand entries. These entries are generated by either VXI interrupts or by the Multibuffer Memory option. If the INT DEN bit in the Control/Status Register (CSR) is enabled, each interrupt event on the VXI bus is stored in the Demand FIFO. If the MBM DEN bit in the CSR is enabled, the Demand FIFO is written with a pattern indicating that the multibuffer memory has encountered a buffer segment crossing.

Demand entries are stored in the FIFO until they are either read by the host computer or sent out onto the Interconnect Highway as Demand messages. If the DMD ENA bit in the CSR is set to zero the V160-xyz3 does not generate any demand messages but continues to store demand events as they occur.

As long as there is at least one demand stored in the Demand FIFO, the DMD PND bit in the CSR is set to a one. If the Demand FIFO contains 2048 and one additional demand occurs, the data is not stored and the DMD OFL bit in the CSR is set. The Demand Overflow bits (DMD OFL) is not reset when the FIFO becomes less than full. A write operation to the CSR with the DMD CLR bit set to a one clears the FIFO and the overflow condition.

When a demand source is entered from an interrupt source, the data that is written into the FIFO is the actual interrupt request level. For example, data = 1 corresponds to interrupt request level 1; data = 6 corresponds to interrupt request level 6. Data from 1 to 7 represents demand sources generated from VXI interrupts. Data equal to 8 is received when the demand source is from the multibuffer memory card.

The following diagram shows the bit pattern for the Demand FIFO Register.




| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-4 | Not Used | These bits are not used. |
| 3-0 | DSC3-0 | Demand Source 3 through 0 reflect the source of the demand. |

***Model V160-xyz3***

The following chart shows the valid demand sources and the corresponding data values.

| DSC 3 | DSC 2 | DSC 1 | DSC 0 | Demand Source |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Reserved |
| 0 | 0 | 0 | 1 | Interrupt Request Level 1 |
| 0 | 0 | 1 | 0 | Interrupt Request Level 2 |
| 0 | 0 | 1 | 1 | Interrupt Request Level 3 |
| 0 | 1 | 0 | 0 | Interrupt Request Level 4 |
| 0 | 1 | 0 | 1 | Interrupt Request Level 5 |
| 0 | 1 | 1 | 0 | Interrupt Request Level 6 |
| 0 | 1 | 1 | 1 | Interrupt Request Level 7 |
| 1 | 0 | 0 | 0 | Multi Buffer Memory |

## TRIGGER SOURCE REGISTER (TSRC) ($20_{16}$)

The Trigger Source Register is a write-only register located at internal address $20_{16}$. This register is used to source any of the following signals:

1.) VXI TTL Trigger lines 0 through 7.
2.) VXI ECL Trigger lines 0 and 1.
3.) Front Panel Trigger Out B and A.
4.) Execute List.
5.) Reset Time Stamp.

A 200 nanosecond pulse is applied to each signal when its corresponding bit in the Trigger Source Register is written to a one. The following diagram shows the bit pattern for the Trigger Source Register.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | RST TSP | LST GO | FP TGB | FP TGA | ECL 1 | ECL 0 | TTL 7 | TTL 6 | TTL 5 | TTL 4 | TTL 3 | TTL 2 | TTL 1 | TTL 0 |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-14 | Not Used | These bits are not used. |
| 13 | RST TSP | Reset Time Stamp is set to clear the Time Stamp Counters. |

| | | |
|---|---|---|
| 12 | LST GO | List Go is set to a one to trigger a list processing operation. |
| 11 | FP TGB | Front Panel Trigger Out B is set to a one to source the front panel trigger out B. |
| 10 | FP TGA | Front Panel Trigger Out A is set to a one to source the front panel trigger out A. |
| 9-8 | ECL1-0 | The ECL1 and ECL0 bits are set to a one to source a trigger on the VXI trigger lines ECL1 and ECL0. |
| 7-0 | TTL7-0 | The TTL7 through TTL0 bits are set to a 1 to assert the VXI trigger lines TTL7 through TTL0. |

## BROADCAST TRIGGER MASK REGISTER (BTMSK) ($24_{16}$)

The Broadcast Trigger Mask Register is a write-only register that is located at internal address $24_{16}$ on the V160-xyz3. This register is used to specify the signal to be sourced when a Broadcast Trigger message is received by the V160-xyz3.

A Grand Interconnect host may issue a Broadcast Trigger message that informs all chassis connected to the highway to assert the data contained in the Broadcast Trigger Mask Register. On the V160-xyz3, this data contains VXI TTL/ECL triggers, the two front panel triggers, an execute list bit, and a reset time stamp bit.

Associated with the Broadcast Trigger mechanism is the Delay Timer. The Delay Timer, if enabled, causes the V160-xyz3 to delay the assertion of the Broadcast Trigger Mask Register until a predetermined interval of time elapses. The delay data is loaded in the Delay Count Register.

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | RST TSP | LST GO | FP TGB | FP TGA | ECL 1 | ECL 0 | TTL 7 | TTL 6 | TTL 5 | TTL 4 | TTL 3 | TTL 2 | TTL 1 | TTL 0 |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-14 | Not Used | These bits are not used. |
| 13 | RST TSP | Reset Time Stamp is set to clear the Time Stamp Counters. |
| 12 | LST GO | List Go is set to a one to trigger a list processing operation. |
| 11 | FP TGB | Front Panel Trigger Out B is set to a one to source the front panel trigger out B. |
| 10 | FP TGA | Front Panel Trigger Out A is set to a one to source the front panel trigger out A. |
| 9-8 | ECL1-0 | The ECL1 and ECL0 bits are set to a one to source a trigger on the VXI trigger lines ECL1 and ECL0. |

| | | |
|---|---|---|
| 7-0 | TTL7-0 | The TTL7 through TTL0 bits are set to a 1 to assert the VXI trigger lines TTL7 through TTL0. |

## LIST MEMORY ADDRESS REGISTER (LMA) ($30_{16}$)

The List Memory Address Register is a write/read register located at address $30_{16}$ on the V160-xyz3. This register is used to set the list memory address prior to writing to or reading from the 32K X 32 List Memory Data Register. Before writing or reading the list memory, a write operation to the LMA is executed to establish the initial address for storing/retrieving list memory data. After the address is loaded, the List Memory Data Register (LMD) may then be accessed. After a write or read operation to the LMD, the List Memory Address is automatically incremented.

The LMA is also used for specifying the initial address at which list processing starts.

The most significant bit in this register may be used to initiate a list operation when set to a one. The following diagram shows the bit pattern for the List Memory Address Register.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XEQ LST | LMA 14 | LMA 13 | LMA 12 | LMA 11 | LMA 10 | LMA 9 | LMA 8 | LMA 7 | LMA 6 | LMA 5 | LMA 4 | LMA 3 | LMA 2 | LMA 1 | LMA 0 |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15 | XEQ LST | Execute is a write-only bit that initiates a list processing operation. |
| 14-0 | LMA14-0 | List Memory Address 14 through 0 specify the list memory address location to access. |

## LIST MEMORY DATA REGISTER (LMD) ($34_{16}$)

The List Memory Data Register is a write/read register located at internal address $34_{16}$. This register allows access to the 32K X 32 list processing memory. This memory is loaded prior to list execution with the VXI commands (instructions) to be executed.

Before a write or read operation is executed to this register, the List Memory Address (LMA) register should be loaded with the initial address location to access. After a write or read operation to the LMD, the LMA is automatically incremented to the next address location. This allows for sequential access to the list memory without the need of reloading the LMA for each access.

The following diagram shows the bit pattern of the List Memory Data Register.

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

***Model V160-xyz3***

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

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 31-0 | LMD31-0 | List Memory Data 31 through 0 are write/read bits that represent the data for list execution. |

### LIST GO (LGO) ($38_{16}$)

The List Go is an internal address at location $38_{16}$. A write operation with any data pattern initiates list execution. This function is useful when it is necessary to execute a list operation that does not require any data transfers.

### LIST TRIGGER REGISTER (LTRG) ($3C_{16}$)

The List Trigger Register is a write-only internal register located at address $3C_{16}$. This register allows the selection of VXI triggers or the occurrence of an interrupt to initiate list execution. The following diagram shows the bit pattern of the List Trigger Register.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | LTG ENA | SEL 4 | SEL 3 | SEL 2 | SEL 1 | SEL 0 |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-6 | Not Used | These bits are not used. |
| 5 | LTG ENA | List Trigger Enable is used to enable/disable list execution on the occurrence of an interrupt or VXI trigger. Setting this bit to a one enables the function and a zero disables it. |
| 4-0 | SEL4-0 | Select 4 through 0 select the source of the list trigger. The following chart shows the valid combinations of these bits and the trigger source that they select. |

| SEL 4 | SEL 3 | SEL 2 | SEL 1 | SEL 0 | List Trigger Source |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | TTL Trigger 0 |
| 0 | 0 | 0 | 0 | 1 | TTL Trigger 1 |
| 0 | 0 | 0 | 1 | 0 | TTL Trigger 2 |
| 0 | 0 | 0 | 1 | 1 | TTL Trigger 3 |
| 0 | 0 | 1 | 0 | 0 | TTL Trigger 4 |
| 0 | 0 | 1 | 0 | 1 | TTL Trigger 5 |
| 0 | 0 | 1 | 1 | 0 | TTL Trigger 6 |
| 0 | 0 | 1 | 1 | 1 | TTL Trigger 7 |
| 0 | 1 | 0 | 0 | 0 | ECL Trigger 0 |
| 0 | 1 | 0 | 0 | 1 | ECL Trigger 1 |
| 0 | 1 | 0 | 1 | 0 | |
| | | through | | | |
| 0 | 1 | 1 | 1 | 1 | Reserved |
| 1 | 0 | 0 | 0 | 0 | Interrupt Request 1 |
| 1 | 0 | 0 | 0 | 1 | Interrupt Request 2 |
| 1 | 0 | 0 | 1 | 0 | Interrupt Request 3 |
| 1 | 0 | 0 | 1 | 1 | Interrupt Request 4 |
| 1 | 0 | 1 | 0 | 0 | Interrupt Request 5 |
| 1 | 0 | 1 | 0 | 1 | Interrupt Request 6 |
| 1 | 0 | 1 | 1 | 0 | Interrupt Request 7 |

## LIST WRITE/READ DATA REGISTER (LWRD) ($40_{16}$)

The List Write/Read Data Register is located at internal address $40_{16}$. This address is used to transfer data to/from the V160-xyz3 during list processing operations. When a write or read to this register is executed, the list processing sequence is initiated. If a write list was specified, the command to this register must be a write operation. Conversely, if a read list was specified, the command to this register must be a read.

*Model V160-xyz3*

## TIMER CONTROL REGISTER (TCR) ($50_{16}$)

The Timer Control Register is located at internal address $50_{16}$. This write-only register is used to specify what signals are to be asserted once the timer has expired. The timer is a 32-bit counter with 100 nanosecond resolution, which yields a range of 500 nanoseconds to 429.5 seconds. After the timer is loaded through the Timer Data Register and enabled in the Control/Status Register, the timer counts down until an underflow occurs. When this occurs, the contents of the Timer Control Register are asserted and the timer is reloaded.

Upon timer expiration, the V160-xyz3 can assert VXI triggers, front panel trigger, or initiate list execution. The following diagram shows the bit pattern for the Timer Control Register.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | RST TSP | LST GO | FP TGB | FP TGA | ECL 1 | ECL 0 | TTL 7 | TTL 6 | TTL 5 | TTL 4 | TTL 3 | TTL 2 | TTL 1 | TTL 0 |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-14 | Not Used | These bits are not used. |
| 13 | RST TSP | Reset Time Stamp is set to clear the Time Stamp Counters. |
| 12 | LST GO | List Go is set to a one to trigger a list processing operation. |
| 11 | FP TGB | Front Panel Trigger Out B is set to a one to source the front panel trigger out B. |
| 10 | FP TGA | Front Panel Trigger Out A is set to a one to source the front panel trigger out A. |
| 9-8 | ECL1-0 | The ECL1 and ECL0 bits are set to a one to source a trigger on the VXI trigger lines ECL1 and ECL0. |
| 7-0 | TTL7-0 | The TTL7 through TTL0 bits are set to a 1 to assert the VXI trigger lines TTL7 through TTL0. |

## TIMER DATA REGISTER (TDR) ($54_{16}$)

The Timer Data Register is located at internal address $54_{16}$. This write-only register is used to specify the timer interval used for sourcing the VXI triggers, front panel triggers, or initiating list operations. This register is composed of a 32-bit counter that is clocked by a 10 MHz clock. The allowable range for the timer is 500 nanoseconds to 429.5 seconds in 100 nanosecond increments.. After the timer is enabled in the Control/Status Register, the timer counts down until it underflows. When the timer expires, the data contained in the Timer Control Register is sourced, and then the timer is reloaded.

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

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

*Model V160-xyz3*

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

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 31-0 | TDR31-0 | Timer Data 31 through 0 are used to specify the timer interval used to source the signals contained in the Timer Control Register. |

The following registers are applicable only on models that include the Multi-Buffer Memory option.

## MULTI-BUFFER MEMORY ADDRESS REGISTER (MBMA) ($80_{16}$)

The Multi-Buffer Memory Address register is located at an internal offset of $80_{16}$. This write/read register is used to specify the address within the buffer to be read. To read a buffer location, this register must first be written with the address of the desired location. The contents of this buffer location may then be obtained be reading the Multi-Buffer Memory Data Register (MBMD). When the MBMD is read, the MBMA register is incremented allowing the next successive location to be read. This permits reads of successive locations without having to specify the address each time the data is read.

The MBMA contains twenty write/read bits to specify the desired location. The bits will select one of the possible 1,048,576 (1Meg) locations. Note: The 1Mbyte model will only contain 262,144 (256K) valid locations.

Refer to the Multibuffered Data Acquisition section of this manual for additional information.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | | | MBM A19 | MBM A18 | MBM A17 | MBM A16 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MBM A15 | MBM A14 | MBM A13 | MMB A12 | MBM A11 | MBM A10 | MBM A9 | MBM A8 | MBM A7 | MBM A6 | MBM A5 | MBM A4 | MBM A3 | MBM A2 | MBM A1 | MBM A0 |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 31-20 | Not Used | These bits are not used and are always read as zeros. |
| 19-00 | MBMA 19-0 | Multi-Buffer memory address bits. Specify the address for the desired location within the buffer. Bits 19 and 18 (MBM A19 and MBM A18) are used on the 4 Mbyte buffer model only. |

*Model V160-xyz3*

## MULTI-BUFFER MEMORY DATA REGISTER (MBMD) ($84_{16}$)

The Multi-Buffer Memory Data Register is a read only location at $84_{16}$. The buffer memory is used to store the read data the V160-xyz3 receives from DMA accesses. During DMA read operations, the V160-xyz3 may either store the read data in a FIFO or the buffer memory. The buffer memory allows storage of a large number of read operations (up to 1M) before the buffer fills and data must be read.

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MBM D15 | MBM D14 | MBM D13 | MBM D12 | MBM D11 | MBM D10 | MBM D9 | MBM D8 | MBM D7 | MBM D6 | MBM D5 | MBM D4 | MBM D3 | MBM D2 | MBM D1 | MBM D0 |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 31-0 | MBMD31-0 | Multi-Buffer memory data. |

## MULTI-BUFFER MEMORY CONTROL REGISTER (MBMC) ($88_{16}$)

This write/read register is located at an internal address of $88_{16}$. This register is used to monitor and clear the flags associated with the multi-buffer memory.

The flags in this register are set as the Buffer Interval Counter (BIC) expires. The BIC is decremented as each DMA operation is executed by the V160-xyz3. When the BIC is exhausted, the next sequential flag (beginning with FLG 1) is set. As the fourth flag is set, the next expiration of the BIC causes the first flag to be set. Once a flag has been set, the overrun flag will be set if the BIC expires on the set flag. The overrun flag is an indication that data will be lost when data is overwritten by subsequent transfers.

This register also contains a feature to clear flags automatically as the buffer is read. The CLR ENA bit is used to enable this feature. When one of the buffer flags (FLG 4-1) is set, the host can then read this section of the buffer, clearing the flag. This allows flags to be cleared without the need for a write operation. When flags are to be cleared in this manner, a read operation of buffer data should have a transfer count equal to the individual buffer size.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | CLR ENA | OVR RUN | FLG 4 | FLG 3 | FLG 2 | FLG 1 |

| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 15-6 | Not Used | These bits are not used and are read as zeros. |
| 5 | CLR ENA | Flag clear enable is used a read/write bit used to automatically clear a flag bit with each read of a MBM buffer . |

***Model V160-xyz3***

| | | |
|---|---|---|
| 4 | OVR RUN | MBM Overrun flag is a read/clear bit which is used to signify an overflow condition. This bit is set whenever the V160-xyz3 needs to set a multibuffer flag and the flag is already set. This indicates 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 | FLG4-0 | MBM flags 1-4 are read/clear bits set sequentially as the Buffer Interval Counter expires during multibuffer data acquisition. A write operation to a flag bit with data set to a one clears the flag. These flags may also be cleared by setting the Clear Enable bit (above) and reading the MBM data register. |

## BUFFER INTERVAL COUNTER (BIC) ($8C_{16}$)

The Buffer Interval Counter (BIC) is a write/read register located at internal address of $8C_{16}$. This counter is used to define the number of DMA transfers placed into each buffer. One to four buffers of equal size may be defined; the size of the buffer(s) dependent on the contents of this register and the Buffer End Address. The individual buffer size can be calculated using the following relationship:

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

Notice that the individual buffer size is the same as what should be placed in the BIC. For example, a total of 64 DMA transfers are to be shared between two buffers. The Buffer End Address should be one less than the total number of transfers (63 = $3F_{16}$). The individual buffer size becomes:

Individual Buffer Size = ( 64 ÷ 2 ) = $20_{16}$.

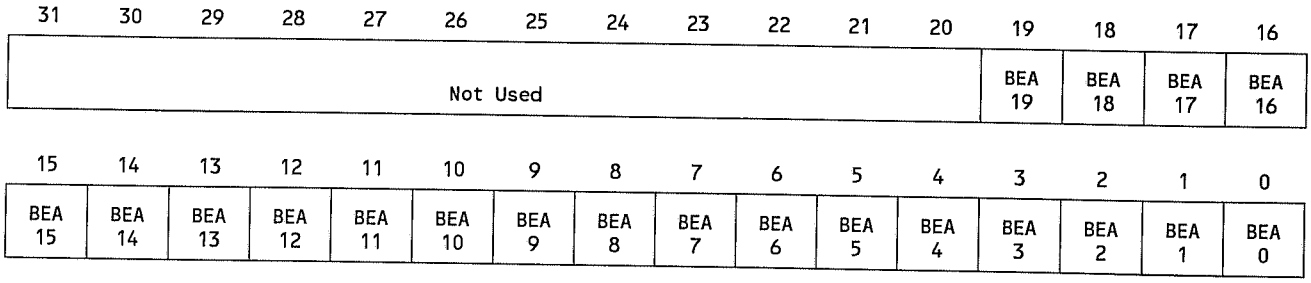
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | | | BIC 19 | BIC 18 | BIC 17 | BIC 16 |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 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 | Meaning |
|---|---|---|
| 31-20 | Not Used | These bits are not used and always read as zeros. |
| 19-0 | BIC 19-0 | Buffer interval count 19-0 are read/write bits that define the number of DMA read transfers the V160-xyz3 executes before a multibuffer flag is set. |

## BUFFER END ADDRESS (BEA) ($90_{16}$)

The Buffer End Address is a write/read register located at offset $90_{16}$. This register is used to define the address at which an internal counter is cleared. This internal counter is the pointer the V160-xyz3 uses to place read data into the multibuffer memory. Clearing this counter places the address pointer at the beginning of the first buffer.

*Model V160-xyz3*




| Bit(s) | Mnemonic | Meaning |
|---|---|---|
| 31-20 | Not Used | These bits are not used and are always read as zeros. |
| 19-0 | BEA 19-0 | Buffer end address 19-0 is write read bits which define the total buffer size the V160-xyz3 uses when storing DMA read data. As the internal address pointer to the MBM reaches the Buffer End Address, it is cleared, allowing it to point to the beginning of the first buffer. |

## SPECIAL HIGHWAY INSTRUCTIONS

Currently, two special instructions initiated by the Interconnect Driver are implemented on the V160-xyz3. Both of these commands are trigger instructions. For more information on the capabilities of these instructions, see the description of the Trigger Source and Broadcast Trigger Mask registers in the section on Internal Registers. The following chart shows the special instructions implemented by the V160-xyz3. The hex data shown corresponds to the first 16-bit word of the special instruction (header).

| Special Instruction (hex) | Function |
|---|---|
| 8040 | Addressed Slave Trigger |
| 8041 | Broadcast Trigger |

## 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 Internal Register Description section for further information on the Trigger Source Register.

*Model V160-xyz3*
The following is the format of the Addressed Slave Trigger instruction.

**First 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 specifies the node address to be accessed.

**Second 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. Refer to the Internal Register Description section for further information on the Broadcast Trigger Mask Register.

***Model V160-xyz3***
The following shows the format of the Broadcast Trigger instruction.

**First 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 |

**Second 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 |

## LIST PROCESSING INSTRUCTION FORMATS

The V160-xyz3 may perform a variety of operations as directed, through its list memory. This 32K X 32 bit memory used to hold the list processing instructions to be executed by the hardware list processor or the DSP. The list memory is typically loaded over the highway using the V160-xyz3's Internal Registers. The list of commands can contain instructions for data transfers or special instructions.

When list processing is initiated, the hardware list processor examines the header of the first list instruction. If the header indicates the instruction is to perform a data transfer within the VXI chassis, the hardware list processor reads the next successive list location(s) to obtain the necessary information for the data transfer. After the transaction is complete, the list processor increments the Command Memory Address and the next instruction is 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 is requested.

The format for list processing instructions is similar to what the driver uses for highway commands. Some parameters, such as node address, are not relevant for list processing instructions and are not used. The first 16-bit word the V160-xyz3 encounters in the list memory is referred to as the instruction header. The following diagram shows the format of the list processing instruction header.

| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CM 1 | CM 0 | Not Used | | | | | | | TM 2 | TM 1 | AM 2 | AM 1 | WS 2 | WS 1 | AD |

*Model V160-xyz3*

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

| CM1 | CM0 | Instruction Type |
|---|---|---|
| 0 | 0 | Reserved |
| 0 | 1 | VXI/VME Instruction |
| 1 | 0 | Special Instruction |
| 1 | 1 | Reserved |

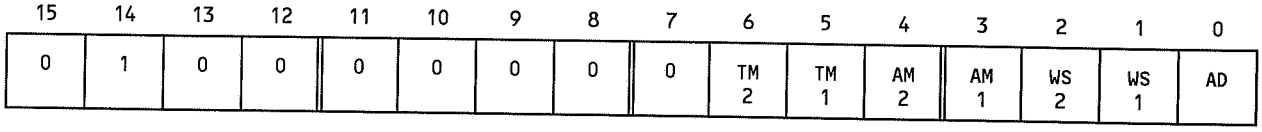
As shown above, VME/VXI and Special Instructions are the only valid types of list processing instructions. Other instruction types are reserved and should not be used.

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 dependent on the command type. For example, VXI/VME headers are followed by the 32-bit VME address.

## VXI/VME INSTRUCTIONS

For all VXI/VME instructions, the instruction header has bits 15 set to a zero and bit 14 set to a one. Bits 13 through 7 are not used. 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.




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

Bits 13 through 7 are set to zeros.

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

*Model V160-xyz3*

| 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. Data for write operations is provided through the List Write/Read Data Register. This register must be written with the write data for the operation. Data obtained from read operations will be placed in the Multi-Buffer Memory or the List Write/Read Data Register. If Multi-Buffer is not enabled, the read data must be read from the List Write/Read Data Register.

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. The data paths for Block Transfers are identical to those for Single Operations. Block Write operations use data provided through the List Write/Read Register. Block Read operations funnel data to the Multi-Buffer Memory or the List Write/Read Data Register. Typically, large blocks of data should be placed into the buffer memory. This allows the driver to take large blocks of data over the highway without the VME transfer overhead.

Single Inline Write Operations are single transfer VXI/VME write operations that have the write data embedded in the list. Since the write data is included in the list, this operation does not need to wait for the driver to supply the data. 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 | Reserved |

***Model V160-xyz3***

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 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.

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). 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | DIR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | AMD 5 | AMD 4 | AMD 3 | AMD 2 | AMD 1 | AMD 0 |

Bit 31 is normally used as the Internal Bit. List processing instructions may not access the V160-xyz3's Internal Registers. 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 V160-xyz3 to the VXI/VME chassis (write operations). This bit is set to a one for transferring data from the VXI/VME address to the V160-xyz3 (read operations). Bits 29 through 22 are not used and must be set to zeros.

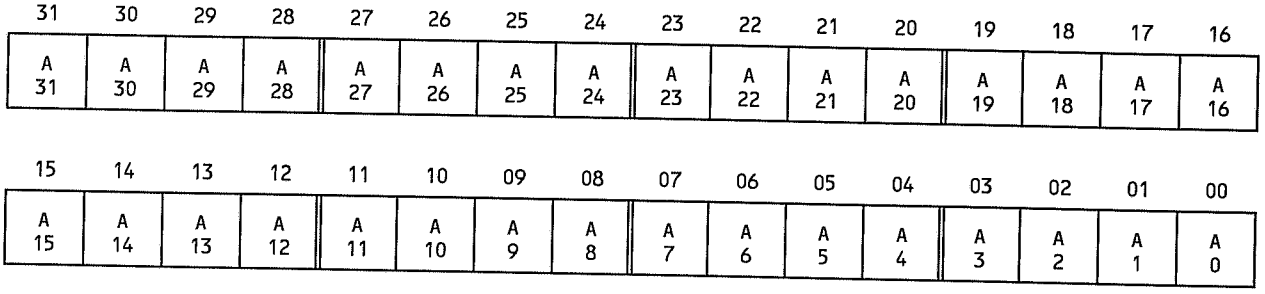
The ADDRESS MODIFIER bits, AMD5 through AMD0, specify the VME address modifier used for the VXI/VME bus cycle. The following chart is a subset of the VME address modifiers.

| 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 |

*Model V160-xyz3*

| Address Modifier Data (Hex) | Function |
|---|---|
| 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.




## VXI/VME SINGLE TRANSFER INSTRUCTIONS

The Header Word, Address Modifier, and Address specification are common for all VXI/VME instructions. For VXI/VME Single Operations, this is the only data required. For each Single Transfer instruction encountered, only one VME/VXI transfer will be executed by the V160-xyz3. 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.

*Model V160-xyz3*

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

**First Longword of Instruction:**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 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 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | AM 2 | AM 1 | WS 2 | WS 1 | AD |

**Second Longword of 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 |

For example, a single write operation is to be made to an address located within a VXI chassis and:

| | | |
|---|---|---|
| Direction | DIR | = 0 (A write operation is needed.) |
| VME/VXI Address | A31-0 | = $20000040_{16}$ (Address to be written.) |
| Address Modifier | AMD5-0 | = $09_{16}$ (A32 Nonpriviledged Data Access) |
| Word Size | WS2-1 | = 2 (16-bit transfer) |
| Access Mode | AM2-1 | = 0 (Not used for single reads/writes.) |
| Abort Disable | AD | = 0 (Abort on VXI timeout.) |

The format for this example instruction is shown below. The write data for this operation must be written to the List Write/Read Data Register.

**First Longword of Instruction:**

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

*Model V160-xyz3*

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | AM 2 | AM 1 | WS 2 | WS 1 | AD |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

**Second Longword of 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 |
| 0 | 0 | 0 | 1 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

For 16-bit transfers, bit for A0 is not used and should be specified as a zero. Similarly, 32-bit longword transfers do not use A0 or A1 and both should be zeros.

## 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 module.

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 List Memory, it loads the List Transfer Count Register with the Transfer Count data found in the list. The List Transfer Count Register is then incremented as each data word is transferred within 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.

*Model V160-xyz3*

The format of the VXI/VME Block Transfer instruction is shown in the following diagram.

**First Longword:**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 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 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | AM 2 | AM 1 | WS 2 | WS 1 | AD |

**Second 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 |

**Third Longword:**

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

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

An example of a Block Read operation is given below.

| | | |
|---|---|---|
| Direction | DIR | = 1 (A read operation is needed.) |
| VME/VXI Address | A31-0 | = $20000040_{16}$ (Start address of block to be read.) |
| Address Modifier | AMD5-0 | = $0F_{16}$ (A32 Supervisory Block Transfer) |
| Word Size | WS2-1 | = 2 (16-bit transfer) |
| Access Mode | AM2-1 | = 0 (Address is incremented.) |
| Abort Disable | AD | = 0 (Abort transfer on VXI timeout.) |
| Number of words to read | | = 2048 words |
| Transfer Count | | = Two's Complement of 2048 = $FFFFF800_{16}$ |

The format for this example instruction is shown below.

*Model V160-xyz3*

**First Longword of Instruction:**

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | AM 2 | AM 1 | WS 2 | WS 1 | AD |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

**Second Longword of 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 |
| 0 | 0 | 1 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

**Third Longword:**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TCR 31 | TCR 30 | TCR 29 | TCR 28 | TCR 27 | TCR 26 | TCR 25 | TCR 24 | TCR 23 | TCR 22 | TCR 21 | TCR 20 | TCR 19 | TCR 18 | TCR 17 | TCR 16 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

*Model V160-xyz3*

| 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 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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.

**First 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 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | AM 2 | AM 1 | WS 2 | WS 1 | AD |

**Second 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 |

*Model V160-xyz3*
**Third 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 |

For example, a single inline write operation is to be made to an address located within a VXI chassis and:

| | | |
|---|---|---|
| Direction | DIR | = 0 (A write operation is needed.) |
| VME/VXI Address | A31-0 | = $20000040_{16}$ (Address to be written.) |
| Address Modifier | AMD5-0 | = $09_{16}$ (A32 Nonpriviledged Data Access) |
| Word Size | WS2-1 | = 2 (16-bit transfer) |
| Access Mode | AM2-1 | = 0 (Not used for single reads/writes.) |
| Abort Disable | AD | = 0 (Abort on VXI timeout.) |
| Write Data | | = $55AA_{16}$ |

The format for this example instruction is shown below.

**First Longword of Instruction:**

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | AM 2 | AM 1 | WS 2 | WS 1 | AD |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

*Model V160-xyz3*

**Second Longword of 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 |
| 0 | 0 | 1 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| 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 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |

## SPECIAL INSTRUCTIONS

There are several special instructions implemented by the V160-xyz3. 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.

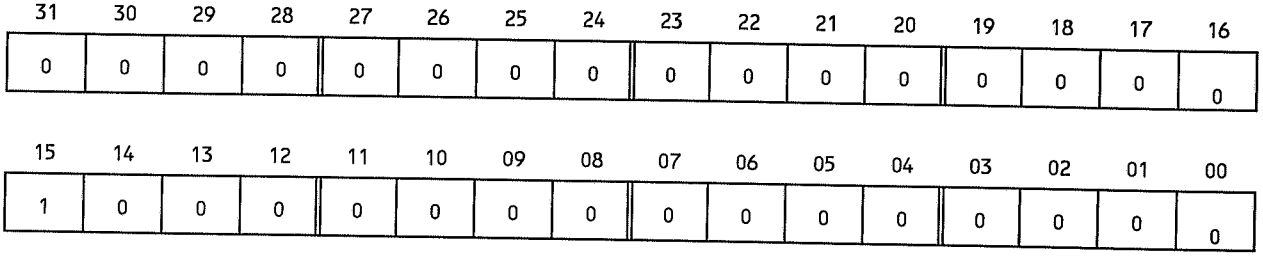
*Model V160-xyz3*

The following chart shows the special instructions implemented by the V160-xyz3. The hex data shown corresponds to the lower 16-bit word of the special instruction (header).

| Special Instruction (hex) | Function |
|---|---|
| 8000 | Halt |
| 8023 | Unconditional Branch |
| 8042 | Source Trigger |
| 8043 | Generate Interrupt |

## 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.




## BRANCH INSTRUCTIONS

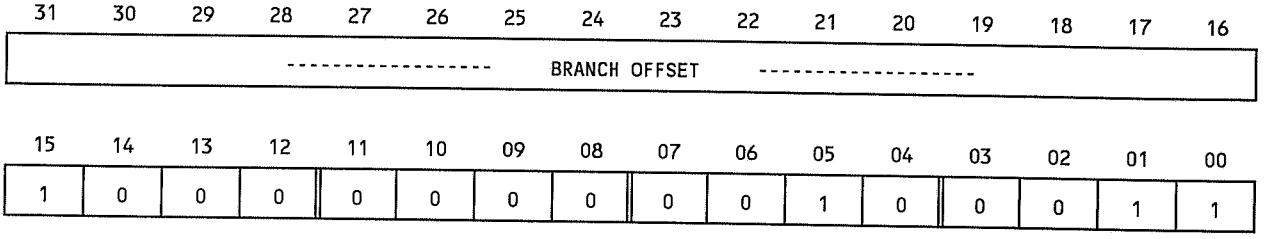
The branch instructions allow the V160-xyz3 list processor to make decisions branch to other locations within the list memory. Whenever a branch instruction is used, the user must be extremely careful that the data returned from a particular list processing operation can be traced to its source.

The branch offsets are 16-bit signed numbers and negative numbers are expressed as two's complement format.

*Model V160-xyz3*

## UNCONDITIONAL BRANCH INSTRUCTION

The Unconditional Branch instruction has an opcode of 8023 hex and is used to make unconditional branches. The following diagram shows the format for the Unconditional Branch instruction.

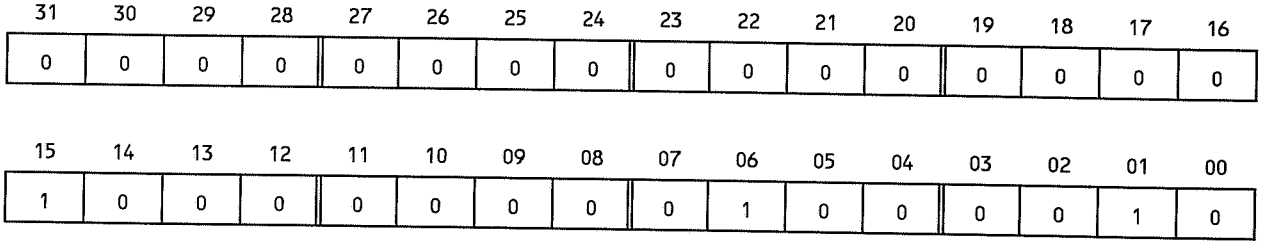



## SOURCE TRIGGER

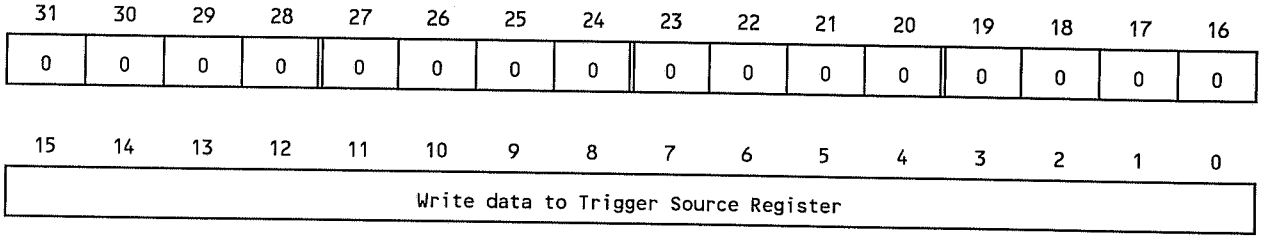
The Source Trigger Instruction has an opcode of 8042 hex and is used to generate a trigger by using the Trigger Source Register. For more information, see the description of the Trigger Source Register in the section on Internal Registers.

The following shows the format for the Source Trigger instruction.

### First Longword




### Second Longword

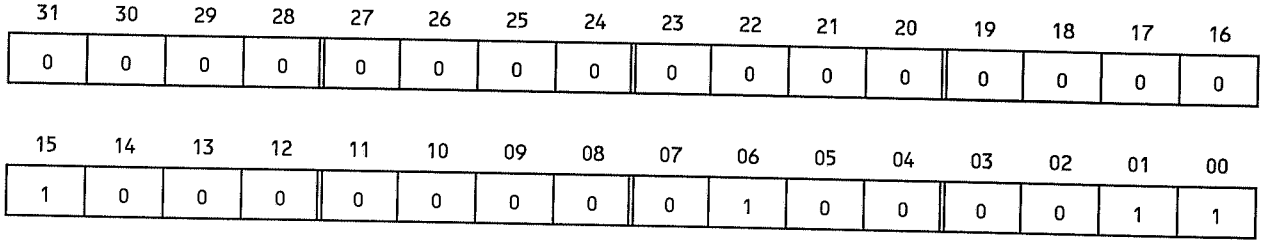


*Model V160-xyz3*
## GENERATE INTERRUPT INSTRUCTION

The Generate Interrupt Instruction has an opcode of 8043 hex and is used to generate an interrupt. In order for this instruction to generate an interrupt to the VXIbus, it must be first enabled in the Interrupt

## CONTROL/STATUS REGISTER.

The following shows the format for the Generate Interrupt instruction.




## LIST PROCESSING OPERATION

The V160-xyz3 provides a convenient method for executing a series of events through its list processing mechanism. This list of commands may contain various operations, including, VME transfers, branch instructions, or trigger instructions. Together, these operations can perform a complex sequence of events without the need for host intervention. Up to 127 slaves may be simultaneously executing unique lists as directed by a single host.

The list processing operation begins with the list being formed at the host and sent over the Interconnect Highway to the V160-xyz3. Several factors govern the content of the list.

1.) The only valid commands are those found in the section defining List Processing Instructions.
2.) The last instruction in valid list must be the HALT instruction.
3.) The list must be fewer that 32K (32768) elements long. (Including instructions, address information, in-line data, etc.)

After the list is loaded, a number of events can be used to start the list processing operation. A list of possible sources follows.

1.) Accessing the List Write/Read Data register.
2.) Writing the Execute List register.
3.) DSP initiates list execution. (Must be supported in firmware.)
4.) Front panel trigger signal - Execute List
5.) Broadcast Trigger Message
6.) Addressed Slave Trigger Message
7.) TTL Trigger line or Interrupt Request - Programmable selection in List Trigger Register
8.) Write the List Memory Address with bit 15 set to a one.
9.) Use onboard timer - Timer Control and Data Registers

As an example, several memory locations are to be manipulated using a V160-xyz3. It is assumed, all necessary setup has been done. A pseudo command sequence follows.

1.) Single-Inline Write to Address A - Data set to $AAAA_{16}$
2.) Single-Inline Write to Address B - Data set to $BBBB_{16}$
3.) Single Read of Address B
4.) Single Read of Address A
5.) Block Read of Address C - 1024 transfers
6.) Halt - Terminate List Execution

Table 13 shows a possible layout for this list of pseudo instructions.

**Table 17. List Processing Example**

| List Memory Address (Base 16) | <——32-bits——> List Memory Data (Base 16) | Instruction Description | Command Number |
|---|---|---|---|
| 0000 | 00094044 | Single-Inline Write Command | |
| 0001 | 80000000 | Address to Write (A) | |
| 0002 | 0000AAAA | Write Data | 1 |
| 0003 | 00094044 | Single-Inline Write Command | |
| 0004 | C0000000 | Address to Write (B) | |
| 0005 | 0000BBBB | Write Data | 2 |
| 0006 | 40094044 | Single Read Command | |
| 0007 | C0000000 | Read Address (B) | 3 |
| 0008 | 40094044 | Single Read Command | |
| 0009 | 80000000 | Read Address (A) | 4 |
| 000A | 400F4025 | Block Read Command | |
| 000B | D0000000 | Read Address (C) | |
| 000C | FFFFFBE8 | Transfer Count (2's Complement) | 5 |
| 000D | 00008000 | Halt Instruction | 6 |

## TIMER INITIATED LIST EXECUTION

The V160-xyz3 provides an operating mode whereby a list of commands can be repetitively executed at a predetermined rate. This rate is referred to as the Tic Rate. This rate can range from 1000 nanoseconds to 1000 seconds.

The Timer Data and Timer Control Registers are used to control the Tic Rate. The List Go (LST GO) bit

in the Timer Control Register must be set to a one to enable this function. The rate is determined by the contents of the Timer Data Register. The Timer Data Register must be written with a 32-bit value that specifies the Tic Rate.

The format for a timer initiated list is similar to what has been described before, with one exception. If a single list of instructions is to be executed at the Tic Rate, the list HALT instruction must be followed by an Unconditional Branch instruction. This branch instruction should direct the list processor to jump to the beginning of the list. As a HALT instruction is encountered in the list, list execution ceases and the List Memory Address is incremented to point to the next list instruction. When the onboard timer expires, triggering list execution, the list processor sees the branch instruction as the first command in the list. The list processor will then begin executing commands until it reaches the HALT instruction again.

As an example, several memory locations are to be manipulated using a V160-xyz3. It is assumed, all necessary setup has been done. A pseudo command sequence follows.

1.) Single-Inline Write to Address A - Data set to $AAAA_{16}$
2.) Single-Inline Write to Address B - Data set to $BBBB_{16}$
3.) Halt - Terminate List Execution
4.) Unconditional Branch to beginning of list.

Table 14 shows a possible layout for this list of pseudo instructions.

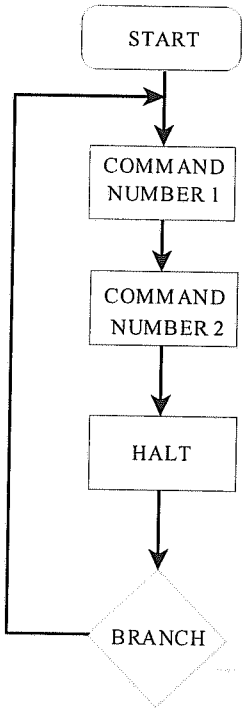
**Table 18. Timer Initiated List Processing Example**

| List Memory Address (Base 16) | <——32-bits——> List Memory Data (Base 16) | Instruction Description | Command Number |
|---|---|---|---|
| 0100 | 00094044 | Single-Inline Write Command | |
| 0101 | 80000000 | Address to Write (A) | |
| 0102 | 0000AAAA | Write Data | 1 |
| 0103 | 94044 | Single-Inline Write Command | |
| 0104 | C0000000 | Address to Write (B) | |
| 0105 | 0000BBBB | Write Data | 2 |
| 0106 | 00008000 | Halt Instruction | 3 |
| 0107 | FFF98023 | Unconditional Branch to start | 4 |

Notice that the branch instruction offset specifies a relative address. The branch offset (FFF9 = -7) specifies that the next command executed should be back seven positions in the list. The list memory address is read and then adjusted by this amount. Relative addressing has been employed so multiple re-locatable lists are possible.

*Model V160-xyz3*

The following flow diagram illustrates this sequence.




## MULTI-BUFFER MEMORY OPERATION

The model V160-xyz3-xB1x and V160-xyz3-xB2x contain one and four megabyte buffer memories, respectively. The purpose of the buffer memory is to store acquired data. Buffering allows a large amount of read data to be collected before the host must intervene and read the data.

Multi-buffering is a scheme where a number of buffers of equal size are defined. The V160-xyz3 allows 1-4 buffers ranging in size from 4 bytes to 1 or 4 Mbytes, depending on the option. The total size of the buffer can be calculated by the following expression.

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

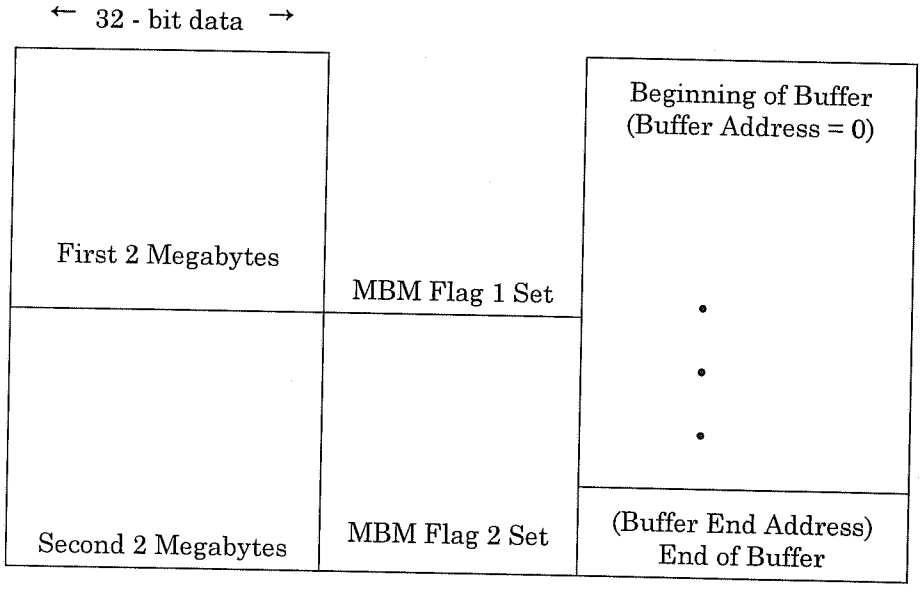
The Total Buffer Size is the value that should be written to the Buffer End Address (BEA) register. Similarly, the Individual Buffer Size dictates the value to be placed in the Buffer Interval Count (BIC) register.

The Buffer End Address specification is used to create a circular buffer. Initially, the received read data is stored in at the beginning of the buffer (buffer address 0). As additional read data words are received, incremental locations of the buffer are filled. When the buffer address reaches the Buffer End Address, it is reset to zero, moving this pointer to the beginning of the buffer.
The Buffer Interval Count is used to help prevent the circular buffer from inadvertently overwriting

acquired data. As read operations occur, the Buffer Interval Count is decremented. When the counter reaches zero, signifying the boundary of an individual buffer, it sets a multibuffer flag. These flags are monitored using the Multi-buffer Memory Control register. Once a flag is set, the Buffer Interval Counter reloads itself to prepare to count the data words placed into the next buffer. This process will continue until the Buffer End Address is reached. The circular buffer then rolls over to point to the beginning of the buffer. It is critical to have the Buffer End Address roll over at the same time the last multibuffer flag is set. If this is not the case, the Buffer Interval Counter will not correctly represent the number of words and set flags at inappropriate times. This can be prevented by using the relationship above to determine the proper Buffer End Address.

The MBM flags can be used by the host in two ways. First, the MBM flag bits may simply be polled by reading the MBM Control register. The second option is to enable the MBM flags to generate a Demand Message. Demand Messages may be inserted into the data stream by any slave on the Interconnect Highway. These messages are sent back to the host, providing an asynchronous means of signaling the host that a buffer has filled. For more information on this process refer to the section on Demand Messages and the manual provided with the appropriate highway driver. In either case, the individual flags can be cleared be writing to the bits with data set to ones or by using the auto-clear feature and reading the individual buffer.

The following diagram illustrates a typical example in which a 4 megabyte ( 1 megaword) buffer is used. The buffer is set up such that a multibuffer flag is set each time one-half of the buffer is filled.




For the above example, the Buffer End Address is set to FFFFF hex. This specification indicates that after the FFFFF hex location is accessed, the buffer rolls over to point to the starting location. Therefore, after the V160-xyz3 has received 1048576 32-bit read data words, the buffer address is reset to zero and the buffer filling continues. Since two buffers are being used, the Buffer Interval Count should be set for one-half of the buffer size. Using the relationship described before:

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

***Model V160-xyz3***

| | | | | |
|---|---|---|---|---|
| Total Buffer Size | = | 1048576 data words | = | Buffer End Address ( FFFFF hex) |
| Number of Buffers | = | 2 | | |
| Individual Buffer Size | = | 524288 | = | Buffer Interval Count (80000 hex) |

In this case, the buffer begins to fill, decrementing the interval counter, until 524288 32-bit words are in the buffer. At this point, the first two megabyte buffer is full, the first MBM flag is set, and the interval counter reloads itself. Setting this flag can prompt the host to read the first buffer by the polling or demand techniques described before. The next read transfer places the data in the first location of the second two megabytes. Again, successive locations are filled until the interval counter sets the second MBM flag and reloads itself. The buffer address will equal the end address (FFFFF hex) and the buffer address is cleared. Further read transfers begin overwriting the data in the first two megabyte buffer. The other two MBM flags are not used in this example because only two buffers are defined. If three buffers were defined, the first three flags would be used.

## DEMAND MESSAGES

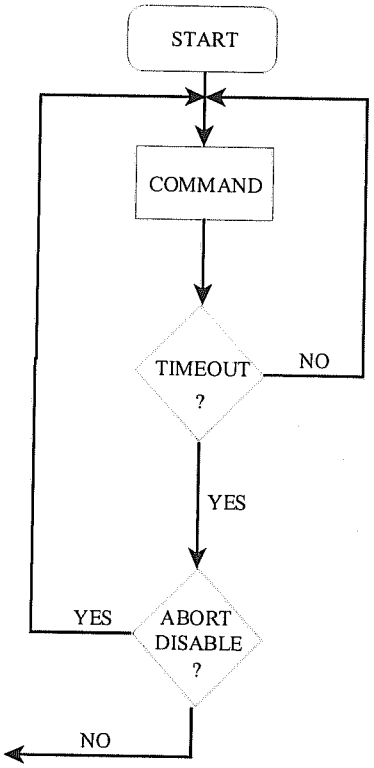
The purpose of Demand Messages is to have a mechanism to inform the host of certain conditions. In the case of the multibuffer memory example, a demand may be generated as each flag is set. After the first flag generates a demand, the host must read the contents of the first buffer or risk losing data when the buffer rolls over.

The V160-xyz3 can generate demand messages originating from three different sources. These sources include the DSP, multibuffer memory (flags) and a VME interrupt. Refer to the description of the Demand FIFO Register (DFR) and the Control Status Register (CSR) for further information.

Demands may be handled in two ways. One method of managing demands is by simply polling the Demand Pending (DMD PND) bit in the Control Status Register and reading the Demand FIFO. The second is to enable a pending demand to generate a highway message. When handled this way, it is not necessary to constantly read the Control Status register to determine if further action should be taken. The V160-xyz3 can also insert these demand messages in the data stream. This allows demands to be seen at the host during large block transfers.

## ABORT DISABLE FUNCTION

The Abort Disable function is used by the V160-xyz3 during list processing and highway transfers. Both cases are similar and its operation is shown by the following figure.




This diagram shows the sequence of the transfer. As each transfer ends, the V160-xyz3 determines whether or not the transfer was prematurely aborted by the onboard timer. If not, the V160-xyz3 continues on to process the next command. If the transfer did timeout, it then checks the Abort Disable bit. If this bit is set, the V160-xyz3 continues on as if a timeout had not occurred. In general this bit should be specified as inactive, allowing the V160-xyz3 to generate the appropriate error messages. For more information on the Abort Disable bit see the sections on Command and List Processing Instruction Formats.

## DSP FIRMWARE

The onboard DSP uses firmware for list processing and serial port operations. This firmware may be periodically updated to include additional functionality as the need arises. This section describes the serial port functions and the process of updating this firmware.

## SERIAL PORT FUNCTIONS

The V160-xyz3 has a serial port on the front panel that provides a communication path with the onboard DSP. The interface to the DSP is one-half of a Signetics SCN2681 dual asynchronous receiver/transmitter (DUART). The other channel of the DUART is not used and its use is reserved for future definition.

*Model V160-xyz3*

The DUART uses a MAXIM MAX232CPE RS-232 driver/receiver. The serial port pin description is shown in the chart below.

| Pin Number(s) | Description |
|---|---|
| 2 | Receive Data (RD) |
| 3 | Transmit Data (TD) |
| 5 | Ground (GND) |
| 1, 4, 6-9 | No Connection |

The baud rate for this channel is selected using SW2 positions five through eight.

# APPENDIX

## ID/Logical Address Register (Offset $00_{16}$)

On Read transactions:

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

On Write transactions:

| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | LA 128 | LA 64 | LA 32 | LA 16 | LA 08 | LA 04 | LA 02 | LA 01 |

## Device Type Register (Offset $02_{16}$)

For Slot 0 configurations:

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

For non-Slot 0 configurations:

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

## Status/Control Register (Offset $04_{16}$)

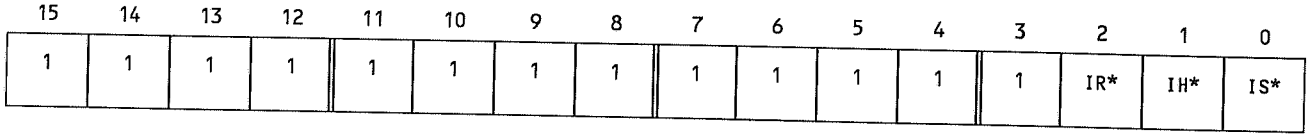
On Read transactions:

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | MODID * | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | Ready | Pass | SYS INB | Soft Reset |

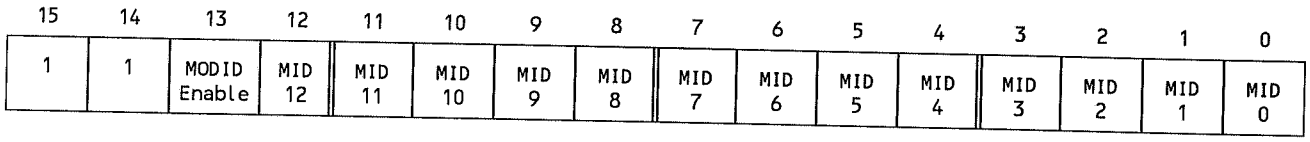
On Write transactions:

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | | | | | SYS INB | Soft Reset |

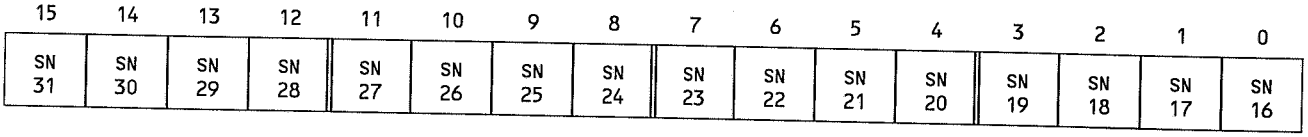
### Attribute Register (Offset $08_{16}$)



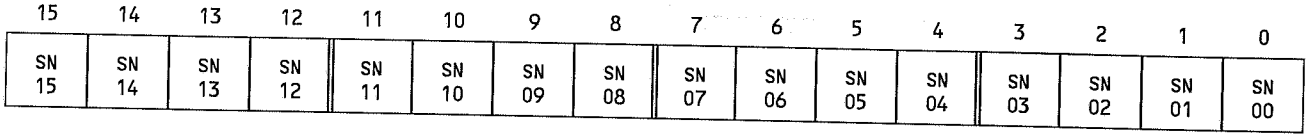

### MODID Register (Offset $08_{16}$)



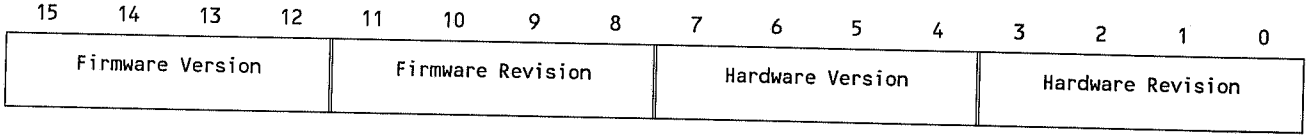

### Serial Number High Register (Offset $0A_{16}$)



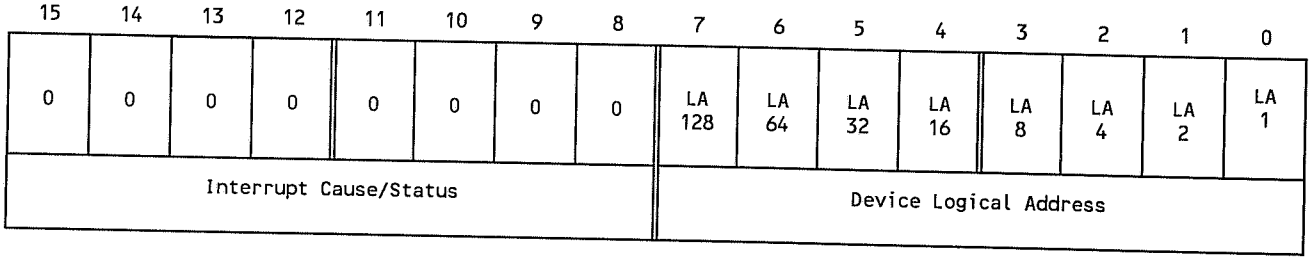

### Serial Number Low Register (Offset $0C_{16}$)



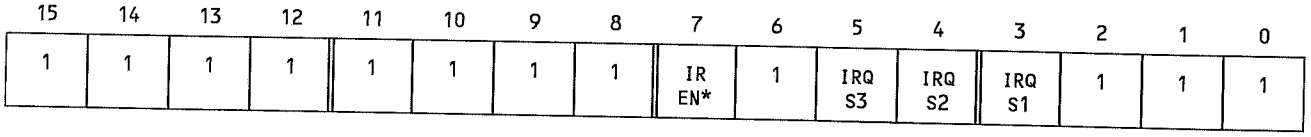

### Version Number Register (Offset $0E_{16}$)




### Interrupt Status Register (Offset $1A_{16}$)

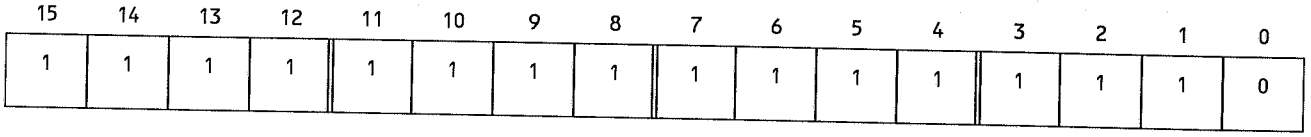


### Interrupt Control Register (Offset $1C_{16}$)



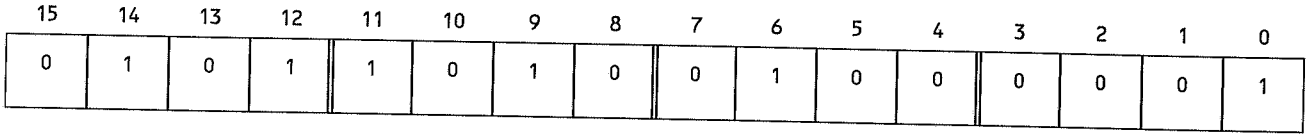
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | IR EN* | 1 | IRQ S3 | IRQ S2 | IRQ S1 | 1 | 1 | 1 |

### Subclass Register (Offset $1E_{16}$)



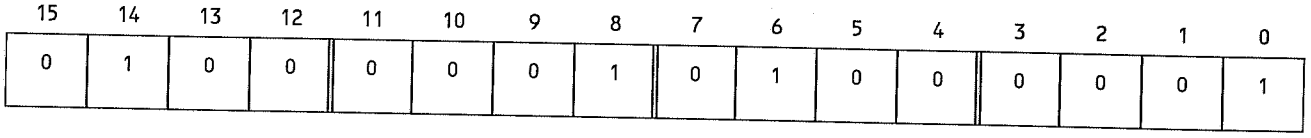
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

### Suffix High Register (Offset $20_{16}$)



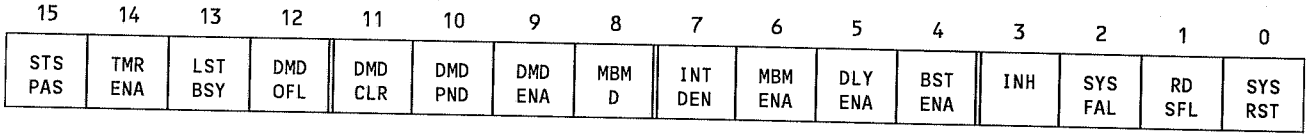
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

### Suffix Low Register (Offset $22_{16}$)



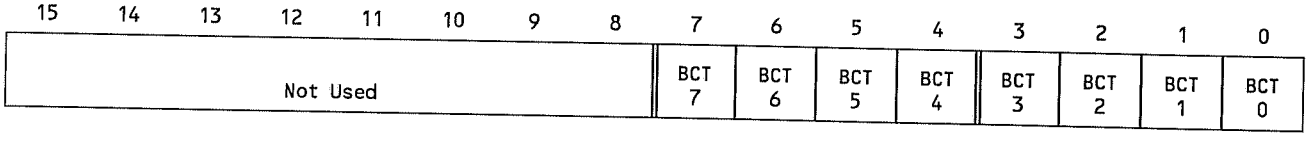
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

### Control/Status Register (CSR) ($00_{16}$)



| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STS PAS | TMR ENA | LST BSY | DMD OFL | DMD CLR | DMD PND | DMD ENA | MBM D | INT DEN | MBM ENA | DLY ENA | BST ENA | INH | SYS FAL | RD SFL | SYS RST |

### Burst Count Register (BCT) ($04_{16}$)



| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | BCT 7 | BCT 6 | BCT 5 | BCT 4 | BCT 3 | BCT 2 | BCT 1 | BCT 0 |

### Delay Count Register (DCT) ($08_{16}$)

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

### Total Transfer Count Register (TTCR) ($0C_{16}$)

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TCR 15 | TCR 14 | TCR 13 | TCR 12 | TCR 11 | TCR 10 | TCR 9 | TCR 8 | TCR 7 | TCR 6 | TCR5 | TCR 4 | TCR 3 | TCR 2 | TCR 1 | TCR 0 |

### List Transfer Count Register (LTCR) ($10_{16}$)

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LCR 15 | LCR 14 | LCR 13 | LCR 12 | LCR 11 | LCR 10 | LCR 9 | LCR 8 | LCR 7 | LCR 6 | LCR 5 | LCR 4 | LCR3 | LCR 2 | LCR 1 | LCR 0 |

### Demand FIFO Register (DFR) ($14_{16}$)

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | | | DSC 3 | DSC 2 | DSC 1 | DSC 0 |

### Trigger Source Register (TSRC) ($20_{16}$)

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | RST TSP | LST GO | FP TGB | FP TGA | ECL 1 | ECL 0 | TTL 7 | TTL 6 | TTL 5 | TTL 4 | TTL 3 | TTL 2 | TTL 1 | TTL 0 |

### Broadcast Trigger Mask Register (BTMSK) ($24_{16}$)

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | RST TSP | LST GO | FP TGB | FP TGA | ECL 1 | ECL 0 | TTL 7 | TTL 6 | TTL 5 | TTL 4 | TTL 3 | TTL 2 | TTL 1 | TTL 0 |

### List Memory Address Register (LMA) ($30_{16}$)

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XEQ LST | LMA 14 | LMA 13 | LMA 12 | LMA 11 | LMA 10 | LMA 9 | LMA 8 | LMA 7 | LMA 6 | LMA 5 | LMA 4 | LMA 3 | LMA 2 | LMA 1 | LMA 0 |

### List Memory Data Register (LMD) ($34_{16}$)

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

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

### List Go (LGO) ($38_{16}$)

The List Go is an internal address at location $38_{16}$. A write operation with any data pattern initiates list execution. This function is useful when it is necessary to execute a list operation that does not require any data transfers.

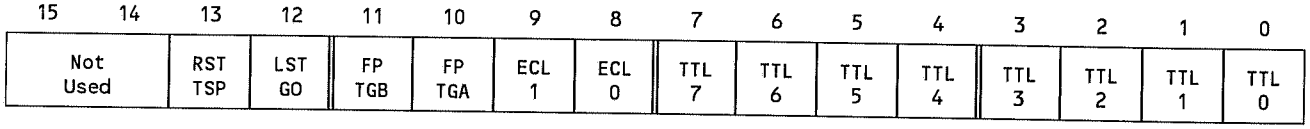
### List Trigger Register (LTRG) ($3C_{16}$)

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Not Used | | | | | | | | | | LTG ENA | SEL 4 | SEL 3 | SEL 2 | SEL 1 | SEL 0 |

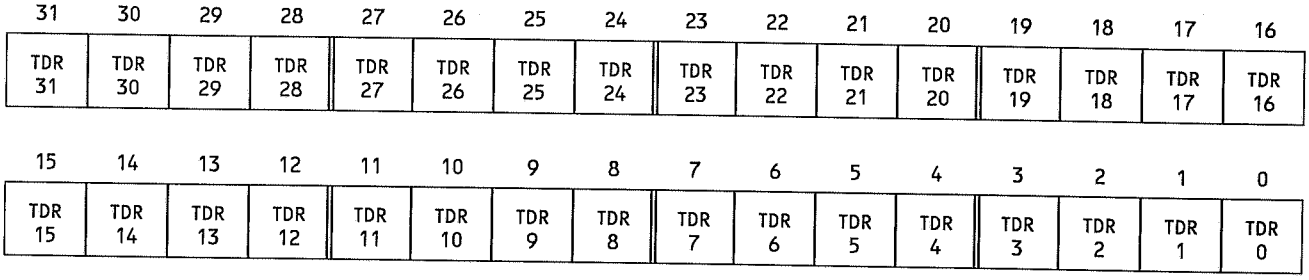
### List Write/Read Data Register (LWRD) ($40_{16}$)

The List Write/Read Data Register is located at internal address $40_{16}$. This address is used to transfer data to/from the V160-xyz3 during list processing operations. When a write or read to this register is executed, the list processing sequence is initiated. If a write list was specified, the command to this register must be a write operation. Conversely, if a read list was specified, the command to this register must be a read.

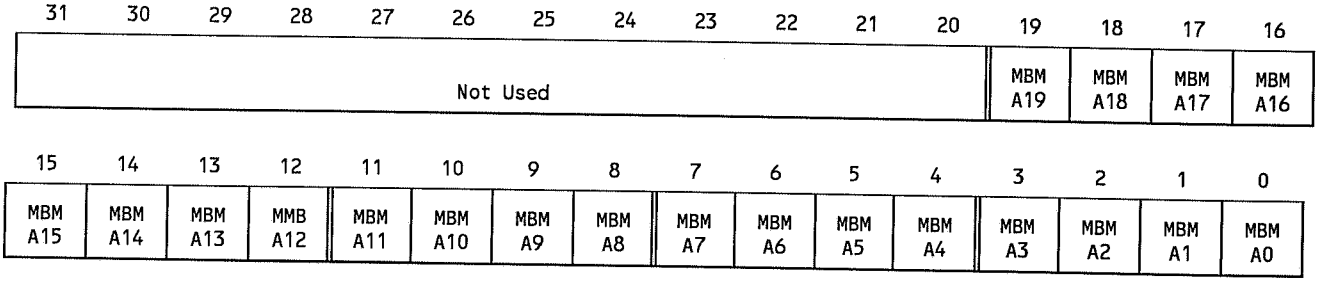
### Timer Control Register (TCR) ($50_{16}$)



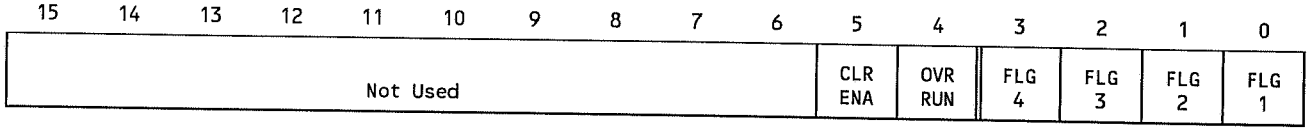

### Timer Data Register (TDR) ($54_{16}$)



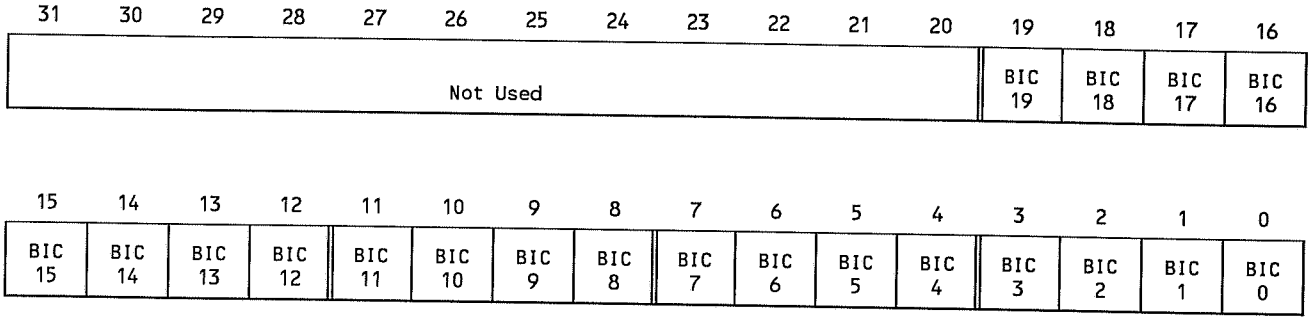

### Multi-Buffer Memory Address Register (MBMA) ($80_{16}$)



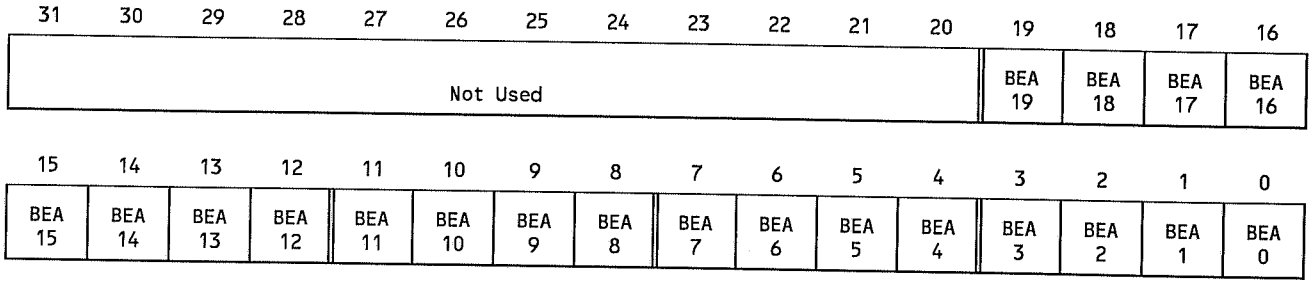

### Multi-Buffer Memory Control Register (MBMC) ($88_{16}$)



### Buffer Interval Counter (BIC) ($8C_{16}$)




### Buffer End Address (BEA) ($90_{16}$)



# 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