VMIATX-5521

ISA bus TO VMEbus LINK

INSTRUCTION MANUAL

DOCUMENT NO. 500-305521-000 B

November 9, 1993

VME MICROSYSTEMS INTERNATIONAL CORPORATION
12090 SOUTH MEMORIAL PARKWAY
HUNTSVILLE, ALABAMA 35803-3308
(205) 880-0444
1-800-322-3616

NOTICE

The information in this document has been carefully checked and is believed to be entirely reliable. While all reasonable efforts to ensure accuracy have been taken in the preparation of this manual, VMIC assumes no responsibility resulting from omissions or errors in this manual, or from the use of information contained herein.

VMIC reserves the right to make any changes, without notice, to this or any of VMIC's products to improve reliability, performance, function, or design.

VMIC does not assume any liability arising out of the application or use of any product or circuit described herein; nor does VMIC convey any license under its patent rights or the rights of others.

VME Microsystems International Corporation

VMIC
SAFETY SUMMARY

THE FOLLOWING GENERAL SAFETY PRECAUTIONS MUST BE OBSERVED DURING ALL PHASES OF THE OPERATION, SERVICE, AND REPAIR OF THIS PRODUCT. FAILURE TO COMPLY WITH THESE PRECAUTIONS OR WITH SPECIFIC WARNINGS ELSEWHERE IN THIS MANUAL VIOLATES SAFETY STANDARDS OF DESIGN, MANUFACTURE, AND INTENDED USE OF THIS PRODUCT. VME MICROSYSTEMS INTERNATIONAL CORPORATION ASSUMES NO LIABILITY FOR THE CUSTOMER'S FAILURE TO COMPLY WITH THESE REQUIREMENTS.

GROUND THE SYSTEM

To minimize shock hazard, the chassis and system cabinet must be connected to an electrical ground. A three-conductor AC power cable should be used. The power cable must either be plugged into an approved three-contact electrical outlet or used with a three-contact to two-contact adapter with the grounding wire (green) firmly connected to an electrical ground (safety ground) at the power outlet.

DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE

Do not operate the system in the presence of flammable gases or fumes. Operation of any electrical system in such an environment constitutes a definite safety hazard.

KEEP AWAY FROM LIVE CIRCUITS

Operating personnel must not remove product covers. Component replacement and internal adjustments must be made by qualified maintenance personnel. Do not replace components with power cable connected. Under certain conditions, dangerous voltages may exist even with the power cable removed. To avoid injuries, always disconnect power and discharge circuits before touching them.

DO NOT SERVICE OR ADJUST ALONE

Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

DO NOT SUBSTITUTE PARTS OR MODIFY SYSTEM

Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the product. Return the product to VME Microsystems International Corporation for service and repair to ensure that safety features are maintained.

DANGEROUS PROCEDURE WARNINGS

Warnings, such as the example below, precede only potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed.

W ARNING

DANGEROUS VOLTAGES, CAPABLE OF CAUSING DEATH, ARE PRESENT IN THIS SYSTEM. USE EXTREME CAUTION WHEN HANDLING, TESTING, AND ADJUSTING.

VMIATX-5521
ISAbus to VMEbus LINK

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF FIGURES

LIST OF TABLES

SECTION 1

INTRODUCTION

1.1 FEATURES

The VMIATX-5521 is a high performance, yet easy-to-use, method of linking the ISA bus together with a VMEbus chassis via a cable and a pair of adapter boards. The VMIATX-5521 has the following features:

a. User-selectable configuration.
b. User-selectable VMEbus interrupts are translated to ISA interrupts.
c. VMEbus slot 1 controller capability.
d. Cable length up to 25 feet.
e. 80386SX or higher ISA bus processor has full access to VMEbus resources.
f. Four independent 512 K VME Windows for flexibility.

1.2 FUNCTIONAL DESCRIPTION

The VMIATX-5521 link provides A32, A24, and A16 VMEbus addressing within four independent "windows" appearing within ISA memory addressing space. The link can transfer data as 16- and 8-bit sizes, and unaligned transfers are supported. Each of the seven possible VMEbus interrupts may also be assigned to one of four ISA interrupts. All transfers are performed as ISA memory operations, so DMA transfers are not supported.

The windows through which the ISA bus machine accesses the VMEbus can be configured to make any 512 K part of VME addressing space visible. Each window may be used by a different task to access the same or different VME resources, or several windows may be used by a single task that requires different byte-swapping transfers. Additionally, several windows may be "stacked" to act as larger windows, thus making up to 2 Mbytes of contiguous VMEbus addressing space available at once to the PC.

The VMIATX-5521 link also supports byte swapping in hardware. Some form of byte swapping is required when interfacing a "big endian" processor like most VMEbus controllers with a "little endian" machine like the 386. Each VME Window can be configured to use different byte-swapping methods, providing a flexible and powerful means of transferring data efficiently between different processors.

1.2.1 Software Requirements

While the information in this manual is sufficient for creating a custom driver, the VMIATX-5521 is supported by a library of support routines available from VMIC as the VMIATX/SW-5521 Support Library, VMIC Document No. 520-305521-000. The VMIC library supports code development for 80386SX or higher ISA bus processors using the MS/DOS 5.0 or higher operating system.

Except for the initialization of four Window Registers, most of the setup procedure for the VMIATX-5521 link involves initializing interrupt control and VME-to-ISA interrupt translation registers. After power-up or reset, the VMIATX-5521 defaults with all interrupts disabled, so if no interrupts are needed, initialization is rather simple. See Section 4 for details concerning interrupt initialization.

1.2.2 Hardware Requirements

The VMIATX-5521 link requires a 16-bit ISA or compatible host machine with an installed Host Adapter Board, a compatible cable, and a VMEbus chassis with the VME Adapter Board installed. It is recommended that the host processor be an Intel-compatible 80386SX or higher.

Both the Host Adapter Board and the VME Adapter Board have several jumper options that should be set before installation. See Section 5 for details concerning these jumpers.

Many VMIATX-5521 registers have a default setting controlled by hardware jumpers that are user-configurable, but all register settings may be overwritten. Note, however, that nearly all registers will be re-initialized to default values on power-up, power interruption, or reset. See Section 4 for register details and Section 5 for jumper configurations.

1.3 REFERENCE MATERIAL LIST

The reader should refer to The VMEbus Specification for a detailed explanation of the VMEbus. The VMEbus Specification is available from the following source:

VMEbus International Trade Association
10229 N. Scottsdale Road
Scottsdale, AZ 85253
(602) 951-8866

Although there is no real definitive specification for the ISA bus, there are many books describing in detail how to program it. Edward Solari's book, AT-Bus-Design, is a good description of the bus from a hardware perspective. This book is published by:

Annabooks
12145 Alta Carmel Court
Suite 250
San Diego, CA 92128

Two useful articles concerning the byte-swapping issue have appeared in VMEbus Systems magazine: "Byte Ordering Problems" by John Black in the March-April 1987 issue, and "Is Byte Ordering a Certification Issue" by Joel Witt in the April 1989 issue. Information regarding reprints is available at the following address:

VMEbus Systems Magazine
2618 S. Shannon
Tempe, AZ 85282
(602) 967-5581

SECTION 2

PHYSICAL DESCRIPTION AND SPECIFICATIONS

REFER TO 800-305521-000 SPECIFICATION

SECTION 3

THEORY OF OPERATION

3.1 OPERATIONAL OVERVIEW

The VMIATX-5521 allows an ISA-equipped PC to become a VMEbus master and access VMEbus slaves. The VMIATX-5521 also passes user-selected interrupts originating on the VMEbus to the ISA bus processor for servicing. In the absence of a VMEbus slot 1 controller, the VMIATX-5521 can become the controller by supplying the following:

a. 16 MHz SYSCLK.
b. IACK*/IACKIN* daisy chain driver.
c. VMEbus SYSRESET* at power-up or under software control.

3.2 BASE ADDRESS SELECTION

The VME Adapter Board functions only as a VMEbus master – it does not have a VMEbus slave interface and therefore has no base address. The Host Adapter Board occupies two user-selectable address spaces:

a. ISA I/O Space – 16 bytes of space may reside on any natural 16-byte boundary beginning at $0100.
b. ISA Memory Space – 2 Mbytes of space may reside on any natural 2048 K boundary except zero.

Although the Host Adapter Board will respond when set to the highest possible 2048K memory base address of $E00000, the system ROM that typically lies at the top of that 2,048 K will prevent the VMIATX-5521 programmer from accessing the full 512 K of the highest VME Window. It is therefore recommended that a base memory address somewhere between the first and last possible location be used.

3.3 ADDRESSABLE SLAVES

The VMIATX-5521 can address any VMEbus slave. The following restrictions apply, however:

a. VMEbus Read-Modify-Write access is not supported.
b. VMEbus Block Transfer mode is not supported.
c. True VMEbus 32-bit longword data access is not supported due to the limitations of the 16-bit ISA bus.

3.4 BUS TIMEOUTS AND BUS ERRORS

The VMIATX-5521 links two functionally incompatible buses: the VMEbus and the ISA bus. Cycle timing between the two types of buses is radically different and is completely constrained by the ISA bus timing. The ISA bus specifies a maximum time limit for an ISA bus transaction to occur. Since the VMIATX-5521's function is to make remote VMEbus resources appear as local ISA bus resources, an excursion outside this maximum time limit has some probability of occurring. This could happen, for example, if the VME Adapter of the VMIATX-5521 resides in a VMEbus chassis with several other VMEbus masters. If the VMEbus in this case is heavily used by the resident masters, and the ISA-linked master requests use of the VMEbus, there is no VMEbus specified time limit for the bus owning master to yield to the requester. In this instance, the cycle initiated on the ISA bus could easily overrun its maximum cycle time. To circumvent a host timeout, the VME Adapter of the VMIATX-5521 activates a timer upon a valid access to any VMEbus space. If the VMIATX-5521 is not granted the VMEbus by the VMEbus controller within a specified period (about 10 sec), the Rerun Response bit in the Error Register will be set and remain so until cleared.

Another potential timeout can occur as the result of a nonresponsive or missing VMEbus slave. To handle this situation, the VME Adapter activates another timer after being granted the VMEbus. If the addressed VMEbus slave does not respond within the timeout period of this second timer, or if a VME bus error occurs for any reason, the Error Response bit in the Error Register will be set and remain so until cleared.

SECTION 4

PROGRAMMING

4.1 PROGRAMMING

It is recommended that the VMIATX-5521 programmer use the VMIATX-5521 Support Library to ease code development for the link. The VMIC driver for the VMIATX-5521 is available as part number VMIATX/SW-5521, and its use is documented in the software driver User's Guide Document Number SWG-305521-000. All information necessary for writing a custom driver is available in this section.

Although the VMIATX-5521 only requires the user to initialize four Window Registers before accessing the VMEbus, some other registers are present to facilitate user information and interrupt generation. While the four Window Registers are always accessible at their user-selectable I/O addresses, the remaining registers are accessible only through a VME Window whose Window Register contains a value of $02. Table 4.1-1 details the VMIATX-5521 Host Adapter Registers. Table 4.1-2 details the VME Windows.

The following addresses are in ISA I/O space and offset relative to the base I/O address set by jumpers on the host adapter board.

Table 4.1-1. Host Adapter Registers

RELATIVE ADDRESS	DESCRIPTION	CONTENTS	ACCESS MODE
$00	Window Register 0	Address Control Byte	Byte (R/W)
$01	Data Control Register 0	Byte-Swapping Control	Byte (R/W)
$02	Window Register 1	Address Control Byte	Byte (R/W)
$03	Data Control Register 1	Byte-Swapping Control	Byte (R/W)
$04	Window Register 2	Address Control Byte	Byte (R/W)
$05	Data Control Register 2	Byte-Swapping Control	Byte (R/W)
$06	Window Register 3	Address Control Byte	Byte (R/W)
$07	Data Control Register 3	Byte-Swapping Control	Byte (R/W)
$08	IRQ 10 Control Register	Interrupt Control Bit	Byte (R/W)
$09	IRQ 11 Control Register	Interrupt Control Bit	Byte (R/W)
$0A	IRQ 12 Control Register	Interrupt Control Bit	Byte (R/W)
$0B	IRQ 15 Control Register	Interrupt Control Bit	Byte (R/W)
$0C	Interrupt 0-3 Assignment Register	Interrupt Assignments	Byte (R/W)
$0D	Interrupt 4-7 Assignment Register	Interrupt Assignments	Byte (R/W)
$0E	Error Register	Error Status Bits	Byte (R-Reset)
$0F	Interrupt Request Register	Interrupt Request Bits	Byte (R)

M5521ATX/T4.1-1

The following addresses are in ISA memory space and offset relative to the base memory address set by jumpers on the host adapter board.

Table 4.1-2. VME Windows

RELATIVE ADDRESS	DESCRIPTION	CONTENTS	ACCESS MODE
$000000 to $07FFFF	VME Window 0	VMEbus Addresses	Any (R/W)
$080000 to $0FFFFF	VME Window 1	VMEbus Addresses	Any (R/W)
$100000 to $17FFFF	VME Window 2	VMEbus Addresses	Any (R/W)
$180000 to $1FFFFF	VME Window 3	VMEbus Addresses	Any (R/W)

M5521ATX/T4.1-2

The following addresses are offset within any VME window whose corresponding window register contains a value of $02.

Table 4.1-3. VME Adapter Registers

RELATIVE ADDRESS	DESCRIPTION	CONTENTS	ACCESS MODE
$13	VME Adapter Board Control and Status Register	Status and Control Bits	Byte (R/W)
$1A	VME Extended Address Register	A24-A32 Bits	Byte (R/W)
$1E	VME User-Defined Extended Address Register	A24-A32 Bits	Byte (R/W)
$1F	VME User-Defined Address Modifier Register	AM0-AM5 Bits	Byte (R/W)
$23	VME Interrupt Status Register	Interrupt Status Bits	Byte (R)
$27	VME Interrupt 1 Acknowledge Register	Interrupt Vector	Byte (R)
$2B	VME Interrupt 2 Acknowledge Register	Interrupt Vector	Byte (R)
$2F	VME Interrupt 3 Acknowledge Register	Interrupt Vector	Byte (R)
$33	VME Interrupt 4 Acknowledge Register	Interrupt Vector	Byte (R)
$37	VME Interrupt 5 Acknowledge Register	Interrupt Vector	Byte (R)
$3B	VME Interrupt 6 Acknowledge Register	Interrupt Vector	Byte (R)
$3F	VME Interrupt 7 Acknowledge Register	Interrupt Vector	Byte (R)

M5521ATX/T4.1-3

4.2 HOST ADAPTER REGISTERS

Sixteen-byte registers exist in ISA I/O space offset relative to the base I/O address set by jumpers on the Host Adapter Board (see Section 5.4 for details). These registers physically exist on the Host Adapter Board and consist of four Window Registers and four Data Control Registers corresponding to the four VME Windows.

In addition, these registers include four IRQ Control Registers, two Interrupt Assignment Registers, and an Interrupt Request Register, all for managing VME-to-ISA interrupt conversion. There is also an Error Register whose contents can flag various interface errors.

4.2.1 Window Registers

There are four Window Registers in ISA I/O space offsets at $00, $02, $04, and $06, respectively controlling the four VME Windows 0-3. The Window Register contains a single value, the Address Control Byte. This value determines what portion of VMEbus addresses may be accessed within the corresponding VME Window. Table 4.2.1-1 lists the Address Control Byte values necessary to make the lowest addresses within the listed addressing mode available. **The Window Registers must always be initialized before accessing the VME Windows.**

Table 4.2.1-1. Address Control Byte Values

VMEbus Access Type	Address Control Byte
VME Adapter Board Registers	$02
Short I/O (A16) Nonprivileged or Supervisory	$04
Extended (A32) Nonprivileged	$40
Extended (A32) Supervisory	$60
Standard (A24) Nonprivileged	$80
Standard (A24) Supervisory	$A0
User-Defined	$C0

M5521ATX/T4.2.1-1

When a Window Register is initialized to an Address Control Byte value from Table 4.2.1-1, host accesses within the associated VME Window will actually access VMEbus space according to the VMEbus access type listed. The values from Table 4.2.1-1 will cause the lowest 512 K of the listed VMEbus space to be available through the VME Window — subsequent 512 K pages may be accessed by incrementing the Address Control Byte by one. Each such increment will move the VME Window up 512 K within the VMEbus addressing space selected.

Since the Window Register contents are undefined at power-up, it is always necessary to initialize the Window Registers before accessing the VME Windows. Also, special cases exist for Short I/O and User-Defined addressing modes. See Section 4.5 for programming details concerning these special cases and more detail concerning VMEbus address paging.

4.2.2 Data Control Registers

There are four Data Control Registers at offsets $01, $03, $05, and $07, respectively, controlling data transfers through VME Windows 0 through 3. The value in

the Data Control Registers determines the byte-swapping technique used for all data transfers through the corresponding VME Window. The register consists of 2-bit pairs, the lowest order pair (bits 0 and 1) controlling byte-swapping during word (16-bit) and longword (32-bit) transfers, while the next bit pair (bits 2 and 3) controls byte positioning within a larger quantity during byte transfers only. Table 4.2.2-1 shows the format of the Data Control Registers.

Table 4.2.2-1. Data Control Register Format

Bit 3	Bit 2	Bit 1	Bit 0	Function
X	X	0	0	No swapping during word or longword transfers
X	X	0	1	Swap bytes during word transfers
X	X	1	0	Swap words during longword transfers
X	X	1	1	Swap bytes or words during word or longword transfers
0	0	X	X	No byte shifting during byte transfers
0	1	X	X	Shift byte position by 1 (invert bit 0 of byte address)
1	0	X	X	Shift byte position by 2 (invert bit 1 of byte address)
1	1	X	X	Shift byte position by 3 (invert bits 0 and 1 of byte address)

M5521ATX/T4.2.2-1

Bits 4 through 7 of the Data Control Registers are undefined. Bits 0 through 3 are reset to zero after reset or power-up, causing no data swapping by default. See Section 4.5.1 for details on the constructive use of byte swapping.

4.2.3 IRQ Control Registers

The IRQ Control Registers exist at offsets $08, $09, $0A, and $0B for IRQ-10, IRQ 11, IRQ 12, and IRQ 15, respectively. These registers contain a single valid bit of information, bit 0: the Interrupt Control Bit. This bit controls whether or not the Host Adapter Board may generate the ISA interrupt associated with it. The Interrupt Control Bit is the functional equivalent of an interrupt mask control bit. All other bits in this register are undefined.

After power-up or reset, the Interrupt Control Bit is in its default state: low. A clear Interrupt Control Bit inhibits the generation of the associated interrupt by the Host Adapter Board. Software must set this bit to enable the associated interrupt, although if an interrupt is generated, the bit will be automatically cleared again by hardware to prevent multiple interrupt requests. Therefore, if the associated interrupt is being used, the host interrupt service routine should always set this bit before terminating to reenable the interrupt.

4.2.4 Interrupt Assignment Registers

The Interrupt Assignment Registers exist at I/O offset addresses $0C and $0D for interrupts 0 through 3 and 4 through 7, respectively, allow the user to steer VMEbus interrupts 1 through 7, plus a special error signal, to any of the four available

ISA bus interrupts (IRQ10, IRQ11, IRQ12, or IRQ15). Each register contains 4-bit pairs associated with a possible interrupt source according to Table 4.2.4-1.

Table 4.2.4-1. Interrupt Assignment Register Format

Interrupt 0-3 Assignment Register	**Bits 7 and 6**	**Bits 5 and 4**	**Bits 3 and 2**	**Bits 1 and 0**
Interrupt Source	VMEbus level 3	VMEbus level 2	VMEbus level 1	Interface Error

Interrupt 4-7 Assignment Register	**Bits 7 and 6**	**Bits 5 and 4**	**Bits 3 and 2**	**Bits 1 and 0**
Interrupt Source	VMEbus level 7	VMEbus level 6	VMEbus level 5	VMEbus level 4

M5521ATX/T4.2.4-1

Each possible interrupt source may be assigned one of four destination ISA bus interrupts according to Table 4.2.4-2. Multiple interrupt sources may be assigned the same physical ISA bus interrupt, but only one interrupt may be processed at a time, since the associated clear IRQ Control Bit will prevent further interrupts while one is being processed.

Table 4.2.4-2. Interrupt Assignment Register Destination

High-order Bit	**Low-order Bit**	**Destination**
0	0	ISA IRQ 10
0	1	ISA IRQ 11
1	0	ISA IRQ 12
1	1	ISA IRQ 15

M5521ATX/T4.2.4-2

4.2.5 Error Register

The Error Register exists at offset $0E in I/O space. If read, its contents will be as shown in Table 4.2.5-1, and any nonzero value indicates that some kind of interface error has occurred. Any write to this register clears all bits.

Table 4.2.5-1. Error Register Format

Bits 7-3	**Bit 2**	**Bit 1**	**Bit 0**
00000	VMEbus Power Bad	Rerun Response	Error Response

M5521ATX/T4.2.5-1

This register is cleared after power-up, reset, or whenever a write operation is performed here. Each of the three meaningful bits in this register indicates some sort of serious error if set:

a. **VMEbus Power Bad (Bit 2):** If set, this bit indicates that the external VMEbus chassis +5 V power has dropped below a critical level since the last time this bit was cleared. Note that this bit does not indicate the current condition of the remote power — it will be set by even a momentary power drop and remain set until cleared.

b. **Rerun Response (Bit 1):** If set, this bit indicates that the VME Adapter has returned a RERUN code at least once since the last time this bit was cleared. A RERUN implies that the VME Adapter was unable to obtain control of the VMEbus under the maximum time limit allowed. This can be caused by another VMEbus master refusing to relinquish control quickly. Note that this bit does not indicate that the most recent access failed — it will be set by any RERUN response from the VMEbus adapter board and remain set until cleared.

c. **Error Response (Bit 0):** If set, this bit indicates that the VME Adapter has returned an ERROR code at least once since the last time this bit was cleared. An ERROR implies some sort of VMEbus error, typically caused by an attempt to access a bus address at which nothing exists, although other situations may also cause a VMEbus error. Note that this bit does not indicate that the most recent access failed — it will be set by any ERROR response from the VMEbus adapter board and remain set until cleared.

4.2.6 Interrupt Request Register

The read-only Interrupt Request Register exists at I/O offset $0F and contains four interrupt activity flags corresponding to the four possible ISA bus interrupts. The format of the Interrupt Status Register is given in Table 4.2.6-1.

Table 4.2.6-1. Interrupt Request Register Format

Bits 7-4	Bit 3	Bit 2	Bit 1	Bit 0
Undefined	ISA IRQ 15	ISA IRQ 12	ISA IRQ 11	ISA IRQ 10

M5521ATX/T4.2.6-1

Any bit set in the Interrupt Request Register indicates that an interrupt of the corresponding type is pending. The actual ISA interrupt may or may not have occurred, depending upon the setting of the associated IRQ Control Registers. The Interrupt Request Register allows for software interrupt polling, since the flags here will always give positive indication of a pending interrupt regardless of whether or not the actual hardware interrupt is enabled.

4.3 VME WINDOWS

The four 512 K VME Windows fill the entire 2 Mbytes of memory space occupied by the Host Adapter Board without any incongruities. Any accesses to a VME Window will be translated to VMEbus accesses as long as the corresponding Window

Register has been initialized first. Actually, initializing **any** of the Host Adapter's I/O registers will enable access to the VME Windows, but an access to a VME Window whose Window Register is uninitialized will cause some undefined access and perhaps an error. It is prudent to always initialize the Window Registers after power-up or reset before accessing the VME Windows.

Aside from having to initialize the Window Registers first, the only inherent constraint upon VME Window accesses is one imposed by the ISA bus itself: true longword transfers are not possible. While software may be written that transfers data as 32-bit longword quantities, these are translated into two separate 16-bit word transfers by the ISA bus hardware. Since the VME resources on the other side of the VME Window are being accessed via the ISA bus, the VME resources can only see two word accesses and not a true longword access. Of course, VME resources accessed through the VME Window may have their own limitations (e.g., registers that can only be addressed as 16-bit words, read-only quantities, etc.), but the only restriction concerning the VMIATX-5521 link itself is that true longword transfers are not possible.

NOTE

THE SAME ISA HARDWARE WHICH CREATES TWO 16-bit TRANSFERS FROM ONE 32-bit TRANSFER WILL CREATE THREE TRANSFERS (8-bit, 16-bit, AND THEN ANOTHER 8-bit) IF THE 32-bit ACCESS IS TO AN ODD ADDRESS IT WILL ALSO CREATE TWO 8-bit TRANSFERS FROM A SINGLE 16-bit TRANSFER TO AN ODD ADDRESS.

4.4 VME ADAPTER REGISTERS

As their name implies, the VME Adapter Registers exist on the VME Adapter Board and control certain VMEbus-specific aspects of the VMIATX-5521 link, including extended address generation and VMEbus interrupt status and acknowledgment. Since these registers are physically located on the VME Adapter Board, the cable between the Host Adapter and the VME Adapter must be connected and the VME Adapter Board must be powered up before any access to these registers is possible.

If the above conditions are met, these registers are available within any VME Window whose Window Register contains a value of $02. The corresponding Data Control Register for this VME Window should also contain a value of $00 to prevent any byte swapping while accessing these registers. All addresses listed for these registers are offsets from the base of the selected VME Window.

4.4.1 Adapter Board Control and Status Register

The VME Control and Status Register exists at offset $13 within the VME Window and conveys information concerning the status of the adapter boards and controls certain global aspects of the VME Adapter. The register is defined according to Table 4.4.1-1.

Table 4.4.1-1. Adapter Board Control and Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Interrupt Control (Read/Write)	VMEbus Reset (Write Only)	Slot 1 Flag (Read Only)	Reserved (Read Only)	Host Adapter Power Indicator (Read Only)	VMEbus Adapter Power Indicator (Read Only)
X	X	0=Enabled 1=Disabled	1=Reset	0=Slot 1 1=Nonslot 1	1	0=bad 1=good	0=bad 1=good

M5521ATX/T4.4.1-1

VME status information can be read from this register, the VMEbus may be reset by setting bit 4, and all passing of VME interrupts to the host may be disabled by setting bit 5. If the entire VMEbus chassis is reset by setting the VMEbus Reset bit, the host should wait at least 300 milliseconds before accessing the VMEbus for the reset to propagate. Both Power Indicator bits reflect only the status of the +5 V logic supplies and convey no information about other supply voltages. The Interrupt Control bit is set after power-up or reset, including a reset generated by setting the VMEbus Reset bit, disabling all possible VMEbus interrupts (but not the error interrupts — see Section 4.2 for details concerning the error interrupt and its assignment). The Slot 1 Flag bit is set or cleared depending upon whether or not the VME Adapter Board is jumpered for slot 1 (jumper 1B — see Section 5.3 for details).

4.4.2 VME Extended Address Register

The VME Extended Address Register exists at offset $1A within the VME Window. It contains the high-order VME address bits required when VME Extended (A32) addressing is selected. Note that this register physically exists in the VME Adapter board and not in the Host Adapter. All eight bits are used to represent address bits A31 through A24: bit 0 provides address bit A24 and bit 7 provides A31.

The VME Extended Address Register serves no function unless a VME Window is accessed whose Window Register has a value between $40 and $7F, inclusive, indicating that accesses within that VME Window are 32-bit Extended addresses. If the Window Register is in this range, the value in this register represents the high-order bits of the VMEbus effective address, while the low-order bits are represented by the ISA offset address within the VME Window and the VME Window offset within VME Extended address space. See Section 4.3 for programming details.

Note that there is only one VME Extended Address Register, even though its value will apply to any of the four VME Windows set for 32-bit Extended addressing. If multiple VME Windows require 32-bit addressing at different addresses, it is imperative that the host program ensure the proper VME Extended Address Register value is installed before accessing the VME Window. If an interrupt service routine requires VMEbus Extended addressing, it is recommended that the value of this register be saved upon entry to the routine and restored upon exit.

The default value of this register after power-up or reset is selected by jumpers 3A (bit 7) through 3H (bit 0) on the VME Adapter. See Section 5.3 for jumper configuration details.

4.4.3 VME User-Defined Registers

The VMIATX-5521 directly supports the standard VMEbus addressing spaces, including Extended (A32), Standard (A24), and Short I/O (A16), all in both supervisory and nonprivileged modes. These address spaces are generally controlled by the Window Register value (reference Table 4.2.1-1), which causes the VMIATX-5521 hardware to automatically generate the appropriate Address Modifier bits (AM0 through AM5) on the VMEbus. While this scheme will suffice for most users, some may require a special VMEbus Address Modifier code for unusual purposes, in which case user-defined VMEbus addressing should be used.

There are two registers associated with user-defined addressing: the VME User-Defined Extended Address Register and the VME User-Defined Address Modifier Register. Selecting User-Defined addressing is straightforward: any VMEbus addressing through a VME Window whose Window Register contains a value from $C0 through $DF will be augmented by the two VME User-Defined Registers detailed below.

a. VME User-Defined Extended Address Register:

The VME User-Defined Extended Address Register exists at offset $1E within the VME Window and is nearly identical in function to the VME Extended Address Register (see Section 4.4.2) in that it supplies the high-order address bits (A31 through A24) for the VMEbus during VME Window transactions. Its format is exactly the same as the other register as well: all eight bits are used, bit 0 provides address bit A24 and bit 7 provides A31. The difference is that the VME User-Defined Extended Address Register comes into play only during User-Defined VMEbus accesses, and even then its value will be meaningless unless the User-Defined Address Modifier causes all 32 address bits to be used.

The default power-up/reset value for the VME User-Defined Extended Address Register is set by jumpers 4A (bit 7 or A31) through 4H (bit 0 or A24). See Section 5.3 for jumper configuration details.

b. VME User-Defined Address Modifier Register:

The VME User-Defined Address Modifier Register exists at offset $1F within the VME Window and controls the custom VMEbus Address Modifier bits during transactions through a VME Window configured for User-Defined access. Table 4.4.3-1 shows the format of this register.

Table 4.4.3-1. VME User-Defined Address Modifier Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X	X	AM5	AM4	AM3	AM2	AM1	AM0

M5521ATX/T4.4.3-2

A complete list of access modes controlled by the Address Modifier bits is listed in Table 3 of the VMEbus Specification, but some common codes are listed in Table 4.4.3-2.

Table 4.4.3-2. Common VMEbus Address Modifier Codes

AM5	AM4	AM3	AM2	AM1	AM0	HEX	ACCESS TYPE AND MODE
0	0	1	0	0	1	$09	Extended (A32) nonprivileged data
0	0	1	0	1	0	$0A	Extended (A32) nonprivileged program
0	0	1	1	0	1	$0D	Extended (A32) supervisory data
0	0	1	1	1	0	$0E	Extended (A32) supervisory program
1	0	1	0	0	1	$29	Short I/O (A16) nonprivileged data
1	0	1	1	0	1	$2D	Short I/O (A16) supervisory data
1	1	1	0	0	1	$39	Standard (A24) nonprivileged data
1	1	1	0	1	0	$3A	Standard (A24) nonprivileged program
1	1	1	1	0	1	$3D	Standard (A24) supervisory data
1	1	1	1	1	0	$3F	Standard (A24) supervisory program

M5521ATX/T4.4.3-2

The power-up/reset default value of the VME User-Defined Address Modifier Register is set by jumpers 4J (AM5) through 4O (AM0). See Section 5.3 for jumper configuration details.

4.4.4 VME Interrupt Status Register

The VME Interrupt Status Register exists at offset $23 within the VME Window and contains the current state of the VMEbus interrupt request lines. The format of this register is given in Table 4.4.4-1.

Table 4.4.4-1. VME Interrupt Status Register Format

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	0

M5521ATX/T4.4.4-1

After receiving an interrupt, the host should read the Interrupt Status Register to determine which VME interrupt occurred. Any valid bits set in the VME Interrupt Status Register indicate an active interrupt. The interrupt request level is directly related to the bit position within the register, thus bit 1 corresponds to IRQ1*, bit 2 corresponds to IRQ2*, and so on. Note that bit 0 is always low since the VMEbus does not have an "IRQ0*". After determining the interrupt level, the host should then read the corresponding Interrupt Acknowledge Register of the highest priority request to obtain the interrupt vector, then service the interrupt. Any bits set in the Interrupt Status Register will remain set until the interrupt has been serviced.

4.4.5 VME Interrupt Acknowledge Registers

There are seven VME Interrupt Acknowledge Registers at offsets $27, $2B, $2F, $33, $37, $3B, and $3F, respectively, corresponding to VME interrupts 1 through 7. By reading one of these registers, the VME Adapter will perform a corresponding interrupt acknowledge cycle on the VMEbus. The 8-bit data value returned is the vector received from the VMEbus slave that caused the interrupt.

4.5 PROGRAMMING DETAILS

4.5.1 Byte-Swapping Details

The following section describes byte/word swapping from the user perspective (i.e., memory-to-memory), but first a brief discussion of the differences between little endian and big endian is in order. Little endian processors such as the 80386 store multiple-byte values in memory with the least significant byte stored in the lowest address. By contrast, big endian processors such as the 68030 store multiple-byte values in memory with the most significant byte stored in the lowest address. Thus, data saved into memory by two processors with different storage techniques cannot automatically be shared because some data will be "backwards". There is no simple way of correcting this problem because different types of data (e.g.,character and floating-point data) need to be corrected in different ways. For this reason, the user must know beforehand what kind of data is to be transferred. For example, character data (1 byte) is stored in memory exactly the same way by either type of processor. Floating-point data (4 bytes) is stored in memory with the bytes in reverse order.

The PC is a little endian machine. As long as the VMIATX-5521 link is only connected to other little endian processors, or to no processor within the VME chassis, byte swapping is not necessary. Since each processor can interpret the other's data accurately, it is best to clear all the bits in the Data Control Registers and operate with

no byte swapping at all. Big endian machines (such as most VMEbus processors) must communicate with the PC (via shared memory, for example), the ISA bus host will have to use some byte-swapping techniques to transfer data without confusion. Note that the burden of byte swapping is on the little endian machine, since other VMEbus processors do not likely have any byte-swapping mechanism available. Byte swapping could be handled exclusively with software significant performance impact. The hardware on-board the VMIATX-5521 ISA Adapter swaps bytes with no loss in throughput.

The least significant bits (bits 0 and 1) in the Data Control Register affect only 16-bit or larger transfers, while the next two bits (bits 2 and 3) affect only byte transfers. Since byte swapping is usually only necessary during word or longword transfers, all possible settings for the first two bits will be detailed below.

If no swap bits in the Data Control Register are set, then no swapping occurs during a transfer regardless of the transfer size. In other words, if a transfer is performed, the byte(s) will be stored in memory at the same offset address in the destination machine as they were in the source machine. Eight distinct possibilities are shown in Figure 4.5.1-1.

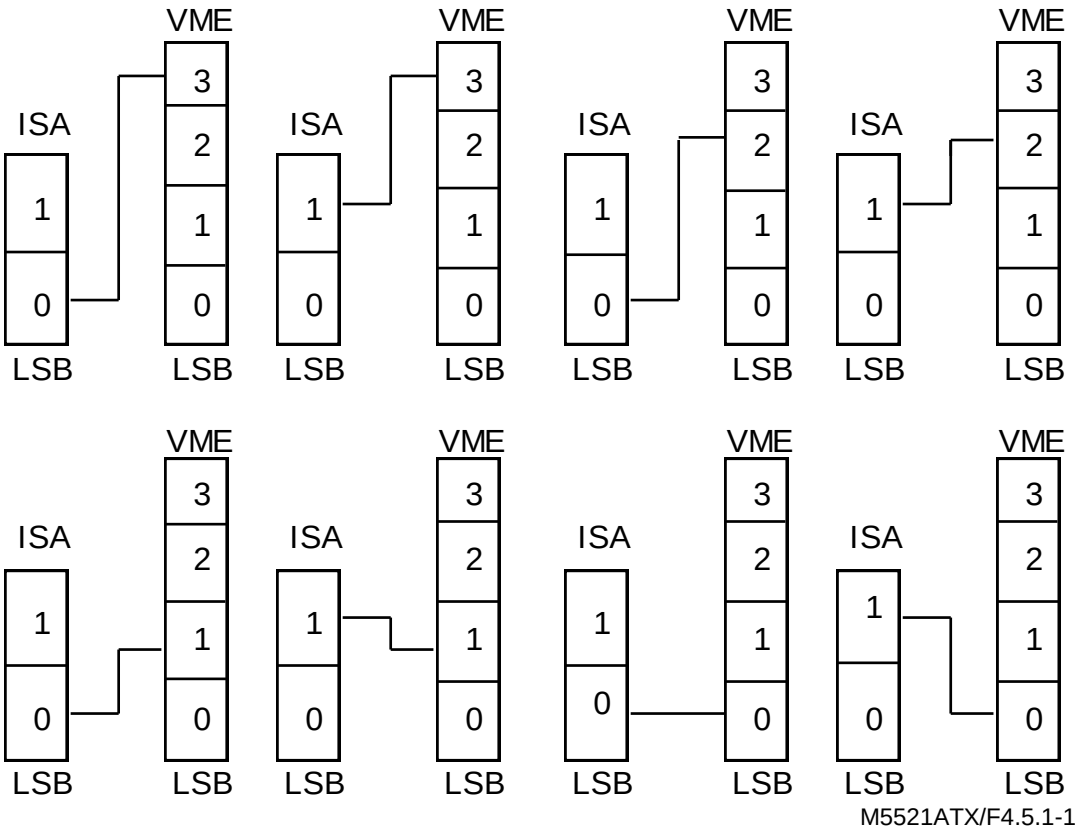


Figure 4.5.1-1. Byte Transfers

If the Data Control Register value is 0001 binary (i.e., byte-swap bit set) then byte swapping will be employed on all transfers. If a transfer is performed, the data will

be stored in the destination machine's memory with the bytes within the words swapped with the memory locations they occupied in the source machine.

If the Data Control Register value is 0010 binary (i.e., word-swap bit set) then word swapping will be employed on all word transfers. If a transfer is performed, the data will be stored in the destination machine's memory with the words (16 bits) within the longwords (32 bits) swapped with the memory locations they occupied in the source machine. Four distinct possibilities are given in Figure 4.5.1-2.

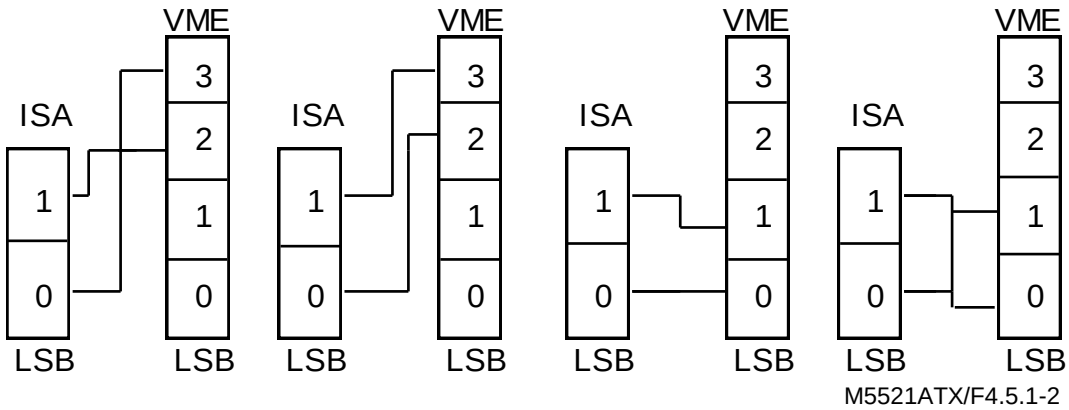


Figure 4.5.1-2. Word Transfer

If the Data Control Register value is 0011 binary (i.e., byte and word swap bits set) then word swapping and byte swapping will be employed on all longword transfers. If a transfer is performed, the data will be stored in the destination machine's memory with the words (16 bits) within the longwords (32 bits) swapped and the bytes within the words swapped with the memory locations they occupied in the source machine.

Bits 2 and 3 of the Data Control Register affect only byte transfers and essentially allow a byte transfer take place at a byte address offset from +0 through +3 relative to the base address of the transfer (see Table 4.2.2-1 for details). This feature may be useful for accessing component bytes within a larger data quantity (e.g., words or longwords), but since the VMIATX-5521 allows byte transfers on both even and odd addresses, it is usually simpler just to access the appropriate byte directly with software and leave bits 2 and 3 clear. It is important to understand that although bytes may be address at even or odd addresses, words and longwords must be addressed at even addresses only in order for the swapping hardware to function as described previously.

4.5.2 VMEbus Addressing

Each of the four VME Windows can be configured to access a portion of VMEbus addressing space in any mode (e.g., standard, extended, nonprivileged, etc.). The particular VME address and access mode within any VME Window depends

primarily upon the value of the Address Control Byte in the Window Register corresponding to that VME Window.

Depending upon the Address Control Byte value, the value in the VME Extended Address Register or the VME User-Defined Registers or neither may also be used to generate an effective VMEbus address. For the usual VME access modes (i.e., short I/O, standard, extended, supervisory or nonprivileged access), the VME address is determined by the Address Control Byte augmented by the VME Extended Address Register if extended addressing is used. If an unusual VME access mode is necessary, User-Defined VME Addressing may be used, wherein the effective VMEbus address, address modifier, and extended address bits (if any) are determined by the Address Control Byte in conjunction with the VME User-Defined Address Modifier Register (and the VME User-Defined Extended Address Register if the custom Address Modifier code uses address bits A31 through A24).

It is important to note that while this documentation mentions "initializing" the Address Control Byte to a certain value, then “Incrementing” it to obtain the proper value, this exact process is not necessary: if the proper Address Control Byte value is simply loaded into the Window Register, the result is the same. The "initialize and increment" method is simply easier to explain and document.

In nearly all cases, the user will want to access the VMEbus in one of the "normal" modes: short I/O, standard, or extended addressing in either supervisory or nonprivileged modes. In these cases, the VMEbus address is determined either by the Address Control Byte of the Window Register alone, or in conjunction with the VME Extended Address Register in the case of extended addressing.

For short I/O VMEbus access, the Address Control Byte should be fixed at $04. When the Address Control Byte is $04, the lowest 64 K of the VME Window will access VMEbus short I/O space in nonprivileged mode. The second 64 K within the VME Window will access the same short I/O space, but in supervisory mode. Thus, if a VMEbus I/O board configured for short I/O supervisory addressing at $0100 needed to be accessed across a VMIATX-5521 link, all that would be necessary would be to select any of the four available VME Windows, initialize its Window Register's Address Control Byte to $04, select the appropriate byte-swapping scheme and configure the Data Control Register accordingly, then address the VME Window at an offset of $10100 from the VME Window's base address. The address of $10100 was created by taking the board address of $0100 and adding an offset of $10000 (64 K) for supervisory access. VME Window address offsets greater than 128 K are undefined when the Address Control Byte is $04.

For standard (A24) VMEbus access, the Address Control Byte should be initialized to either $80 for nonprivileged access or $A0 for supervisory access. This initial value will cause the VME Window to make available the lowest VMEbus standard addresses, thus the base VME Window address will access VMEbus standard address $000000, and a VME Window address offset of $04 will access the VMEbus at standard $000004 and so on, through the entire VME Window.

The amount of VMEbus addressing space available at once is 512 K — the size of one VME Window. Since the VMEbus addressing region is much larger

(16 Mbytes for standard mode, 4 Gbytes for extended mode), it will be necessary to increment the Address Control Byte to access greater VMEbus addresses. Each increment above the initial Address Control Byte value will cause the VME Window to access the next 512 K page of VMEbus addresses.

For example, an Address Control Byte value of $80/$A0 will make the first 512 K of VMEbus standard addresses available within the VME Window while an incremented Address Control Byte value of $81/$A1 will make the next 512 K available (that is, the base VME Window address will correspond to a VMEbus standard address of $080000), and so on up to an Address Control Byte value of $9F/$BF, which will make the last 512 K of VMEbus standard addressing space available (that is, the base VME Window address will correspond to a VMEbus standard address of $F80000).

Extended VMEbus addresses within a VME Window may be generated in a manner similar to that for standard addresses detailed previously, with a few major differences: the initial Address Control Byte value is $40 for extended nonprivileged access mode or $60 for extended supervisory access, and the equivalent standard address is augmented by the value in the VME Extended Address Register, which creates the high-order bits of the extended address.

For an example of extended VME addressing over the VMIATX-5521 link, assume a PC with a VMIATX-5521 link installed needs to access a VME board configured for extended nonprivileged address $F9E00100. Given this situation, the Address Control Byte of the chosen Window Register should be initialized to $40, and the high-order byte of the target address (in this case, $F9) should be loaded into the VME Extended Address Register. With these two registers initialized, the VME Window will begin at VME extended address $F9000000 and all that is necessary is to increment the Address Control Byte to allow access to the target. The number of increments needed is calculated based on the lower 24 bits of the target address (here, $E00100) just as the standard addresses described previously: in this case, the value must be incremented 28 times (or $1C HEX, from $E00000 target base address/512 K VME Window size) to allow the VME Window to access VMEbus extended addresses in the $E00000 to $EFFFFF range, resulting in an Address Control Byte value of $5C. With the Address Control Byte set to $5C and the VME Extended Address Register holding $F9, offset $100 into the VME Window will actually address the VME board at extended VME address $F9E00100, the target address.

It is important to realize that there is only one VME Extended Address Register, and its value is shared by all VME Windows configured for extended VME addressing. Multiple VME Windows configured for extended addressing with different high-order bytes must be managed carefully by the host to ensure the proper value is contained in the VME Extended Address Register before accessing the VME Window.

In case the "normal" VMEbus addressing modes (i.e., short I/O, standard, extended, supervisory or nonprivileged access) are insufficient for an application, the VMIATX-5521 also supports custom user-defined addressing, where the user supplies the VMEbus Address Modifier bits. In this case, the effective VMEbus address and access mode is determined by the Address Control Byte of the Window Register in

conjunction with the VME User-Defined Address Modifier Register. As usual, the Data Control Register controls which byte-swapping technique are used on all transfers through the VME Window, but do not otherwise affect VME addressing.

If user-defined addressing is desired, the Address Control Byte of the appropriate Window Register should be initialized to $C0 and the custom address modifier should be loaded into the corresponding field in the VME User-Defined Address Register along with the extended address bits A31-A24 in the VME-User-Defined Extended Address Register, if needed. Once this is done, the lowest address in the VME Window will access the lowest address within the desired VMEbus address space, offset by the value of the VME User-Defined Extended Address Register if the custom access mode does not ignore A31-A24 (the high-order address bits are **always** generated on the VMEbus, although they may be ignored depending on the address modifier). From there, increasing address offsets within the VME Window correspond directly to increasing addresses on the VMEbus, up through the top of the VME Window.

Since the size of the VME Window is only 512 K, it is likely that the entire user-defined addressing space will not fit within a single VME Window. Greater addresses within the user-defined address space may be generated by incrementing the Address Control Byte up to a maximum Address Control Byte value of $DF. Each such increment will cause the base VME Window address to move up through the user-defined addressing space by an amount equal to the VME Window size (512 K), just as described in the previous section. In fact, user-defined addressing is exactly the same as normal addressing with two exceptions: the VME Address Modifier is user specified rather than automatically generated based on the Address Control Byte, and the extended address bits, if used, are generated from the VME User-Defined Extended Address Register rather than the VME Extended Address Register.

It is important to note that there is just one VME User-Defined Address Modifier Register and VME User-Defined Extended Address Register whose values are used for all VME Windows configured for user-defined VME addressing. Multiple VME Windows configured for user-defined addressing with different address modifiers and/or different high-order address bits must be managed carefully by the host to ensure the proper values are contained in the VME User-Defined Registers before accessing the VME Window.

4.5.3 Interrupt Servicing

There are seven possible VMEbus interrupts and one possible error interrupt which must be translated to four or less ISA bus interrupts, IRQ10, IRQ11, IRQ12, or IRQ15. The following steps should be taken to set up interrupts passing from the VMEbus to the ISA bus.

NOTE

ONE OR MORE OF THESE ISA BUS INTERRUPTS MAY NOT BE AVAILABLE ON A SPECIFIC ISA MACHINE DUE TO ITS USE BY OTHER ISA HARDWARE. UNLIKE VMEbus INTERRUPTS, ISA INTERRUPTS MAY NOT BE SHARED. IT IS THE USER'S RESPONSIBILITY TO DETERMINE AVAILABILITY.

a. If interrupts are not already disabled (interrupts are automatically disabled after power-up or reset), disable them by setting the Interrupt Control Bit in the four IRQ Control Registers.
b. Assign any VME interrupts and the error interrupt to the desired ISA bus interrupt(s) by setting the appropriate bit pairs in the two Interrupt Assignment Registers.
c. Set up the interrupt manager on the VMEbus interrupt source. The method will vary according to the hardware used, but will at least involve prioritizing the interrupt and assigning an interrupt vector.
d. Enable ISA bus interrupts by setting the least significant bit in each IRQ Control Register for each ISA IRQ line used.
e. Enable VMEbus interrupts by clearing the Interrupt Control bit in the Adapter Board Control and Status Register.

Once interrupts are enabled, an interrupt may occur from any of the seven possible VMEbus interrupts or the VMIATX-5521 error interrupt. While the actual interrupt service routine function is up to the programmer, the following steps should always be taken in the interrupt service routine:

a. Read and store the values in a Window Register and Data Control Register pair.
b. Write $02 to the Window Register and $00 to the Data Control Register. This will make the VME Adapter Board registers available in the associated register with no byte swapping.
c. If multiple interrupt sources are assigned the same ISA interrupt, determine the source of the interrupt.
d. If the interrupt was VMEbus-triggered (i.e., it was not the error interrupt), read the corresponding VME Interrupt Acknowledge Register to perform a VMEbus interrupt acknowledge cycle and obtain the interrupt vector.
e. Perform whatever actions are necessary to service the interrupt.
f. If the interrupt was the error interrupt, reset the Error Register by performing a write to it.
g. Restore the Window Register and Data Control Register values saved previously.
h. Re-enable the interrupt by setting the Interrupt Control Bit of the Interrupt Control Register corresponding to the ISA interrupt that occurred.

If the number of possible interrupt sources is limited and enough of the four ISA bus interrupts are unused, it may be feasible to assign one ISA interrupt to each of the interrupt sources. If so, an interrupt source will be known as soon as the interrupt occurs and the software can respond accordingly without otherwise having to discover the interrupt source. A more typical situation, however, involves assigning several possible interrupt sources to just a few ISA interrupt lines. A worst-case situation exists

wherein all eight possible interrupt sources are assigned a single ISA IRQ line. In this or any similar case, the following procedure may be used to determine the source of the interrupt.

a. Read the Error Register. Any flags set therein indicate an error interrupt occurred.
b. Read the VME Interrupt Status Register. Any flags set correspond to pending VMEbus interrupts of the associated level.

SECTION 5

CONFIGURATION AND INSTALLATION

5.1 UNPACKING PROCEDURES

```
*****************
*     CAUTION     *
*****************
```

SOME OF THE COMPONENTS ASSEMBLED ON VMIC's PRODUCTS MAY BE SENSITIVE TO ELECTROSTATIC DISCHARGE AND DAMAGE MAY OCCUR ON BOARDS THAT ARE SUBJECTED TO A HIGH ENERGY ELECTROSTATIC FIELD. WHEN THE BOARD IS TO BE LAID ON A BENCH FOR CONFIGURING, ETC., IT IS SUGGESTED THAT CONDUCTIVE MATERIAL BE INSERTED UNDER THE BOARD TO PROVIDE A CONDUCTIVE SHUNT. UNUSED BOARDS SHOULD BE STORED IN THE SAME PROTECTIVE BOXES IN WHICH THEY WERE SHIPPED.

Upon receipt, any precautions found in the shipping container should be observed. All items should be carefully unpacked and thoroughly inspected for damage that might have occurred during shipment. The board(s) should be checked for broken components, damaged printed circuit board(s), heat damage, and other visible contamination. All claims arising from shipping damage should be filed with the carrier and a complete report sent to VMIC together with a request for advice concerning the disposition of the damaged item(s).

5.2 PHYSICAL INSTALLATION

```
*****************
*     CAUTION     *
*****************
```

DO NOT INSTALL OR REMOVE BOARDS WHILE POWER IS APPLIED.

De-energize the equipment and insert the boards into an appropriate slot of the chassis. While ensuring that the boards are properly aligned and oriented in the supporting card guides, slide the boards smoothly forward against the mating connector until firmly seated.

5.3 VME ADAPTER BOARD JUMPER CONFIGURATIONS

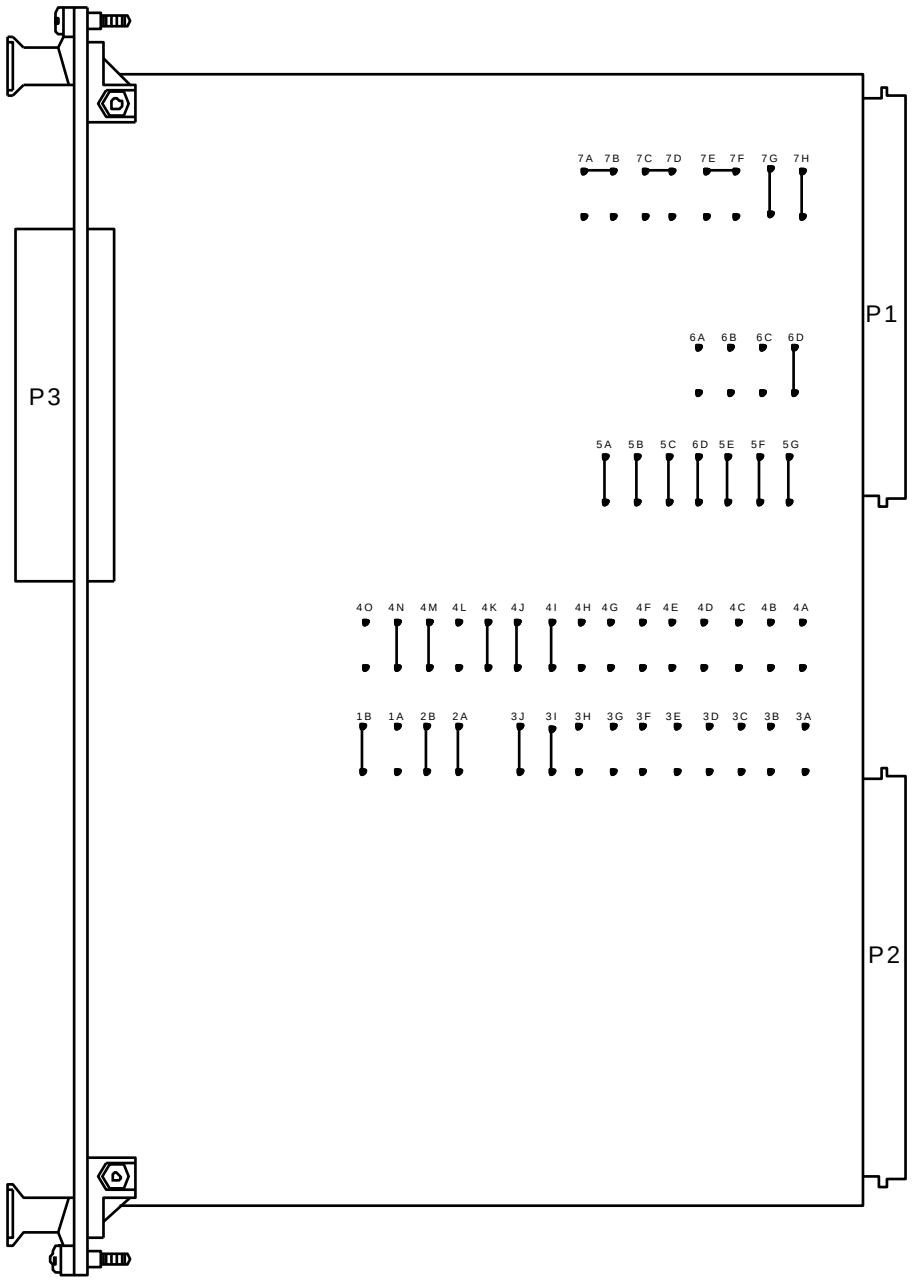
The VME Adapter Board has several jumper fields that are preconfigured for certain defaults at the factory but which may need to be changed according to the host system configuration.

NOTE

MANY JUMPER FIELDS CONTROL POWER-UP/RESET DEFAULT REGISTER VALUES ONLY. IN ALL CASES, THESE DEFAULT REGISTER VALUES MAY BE SUBSEQUENTLY OVERWRITTEN BY SOFTWARE.

5.3.1 VME Adapter Board Jumper Locations

Figure 5.3.1-1. shows the locations of all jumpers on the VME Adapter Board.



factory default jumpers shown M5521ATX/F5.3.1-1

Figure 5.3.1-1. VME Adapter Board Jumper Locations

5.3.2 Slot 1 Select (Jumper 1B)

If the slot 1 select jumper (jumper 1B) is installed, the VME Adapter provides the following system controller functions in the VMEbus chassis:

a. 16 MHz system clock.
b. Interrupt acknowledge (IACK*/IACKIN*) daisy chain driver.
c. VMEbus system reset (SYSRESET*) at power-up and under program control.

This jumper should be installed if the VME Adapter is in the slot 1 position in the VMEbus chassis and there are no other VMEbus masters present in this chassis. If other VMEbus masters are present, the VME Adapter must not be placed in slot 1.

5.3.3 A25/A28 Mode (Jumpers 1A, 2A, 2B, 3I, 3J, 4I)

Jumper 1A **must** be removed for proper VMIATX-5521 operation. When removed (indicating A28 mode), jumpers 2A, 2B, 3I, 3J, and 4I have no meaning, whether jumpered or not. Jumpers are generally factory installed in these locations (except 1A), although they may be removed.

5.3.4 Extended Address Register Default (Jumpers 3A through 3H)

Jumpers 3A through 3H provide the power-up/reset default VME Extended Address Register bits. Jumper 3A corresponds to VME Extended Address Register bit 7 and jumper 3H corresponds to register bit 0. If a jumper is installed, then the corresponding register bit is a logic “zero”. If the jumper is not installed, then the register bit is a logic “one”.

5.3.5 User-Defined Extended Address Register Default (Jumpers 4A through 4H)

Jumpers 4A through 4H provide the power-up/reset default VME User-Defined Extended Address Register bits. Jumper 4A corresponds to VME User-Defined Extended Address Register bit 7 and jumper 4H corresponds to bit 0. If a jumper is installed, then the corresponding register bit is a logic “zero”. If the jumper is not installed, then the register bit is a logic “one”.

5.3.6 User-Defined Address Modifier Register Default (Jumpers 4J through 4O)

Jumpers 4J through 4O provide the power-up/reset default VME User-Defined Address Modifier Register bits. Jumper 4J corresponds to VME User-Defined Address Modifier Register bit 5 and jumper 4O corresponds to bit 0. If a jumper is installed, then the corresponding register bit is a logic “zero” If the jumper is not installed, then the register bit is a logic “one”.

5.3.7 VMEbus Interrupt Request Pass-Through (Jumpers 5A through 5G)

Jumpers 5A through 5G control which VMEbus interrupts may be passed through the link to cause host interrupts. Jumper 5A controls VMEbus interrupt IRQ1 while jumper 5G controls VMEbus interrupt IRQ7. If a jumper is installed, the corresponding interrupt is passed through the link. If a jumper is removed, the corresponding VMEbus interrupt is permanently disabled: it cannot interrupt the host.

5.3.8 VME Bus Request and Grant Level (Jumpers 6A through 6D and 7A through 7H)

Jumpers 6A through 6D in conjunction with jumpers 7A through 7H determine the VMEbus request and grant level of the VME Adapter Board, and thus the link itself. The jumpers should be set according to Figures 5.3.8-1 through 5.3.8-4 depending upon the desired VMEbus request/grant level.

Figure 5.3.8-1. VMEbus Request/Grant Level 0 Jumpers

M5521ATX/F5.3.8-1

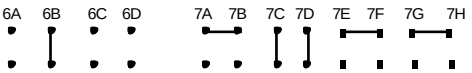


Figure 5.3.8-2. VMEbus Request/Grant Level 1 Jumpers

M5521ATX/F5.3.8-2

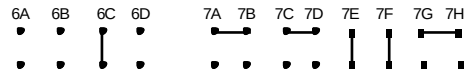
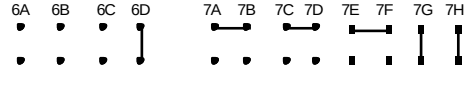


Figure 5.3.8-3. VMEbus Request/Grant Level 2 Jumpers

M5521ATX/F5.3.8-3



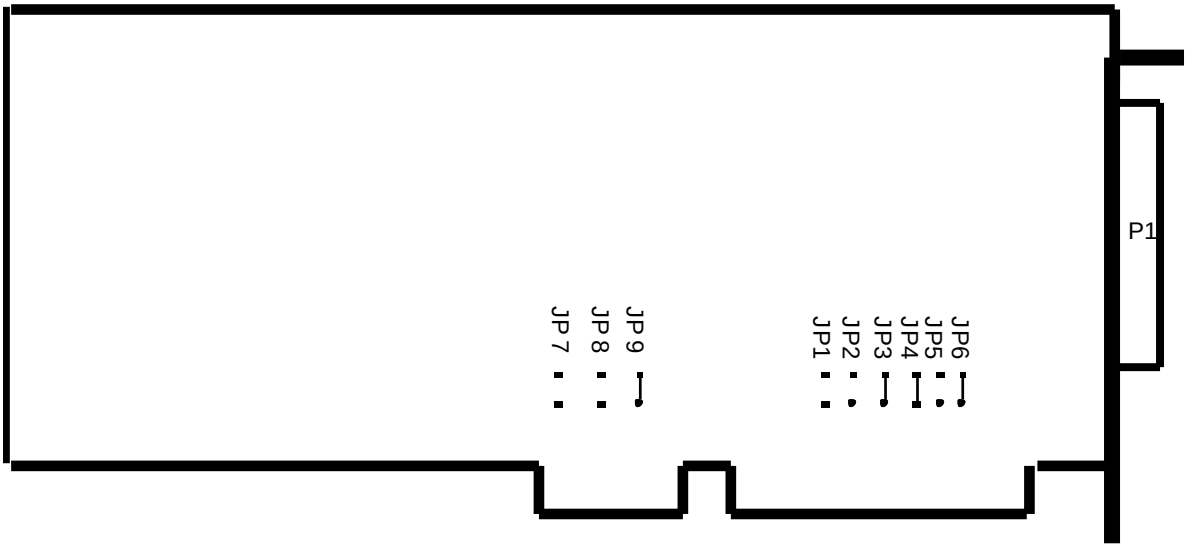
This is the factory default setting.

Figure 5.3.8-4. VMEbus Request/Grant Level 3 Jumpers

M5521ATX/F5.3.8-4

5.4 HOST ADAPTER BOARD JUMPER CONFIGURATIONS

The Host Adapter Board contains only two jumper fields. Jumpers JP1 through JP6 determine the ISA I/O base address for the Host Adapter registers and jumpers JP7 through JP9 determine the base memory address of the VME Windows. The jumper locations are illustrated in Figure 5.4-1.



factory default jumpers shown

M5521ATX/F5.4-1

Figure 5.4-1. Host Adapter Board Jumper Locations

5.4.1 Host Adapter Base I/O Address (Jumpers JP1 through JP6)

Jumpers JP1 through JP6 determine the base ISA I/O address for the Host Adapter registers. The Host Adapter registers occupy 16 bytes anywhere within the 1 K of available ISA I/O space (i.e., $000-$3F0 on even 16-byte boundaries). The jumpers essentially control the upper six bits of the 10-bit I/O address: JP1 controls the most significant address bit A9 and JP6 controls address bit A4 (the least significant four bits A3 through A0 are considered to be zero for base address determination). An installed jumper indicates a logic “zero” while an uninstalled jumper indicates a logic “one”. Table 5.4.1-1 illustrates the default jumper setting and its relationship to the base I/O address of the registers.

Table 5.4.1-1. Default Host Adapter I/O Base Address

I/O Address Bits and Jumper Setting	Resulting Base I/O

A9 JP1	A8 JP2	A7 JP3	A6 JP4	A5 JP5	A4 JP6	A3	A2	A1	A0	Address in Hexadecimal
1	1	0	0	1	0	0	0	0	0	$320 (default)

M5521ATX/T5.4.1-1

0 = jumper installed, 1 = no jumper installed

The default base I/O address of $320 should work well in most systems but should be changed if addresses if the range $320-$32F conflict with other I/O addresses used in the system. Note that base I/O addresses below $100 are reserved for system functions on most ISA bus systems and should never be used.

5.4.2 Host Adapter Base Memory Address (Jumpers JP7 through JP9)

Jumpers JP7 through JP9 determine the base ISA memory address for the VME Windows. The four VME Windows occupy a total of 2 Mbytes within the 24-bit (16 Mbyte) ISA addressing space and must lie on a natural 2 Mbyte boundary. The three jumpers essentially control the three most significant bits of the 16-bit ISA memory address, with an installed jumper indicating a logic “zero” and an uninstalled jumper indicating a logic “one”. Jumper JP7 controls A23, JP8 controls A22, and JP9 controls A21, and all lower address bits should be considered zero for base address calculation.

The default base memory address is $C00000, so jumpers JP7 and JP8 are left open and jumper JP9 is installed at the factory. Note that a base address of $000000 is possible but should be avoided in order to prevent system conflicts. A base address of $E00000 should also be avoided for similar reasons.

SECTION 6

MAINTENANCE AND WARRANTY

6.1 MAINTENANCE

This section of the technical manual provides information relative to the care and maintenance of VMIC's products. Should the products malfunction, the user should verify the following:

a. Software
b. System configuration
c. Electrical connections
d. Jumper or configuration options
e. Boards fully inserted into their proper connector location
f. Connector pins are clean and free from contamination
g. No components of adjacent boards are disturbed when inserting or removing the board from the VMEbus card cage
h. Quality of cables and I/O connections

User level repairs are not recommended. Contact VMIC for a Return Material Authorization (RMA) Number. This RMA Number must be obtained prior to any return.

6.2 MAINTENANCE PRINTS

The appendix(ices) to this manual contain(s) drawings and diagrams for reference purposes.

6.3 WARRANTY

VMIC's Standard Products are warranted to be free from defects in material and workmanship for a period of two years (24 months) from the date of shipment. In discharge of this warranty, VMIC, at its option, agrees to either repair or replace, at VMIC's facility and at VMIC's discretion, any part, component, subassembly accessory, or any hardware, software, or system product, which under proper and normal use proves defective in material and workmanship.

The customer shall provide notice to VMIC of each such defect within a reasonable time after the customer's discovery of such defect.

In order to return the defective product(s) or part(s), the customer must contact VMIC's Customer Service Department to obtain a Call Ticket Number. The defective product(s) or part(s) must also be properly boxed and weighed. After a VMIC Call Ticket Number and RMA Number have been obtained, the defective product(s) or

part(s) may be returned (transportation collect for surface UPS) to VMIC. Any replaced or repaired product(s) or part(s) will be shipped back to the customer's at the expense of VMIC (also UPS surface).

The customer should be aware that the above process can sometimes take up to eight (8) days for the shipment to reach VMIC. The customer has the option to ship the defective product(s) or part(s) at the customer's own expense if the customer cannot afford this possible delay.

There shall be no warranty or liability on any VMIC product(s) or part(s) that is (are) damaged or subjected to accident(s), perils of nature, negligence, overtemperature, overvoltage, misapplication of electrical power, insertion or removal of boards from backplanes and/or I/O connectors with power applied by the customer(s), appointee(s), or any other person(s) without the expressed approval of VMIC.

Final determination of warranty eligibility shall be made by VMIC, and if a warranty claim is considered invalid for any reason, the customer will be charged for services performed and expenses incurred by VMIC in repair, handling and shipping the returned product or part. Determination as to whether the item is within warranty, coverage shall not be unreasonably withheld.

The warranty period of the replacement or repaired product(s) or part(s) shall terminate with the termination of the warranty period with respect to the original product(s) or part(s) for all replacement parts supplied or repairs made during the original warranty period.

THE FOREGOING WARRANTY AND REMEDY ARE EXCLUSIVE AND VMIC SHALL HAVE NO OTHER OR ADDITIONAL LIABILITY TO BUYER OR TO ANYONE CLAIMING UNDER BUYER (THIRD PARTY) UNDER ANY OTHER AGREEMENT OR WARRANTY, EXPRESS OR IMPLIED EITHER IN FACT OR BY OPERATION OF THE LAW, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS, STATUTORY, OR OTHERWISE. VMIC SHALL HAVE NO LIABILITY FOR SPECIAL OR CONSEQUENTIAL DAMAGES OF ANY KIND OR FROM ANY CAUSE ARISING OUT OF THE INSTALLATION OR USE OF ANY PRODUCT FURNISHED HEREUNDER.

6.4 OUT-OF-WARRANTY REPAIR POLICY

The following sections describe VMIC's policy on repairs and warranties on repaired products.

6.4.1 Repair Category

VMIC's repair policy of standard products is divided into two categories, depending on the item to be repaired. These categories are:

a. Product Exchange
b. Fixed Price Repair

Category 1 (product exchange) represents the fastest turn around of the two categories. In this case, the customer sends the malfunctioning product to VMIC. VMIC will return an operational product to the customer within 72 hours of receipt provided VMIC has the product in stock.

Provided that the returned product is repairable customers should contact VMIC prior to returning products for repair to determine stocking status.

Category 2 (Fixed Price Repair) applies to products returned to VMIC for repair and subsequent return to the customer.

Return authorizations are required on all product repairs, and all purchase orders should refer to VMIC's RMA Number which is assigned by VMIC's Customer Service Department.

6.4.2 Repair Pricing

Contact your factory representative for repair pricing. Current pricing can be found in the Repair and Replacement Policy in the most current Standard Conditions of Sales Document (F0109-91). Refer to exclusions (Section 6.4.7).

6.4.3 Payment

Payment is due upon delivery or at VMIC's option, net thirty (30) days from the date of delivery. Payment should be made to:

VME Microsystems International Corporation
12090 South Memorial Parkway
Huntsville, Alabama 35803-3308
Attention: Accounts Receivable

VMIC allows a one (1) percent discount for payment made within ten (10) days of invoice date or a two (2) percent discount on payment made prior to shipment of order. This payment discount, however, does not apply to freight.

6.4.4 Shipping Charges

Shipping charges are the customer's responsibility, with the exception of warranty repairs, whereby VMIC will pay the return to customer shipping charges.

6.4.5 Shipping Instructions

The type of packaging used to ship the product depends on whether the product is shipped singly, in a chassis, or packaged with other boards. The shipper should carefully pack the product(s), using the same precautions listed in the "unpacking procedures". The user should utilize the same (or equivalent) protective packaging container for re-shipment as provided by VMIC. Approved ESD procedures are recommended when handling VMIC's products.

6.4.6 Warranty on Repairs

Products repaired by VMIC are warranted against defects in workmanship and material for a period of ninety (90) days from date of shipment to the customer for all products that were repaired out of warranty. See Standard Conditions of Sale for products repaired within the warranty.

6.4.7 Exclusions

Repair rates may not apply to products which have received unusual physical or electrical damage. In such cases, VMIC will provide an estimated price for product repair or replacement. The customer may then choose to have the product repaired at the estimated price, returned unrepaired at no charge, or replaced at VMIC's current list price.

APPENDIX A

ASSEMBLY DRAWING, PARTS LIST, AND SCHEMATIC