VMIPCI-5576

REFLECTIVE MEMORY BOARD

PRODUCT MANUAL

DOCUMENT NO. 500-855576-000 A

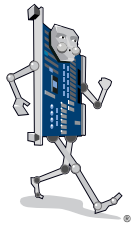
November 21, 1996

VME MICROSYSTEMS INTERNATIONAL CORPORATION
12090 SOUTH MEMORIAL PARKWAY
HUNTSVILLE, AL 35803-3308
(205) 880-0444 Fax No. (205) 882-0859
(800) 322-3616

COPYRIGHT AND TRADEMARKS

UIOC®

WinUIOC®



(I/O Man Figure)

(VMIC LOGO)

VME Microsystems International Corporation

RECORD OF REVISIONS

REVISION LETTER	DATE	PAGES INVOLVED	CHANGE NUMBER
A	11/21/96	Release	96-0801

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 A R N I N G

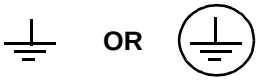
DANGEROUS VOLTAGES, CAPABLE OF CAUSING DEATH, ARE PRESENT IN THIS SYSTEM. USE EXTREME CAUTION WHEN HANDLING, TESTING, AND ADJUSTING.

SAFETY SYMBOLS

GENERAL DEFINITIONS OF SAFETY SYMBOLS USED IN THIS MANUAL

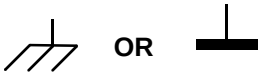
Instruction manual symbol: the product is marked with this symbol when it is necessary for the user to refer to the instruction manual in order to protect against damage to the system.

Indicates dangerous voltage (terminals fed from the interior by voltage exceeding 1000 volts are so marked).



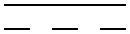
Protective conductor terminal. For protection against electrical shock in case of a fault. Used with field wiring terminals to indicate the terminal which must be connected to ground before operating equipment.

Low-noise or noiseless, clean ground (earth) terminal. Used for a signal common, as well as providing protection against electrical shock in case of a fault. Before operating the equipment, terminal marked with this symbol must be connected to ground in the manner described in the installation (operation) manual.

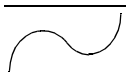


Frame or chassis terminal. A connection to the frame (chassis) of the equipment which normally includes all exposed metal structures.

Alternating current (power line).



Direct current (power line).



Alternating or direct current (power line).

WARNING

The WARNING sign denotes a hazard. It calls attention to a procedure, a practice, a condition, or the like, which, if not correctly performed or adhered to, could result in injury or death to personnel.

CAUTION

The CAUTION sign denotes a hazard. It calls attention to an operating procedure, a practice, a condition, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the system.

NOTE:

The NOTE sign denotes important information. It calls attention to a procedure, a practice, a condition or the like, which is essential to highlight.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Overview

Features

The VMIPCI-5576 is a high-performance, daisy-chained PCI-to-reflective memory network interface. Data is transferred to each node on the reflective memory network by writing to on-board global RAM. The data is automatically sent to the corresponding location in memory on all reflective memory boards on the network. The reflective memory network can include PCI bus, VMEbus, and Multibus I systems.

The VMIPCI-5576 reflective memory board has several useful and unique features:

- High-speed, easy-to-use fiber-optic network (170 Mbaud serially)
- Data written to memory in one node is also written to memory in all nodes on the network
- Up to 2,000 meters between nodes (maximum 256 nodes)
- Data transfer rate of 6.2 Mbyte / sec (without redundant transfer)
- Data transfer rate of 3.2 Mbyte / sec (redundant transfer)
- Any node on the network can generate an interrupt in any other node on the network or in all network nodes with a single command
- Error detection
- Redundant transmission mode for suppressing errors
- PCI-initiated DMA capability
- No processor involvement in the operation of the network, no processor overhead
- Up to 1 Mbyte of reflective memory
- D32, D16, and D8 memory access

- Single PCI 5 V slot, long card
- Configurable endian conversions for multiple CPU architectures on network
- Communication link compatible with VMIVME-5576 and VMIMB1-5576
- Software addressable digital output bit available at output connector for use with optical switch board or any user-defined purpose

Functional Description

Figure 1-1 on page 4 shows the block diagram of the reflective memory board. Note that the FIFO memory is on the same bus as the SRAM memory. The user sees only the SRAM memory and is not aware of the FIFO memory, which does the actual bus transfer to the other boards. The reflective memory board appears to the user as standard SRAM memory and can be used as such. The only effect noticeable to the user due to the presence of the communications bus is that the SRAM takes slightly longer to acknowledge when the receive FIFO is writing to RAM.

Software Requirements

The reflective memory board upon power up is able to establish the board-to-board link without any software program setup. The byte register which controls the interrupt generation on the reflective memory board is initialized to mask all interrupts upon power up.

If interrupts are desired, the appropriate byte register must be initialized through software control. The Control and Status Register (CSR2) controls the Fail LED on the reflective memory board. The user must write to the CSR2 after any test software is run successfully on the board in order to turn OFF the Fail LED. There is no automatic diagnostic software on the board to perform a self-test. The LED being ON does not indicate a failure on-board unless the PCI bus software has turned it ON. LED ON is a standard power-up mode for the Fail LED.

Hardware Requirements

A few jumpering requirements must be followed. The first requirement is that each reflective memory board on the communications bus must have a unique node ID address (jumper-selectable on-board). No two nodes can share the same node number (0,1...255). Nodes may be physically located in any order. Each reflective memory board may be mapped into a different address space. For the VMIPCI-5576s, data appears in the same location in each node relative to the base 1 Mbyte boundary to which each reflective memory is mapped.

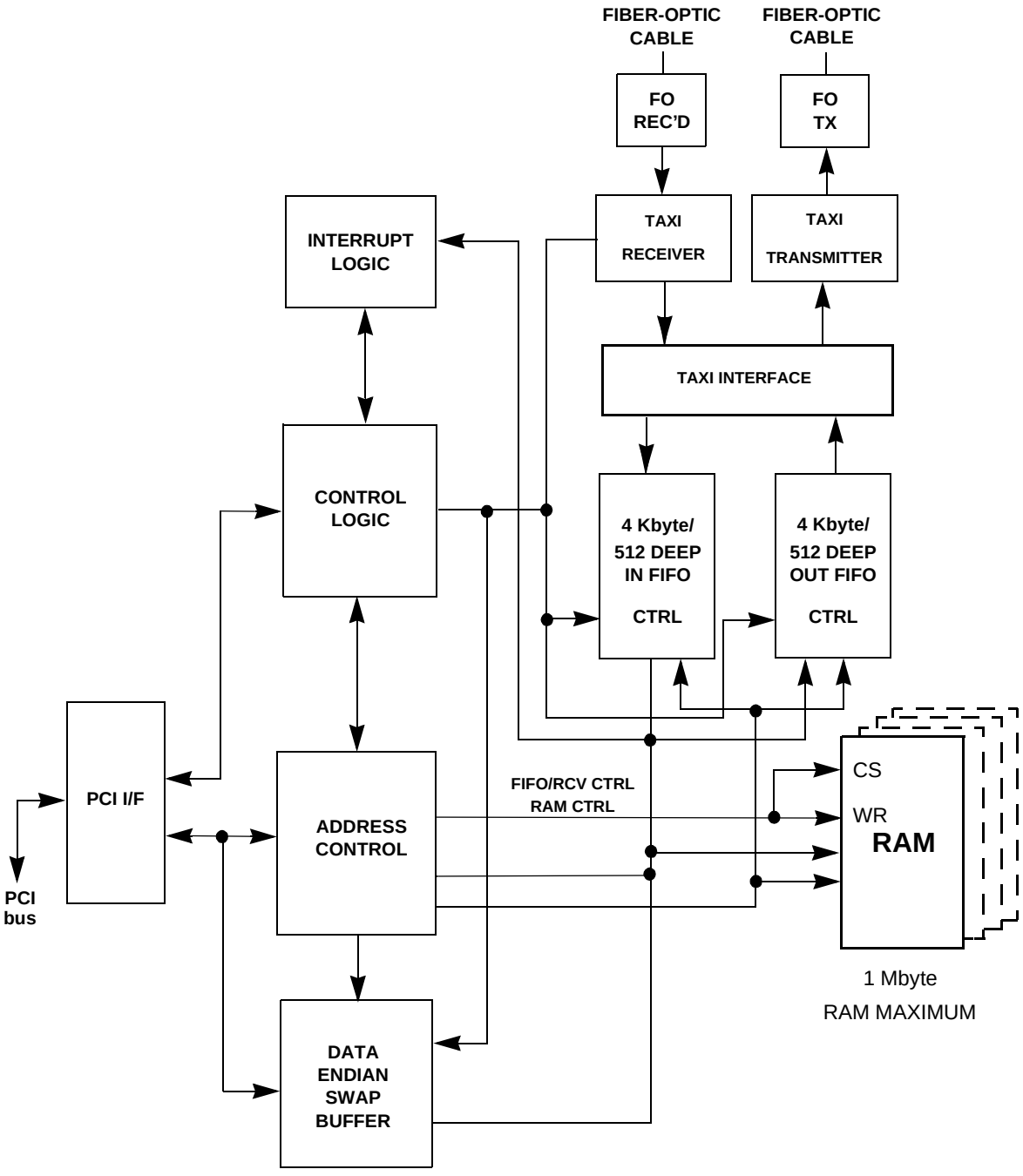


Figure 1-1. VMIPCI-5576 Functional Block Diagram

Reference Material List

Refer to "PCI Local Bus Specification" for a detailed explanation of the PCI Local bus. The PCI Local bus Specification is available from the following source:

PCI Special Interest Group

P.O. Box 14070

Portland, OR 97214

U.S.: (800) 433-5177

International: (503) 797-4207

FAX: (503) 234-6762

Refer to "The VMEbus Specification" for a detailed explanation of the VMEbus. "The VMEbus Specification" is available from the following source:

VITA

VMEbus International Trade Association

7825 East Gelding Dr. Suite 104

Scottsdale, AZ 85260

(602) 951-8866

FAX: (602) 951-0720

Email: info@vita.com

Physical Description and Specifications

Refer to 800-855576-000 specification, available from VMIC.

Safety Symbols

The **WARNING** sign denotes a hazard. It calls attention to a procedure, a practice, a condition, or the like, which, if not correctly performed or adhered to, could result in injury or death to personnel.

The **CAUTION** sign denotes a hazard. It calls attention to an operating procedure, a practice, a condition, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the system.

NOTE: 

The **NOTE** sign denotes important information. It calls attention to a procedure, a practice, a condition or the like, which is essential to highlight.

CHAPTER

1

Theory of Operation

Contents

Introduction

The VMIPCI-5576 is a high-performance, daisy-chained PCI-to-reflective memory network board. Data is transferred to each node on the network by writing to on-board global RAM. Data written to any node appears in all other nodes without programming or other intervention by the user. The link between the nodes is two fiber-optic cables which pass node ID, data, and if enabled, interrupt information between adjacent boards on the link until the ring is complete.

The VMIPCI-5576 allows a user to read or write the RAM address space at will. All memory writes are stored in SRAM on the board and also put in a FIFO to be broadcast to all other nodes. An interrupt command may also be written to any or all nodes by writing a data word into a specific location. The data value written dictates which node(s) will receive the interrupt. The interrupt is sent out in the order the data was received from the PCI bus. If a block of data was written to the board before the interrupt command is sent, then the data is broadcast to all boards before the interrupt command is broadcast.

Data not generated by the local node is placed in the receive FIFO. The receive FIFO then places data into RAM and into the transmit FIFO to be sent to the next node on the link. Since a node can receive data from both the local PCI bus and the link, data transfers from the PCI bus may be inserted among those from the link.

If both the local PCI bus and the receive FIFO access the RAM and transmit FIFO simultaneously, priority is given to the receive FIFO. In any other case, the first one to access the RAM and transmit FIFO receives priority.

Base Address Selection

The base address of the VMIPCI-5576 is automatically set by the BIOS. This is done through the use of PCI bus Configuration registers which are discussed in detail in the following sections.

Network Configuration

The VMIPCI-5576 system is a fiber-optic daisy-chain ring as shown in Figure 1-1 on page 1-5. Each transfer is passed from node-to-node until it has gone all the way around the ring and reaches the originating node. Each node retransmits all transfers that it receives except those that it originated. Nodes are allowed to insert transfers between transfers passing through.

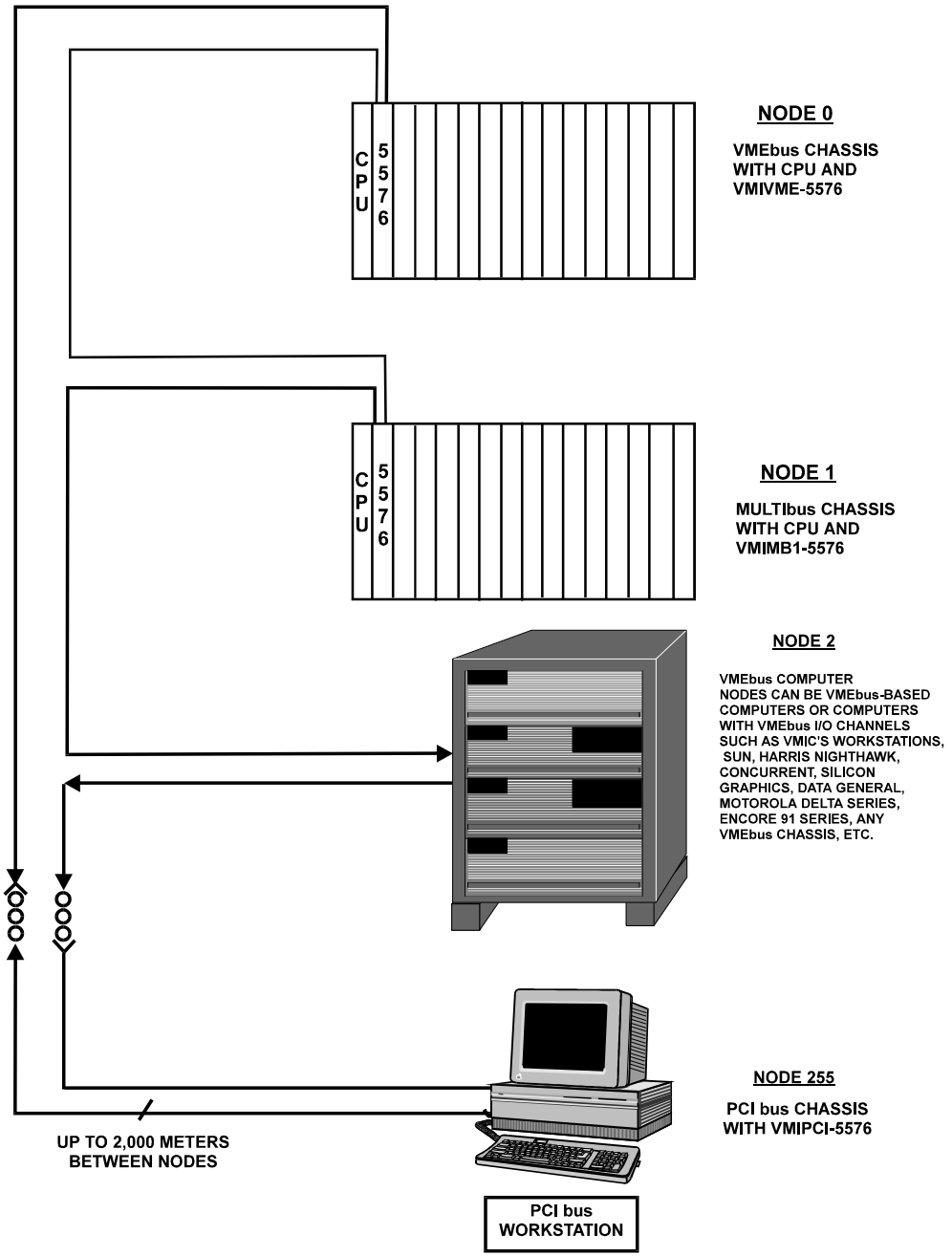


Figure 1-1. Network Example Using Reflective Memory System

Interrupt Transfers

In addition to transferring data between nodes, the VMIPCI-5576 will allow any processor in any node to generate an interrupt on any other node. These interrupts would generally be used to indicate to the receiving node that new data has been sent and is ready for processing. These interrupts are also used to indicate that processing of old data is completed and the receiving node is ready for new data.

Three interrupts are available. The user can define the function for each interrupt. In addition, any processor on the network can generate an interrupt on all nodes on the network. Interrupts are generated by simply writing to a single VMIPCI-5576 register. All data and interrupt command transfers contain the node number of the node that originated the transfer. This information is used primarily so the originating node can remove the transfer from the network after the transfer has traversed the ring. The node identification is also used by nodes receiving interrupt commands. When a node receives an interrupt command for itself, it places the identification number of the originating node in a FIFO. Up to 512 interrupts can be stacked in the FIFO. During the interrupt service routine, the identification of the interrupting node can be read from the FIFO.

Addressing Features

Not all nodes have to be configured with the same memory size. Nodes may be configured to make optimum use of memory. There are restrictions to different memory options on the same fiber-optic network due to the relative address which is passed around the network. The 256 Kbyte, 512 Kbyte, and 1 Mbyte memory options can reside on the same network.

The ring of VMIPCI-5576 or compatible products pass a relative address around the fiber-optic network, this creates the restriction that 256 Kbyte, 512 Kbyte, and 1 Mbyte boards pass a 1 Mbyte relative address. The PCI address configuration registers always ask the host to allocate the memory for a full network system. This ensures that the smaller memory boards are placed on the bottom of the ring's relative address boundary (for example, a 256 Kbyte board requests 1 Mbyte of PCI memory). There is a provision in the board control registers to add an offset to boards that do not have all the memory available. Control and Status Register 2 (CSR2) has offset bits which can be used to map smaller address blocks into different address segments in memory (for example, a 256 Kbyte board which would automatically be put at the bottom of the 1 Mbyte address range could be mapped between 256 and 511 Kbyte, 512 to 767 Kbyte, or 768 Kbyte to 1 Mbyte). The other memory options have this same feature within the restrictions of the network. Detailed example of the address offset features are presented in Section 3 on page 3-10, "CSR2 description."

Node Latency

If the fiber-optic data bandwidth has not been exceeded, data latency is typically 1.5 µsec/node in single transfer mode. Longer latencies result if data input rates exceed 6.2 Mbyte/sec sustained. The PCI bus can burst data into the transmit FIFO and RAM at 44 Mbyte/sec until the TX FIFO becomes full at which time the PCI bus interface would generate a STOP so that data is not lost in the TX FIFO.

Configuration Space

This portion of the theory of operation defines the programming model and usage rules for the configuration register space in PCI-compliant devices. The intent of the PCI configuration space definition is to provide an appropriate set of configuration "hooks" which satisfies the needs of current and anticipated system configuration mechanisms.

Configuration Space Organization

This section defines the organization of configuration space registers and imposes a specific record structure or template on the 256-byte space. This space is divided into a predefined header region and a device dependent region. Devices implement only the necessary and relevant registers in each region. A device's configuration space must be accessible at all times, not just during system boot. The predefined header region has a size of 64 bytes and every device must support the register layout of this region. This region consists of fields that uniquely identify the device and allow the device to be generically controlled. See Figure 1-2 for the Configuration Space Header on page 1-10.

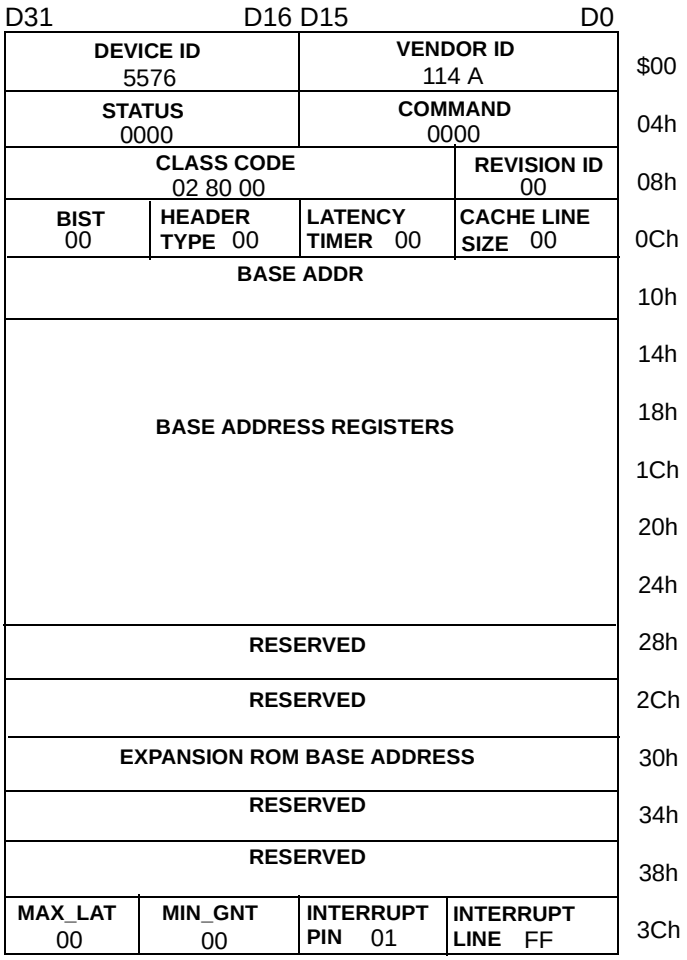


Figure 1-1. Configuration Space Header

All compliant devices must support the Vendor ID, Device ID, Command and Status fields in the header. Implementation of other registers is optional (i.e., they can be treated as reserved registers) depending on the device functionality. If a device supports the function that the register is concerned with, the device must implement it in the defined location and with the defined functionality.

Configuration Space Functions

PCI has the potential for greatly increasing the ease in which systems may be configured. To realize this potential, all PCI devices must provide certain functions that system configuration software can utilize. This section also lists the functions that need to be supported by PCI devices by way of registers defined in the predefined header portion of the configuration space. The exact format of these registers (i.e., number of bits implemented) is device-specific. However, some general rules must be followed. All registers must be capable of being read back and the data returned must indicate the value that the device is actually using.

Configuration space is intended for configuration, initialization, and catastrophic error handling functions. Its use should be initialization software and error handling software. All operational software must continue to use I/O and/or memory space accesses to manipulate device registers.

Device Identification

Five fields in the predefined header deal with device identification. All PCI devices are required to implement these fields. Generic configuration software will be able to easily determine what devices are available on the system's PCI bus(ses). All of these registers are read-only.

Vendor ID: This field identifies the manufacturer of the device. Valid vendor identifiers are allocated by the PCI SIG to ensure uniqueness. $114A is VMIC's vendor ID.

Device ID: This field identifies the particular device. This identifier is allocated by the vendor. The Device ID for the VMIPCI-5576 is 5576.

Revision ID: This register specifies a device-specific revision identifier. The value is chosen by the vendor. Zero is an acceptable value. This field should be viewed as a vendor-defined extension to the Device ID.

Header Type: This byte identifies the layout of bytes $10 through $3F in configuration space and also whether or not the device contains multiple functions. Bit 7 in this register is used to identify a multi-function device. If the bit is 0, then the device has multiple functions. Bits 6 through 0 specify the layout of bytes $10 through $3F. One encoding, $00, is defined and specifies the layout shown in Figure 1-2 on page 1-10. All other encodings are reserved.

Class Code: The Class Code register is read-only and is used to identify the generic function of the device. The register is broken into three byte-size fields. The upper byte (at offset $0B) is a base class code which broadly classifies the type of function the device performs. The middle byte (at offset $0A) is a subclass code which identifies more specifically the function of the device. The lower byte (at offset $09) identifies a specific register-level programming interface (if any) so that device independent software can interact with the device. The VMIPCI-5576 class code is $28000.

Device Control

The Command register provides course control over a device's ability to generate and respond to PCI cycles. When a zero (0) is written to this register, the device is logically disconnected from the PCI bus for all accesses except configuration accesses. All devices are required to support this base level of functionality. Individual bits in the Command register may or may not be implemented depending on a device's functionality. For instance, devices that do not implement an I/O space probably will not implement a writable element at bit location zero (0) of the Command register. Devices typically come up with all zeros in this register. Figure 1-3 shows the layout of the register and Table 1-1 on page 1-14 explains the definitions of the different bits in the Command register.

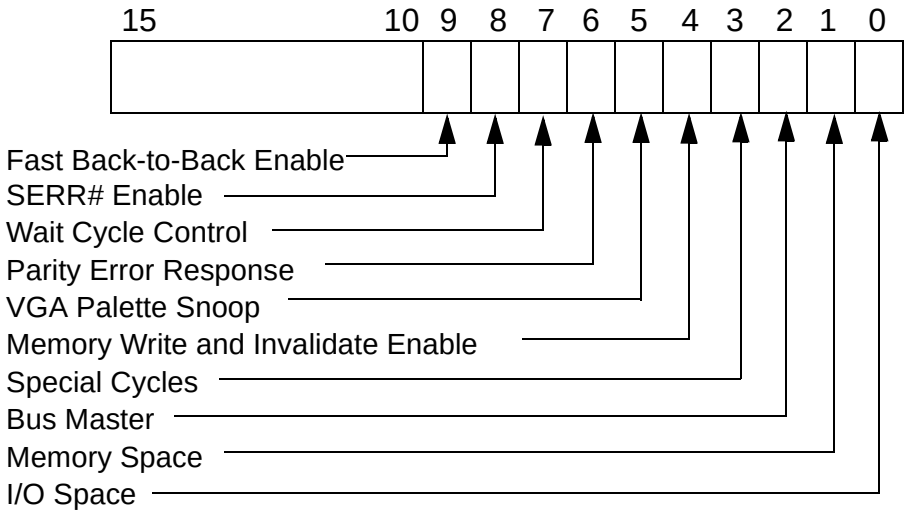


Figure 1-1. Command Register Layout

Table 1: Command Register Bits

Bit Location	Bit Definitions
0	Controls a device's response to I/O space accesses. A value of zero disables the device response. A value of one allows the device to respond to I/O space accesses.
1	Controls a device's response to memory space accesses. A value of zero disables the device response. A value of one allows the device to respond to memory space accesses.
2	Controls a device's abilitiy to act as a master on the PCI bus. A value of zero disables the device from generating PCI accesses. A value of one allows the device to behave as a bus master.
3	Controls a device's action on Special Cycle Operations. A value of zero causes the device to ignore all Special Cycle Operations. A value of one allows the device to monitor Special Cycle Operations.
4	This is an enable bit for using the Memory Write and invalidate command. When this bit is one, masters can generate the command. When this bit is zero, Memory Write must be used instead. State after RST# is zero. This bit must be implemented by master devices that can generate the Memory Write and Invalidate command.
5	This bit controls how VGA-compatible devices handle accesses to their palette registers. When this bit is set, special palette snooping behavior is enabled (i.e., device must not respond). When the bit is reset, the device should treat palette accesses like all other accesses. VGA-compatible devices should implement this bit.
6	This bit controls the device's response to parity errors. When the bit is set, the device must take its normal action when a parity error is detected. When the bit is reset, the device must ignore any parity errors that it detects and continue normal operation. This bit must be set to zero after RST#. Devices that check parity must implement this bit. Devices are still required to generate parity even if parity checking is disabled.
7	This bit is used to control whether or not a device does address/data stepping. Devices that never do stepping must hardwire this bit to zero. Devices that always do stepping must hardwire this bit to one. Devices that can do either, should make this bit read/write and have it initialize to one after RST#.
8	This bit is an enable bit for the SERR# driver. A value of zero disables the SERR# driver. A value of one enables the SERR# driver. This bit's state after reset is zero. All devices that have an SERR# pin must implement this bit. This bit (and bit 6) must be ON to report address parity errors.
9	This optional read/write bit controls whether or not a master can do fast back-to-back transactions to different devices. Intialization software will set the bit if all targets are fast back-to-back capable. A value of one means the master is allowed to generate fast back-to-back transactions to different agents. A value of zero means fast back-to-back transactions are only allowed to the same agent.
10 through 15	Reserved

Device Status

The Status register is used to record status information for PCI bus-related events. The definition of each of the bits is given in Table 1-2 on page 1-16, and the layout of the register is shown in Figure 1-4 below. Devices would not need to implement all bits, depending on device functionality. For instance, a device that acts as a target but will never signal target-abort, would not implement bit 11.

Reads to this register behave normally. Writes are slightly different in that bits can be reset, but not set. A bit is reset whenever the register is written, and the data in the corresponding bit location is a one (1). For instance, to clear bit 14 and not affect any other bits, write the value 0100_0000_0000_0000b to the register.

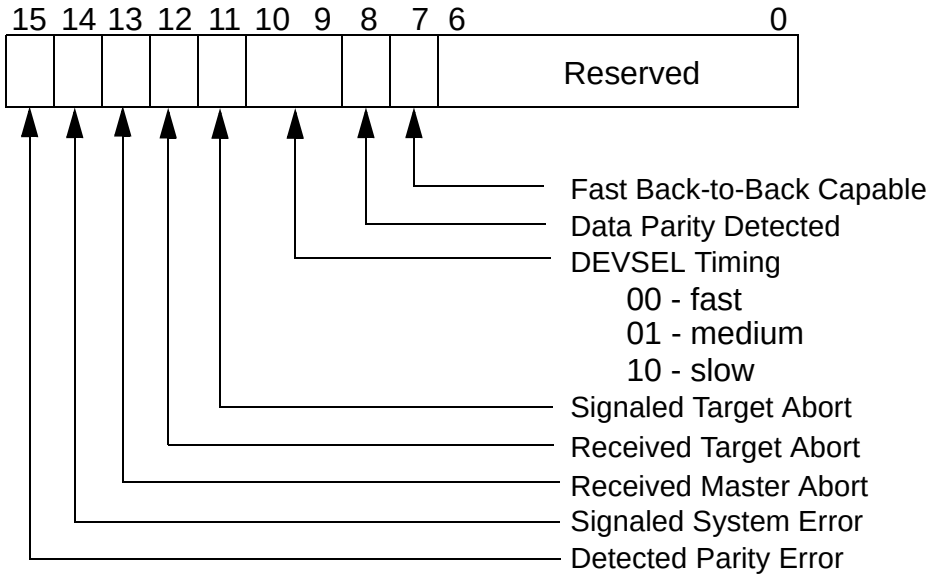


Figure 1-1. Status Register Layout

Table 2: Status Register Bits

Bit Location	Descriptions
0 through 6	Reserved
7	This optional read-only bit indicates whether or not the target is capable of accepting fast back-to-back transactions when the transactions are not to the same agent. This bit can be set to one (1) if the device can accept these transactions, and must be set to zero (0) otherwise.
8	This bit is only implemented by bus masters. It is set when three conditions are met: 1) the bus agent asserted PERR# itself or observed PERR# asserted; 2) the agent setting the bit acted as the bus master for the operation in which the error occurred; and 3) the Parity Error Response bit (Command register) is set.
9 through 10	These bits encode the timing of DEVSEL#. The bits are encoded as 00b for fast, 01b for medium, and 10b for slow (11b is reserved). These bits are read-only and must indicate the slowest time that a device asserts DEVSEL# for any bus command except Configuration Read and Configuration Write.
11	This bit must be set by a target device whenever it terminates a transaction with target-abort. All master devices must implement this bit.
12	This bit must be set by a master device whenever its transaction is terminated with target-abort. All master devices must implement this bit.
13	This bit must be set by a master device whenever its transaction (except for Special Cycle) is terminated with master-abort. All master devices must implement this bit.
14	This bit must be set whenever the device asserts SERR#. Devices which will never assert SERR# do not need to implement this bit.
15	This bit must be set by the device whenever it detects a parity error, even if parity error handling is disabled as controlled by bit 6 in the Command register.

Miscellaneous Functions

This section describes the registers that are device independent and only need to be implemented by devices that provide the described function.

Interrupt Line

The Interrupt Line register is an 8-bit register used to communicate interrupt line routing information. The register is read/write and must be implemented by any device (or device function) that uses an interrupt pin. POST software will write the routing information into this register as it initializes and configures the system.

The value in this register tells which input of the system interrupt controller(s) the device's interrupt pin is connected to. Device drivers and operating systems can use this information to determine priority and vector information. Values in this register are system architecture specific.[1]

NOTE: 

[1] FOR x86 BASED PCs, THE VALUES IN THIS REGISTER CORRESPONDS TO IRQ NUMBERS (0 THROUGH 15) OF THE STANDARD DUAL 8259 CONFIGURATION. THE VALUE 255 IS DEFINED AS MEANING "UNKNOWN" OR "NO CONNECTION" TO THE INTERRUPT CONTROLLER. VALUES BETWEEN 15 AND 255 ARE RESERVED.

Interrupt Pin

The Interrupt Pin register tells which interrupt pin the device (or device function) uses. A value of one (1) corresponds to INTA#. A value of two (2) corresponds to INTB#. A value of three (3) corresponds to INTC#. A value of four (4) corresponds to INTD#. Devices (or device functions) that do not use an interrupt pin must put a zero (0) in this register. This register is read-only.

Latency Timer Register (LAT)

The latency timer register is an 8-bit wide register located in the PCI configuration space at offset $0D. Bits 7 through 3 are read and write accessible while bits 2, 1, and 0 are read-only. This register initializes to 00.

The latency timer register has meaning only when the VMIPCI-5576 PCI I/F is used as a bus master (DMA enabled) and pertains to the number of PCI bus clocks that this master will be guaranteed. The nonzero value is internally decremented after this device has been granted the bus and has asserted FRAME#. Prior to this latency timer count reaching zero, this device can ignore the removal of the bus grant and may continue the use of the bus for data transfers. This PCI configuration register is used to ensure that no resource dominates the PCI bus and that bus arbitration occurs on a regular basis. The VMIPCI-5576 will request the PCI bus when the DMA process is enabled and the VMIPCI-5576 is ready to move data. The complete transfer block size may be broken into smaller burst automatically by the PCI I/F since the VMIPCI-5576 must release the bus at regular intervals. The maximum value the user can write to this register is F8. The user only has to write this value one time, then the value is used by the counter on all master cycles.

Base Addresses

One of the most important functions for enabling superior configurability and ease-of-use is the ability to relocate PCI devices in the address spaces. At system powerup, device independent software must be able to determine what devices are present, build a consistent address map, and determine if a device has an expansion ROM. Each of these areas are covered in the following sections.

Address Maps

Power up software needs to build a consistent address map before booting the machine to an operating system. This means it has to determine how much memory is in the system, and how much address space the I/O controllers in the system require. After determining this information, power up software can map the I/O controllers into reasonable locations and proceed with system boot. In order to do this mapping in a device independent manner, the base registers for this mapping are placed in the predefined header portion of configuration space.

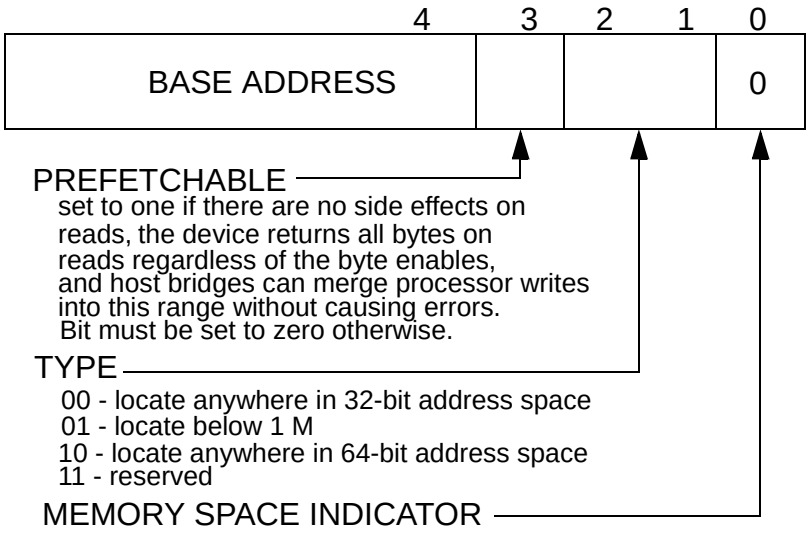


Figure 1-1. Base Address Register for Memory

Bit 0 in all base registers is read-only and used to determine whether the register maps into Memory I/O space. Base registers that map to Memory space must return a zero (0) in bit 0. Base registers that map to I/O space must return a one (1) in bit 0.

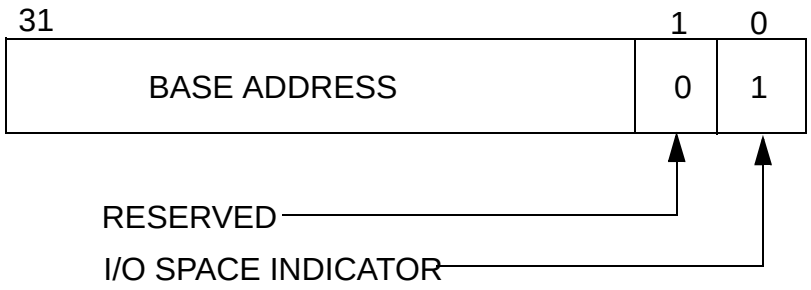


Figure 1-1. Base Address Register for I/O

Base registers that map into I/O space are always 32 bits with bit 0 hardwired to a one (1), bit 1 is reserved and must return zero (0) on reads, and the other bits are used to map the device into I/O space[1].

Base registers that map into memory space (Figure 1-6 can be 32 bits wide to support mapping into a 64-bit address space) with bit 0 hardwired to a zero (0). For memory base registers, bits 2 and 1 have an encoded meaning as shown in Table 1-3 below. Bit 3 should be set to one (1) if the data is prefetchable, and reset to zero (0) otherwise. A device can mark a range as prefetchable[2] if there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables, and host bridges can merge processor writes into this range[2] without causing errors. Bits 0 through 3 are read-only.

Table 3: Bits 2/1 Encoding

Bits 2/1	Definition
00	Base register is 32 bits wide and mapping can be done anywhere in the 32-bit memory space.
01	Base register is 32 bits wide but must be mapped below 1 Mbyte in memory space.
10	Base register is 64 bits wide and can be mapped anywhere in the 64-bit address space.
11	Reserved

The number of upper bits that a device actually implements depends on how much of the address space the device will respond to. A device that wants a 1 Mbyte memory address space (using a 32-bit base address register) would build the top 12 bits of the address register hardwiring the other bits to zero (0).

Power up software can determine how much address space the device requires by writing a value of all ones to the register and then reading the value back. The device will return zeros in all don't care address bits, effectively specifying the address space required.

This design implies that all address spaces used are a power of two in size, and are naturally aligned. Devices are free to consume more address space than required, but decoding down to a 4 Kbyte space for memory and 256 bytes for I/O is suggested for devices that need less than those amounts. Devices that do consume more address space than they use are not required to respond to the unused portion of that address space.

Six DWORD locations are allocated for Base Address registers starting at offset $10 in configuration space. The first Base Address register is always located at offset $10. The second register may be at offset $14 or $18 depending on the size of the first. The offsets of subsequent Base Address registers are determined by the size of previous Base Address registers.

A typical device will require one memory range for its control functions. Some graphics devices may use two ranges, one for control functions and another for a frame buffer. A device that wants to map control functions into both memory and I/O space at the same time must implement two base registers (one Memory, one I/O). The driver for that device might only use one space in which case the other space will be unused. Devices should always allow control functions to be mapped into memory space.

NOTE:

[1]THE I/O ADDRESS WILL HAVE 01 IN THE LEAST SIGNFICANT BITS, BUT THE BASE IS AT 00. IF, FOR EXAMPLE, THE CONFIGURATION PLACES THE I/O ADDRESS AT $FF00 IN THE I/O SPACE, THE BASE ADDRESS REGISTER WILL READ $FF01.

[2]ANY DEVICE THAT HAS A RANGE THAT BEHAVES LIKE NORMAL MEMORY, BUT DOESN'T PARTICIPATE IN PCI's CACHING PROTOCOL, SHOULD MARK THE RANGE AS PREFETCHABLE. A LINEAR FRAME BUFFER IN A GRAPHICS DEVICE IS AN EXAMPLE OF A RANGE THAT SHOULD BE MARKED PREFETCHABLE.

CHAPTER

2

Configuration and Installation

Contents

Introduction

This chapter is used to set the configuration of the board, depending on your particular need. Jumper configuration, fiber-optic link configuration, and board layout are also addressed in this chapter.

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 PLACED ON A BENCH FOR CONFIGURING, ETC., IT IS SUGGESTED THAT CONDUCTIVE MATERIAL SHOULD 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).

Physical Installation

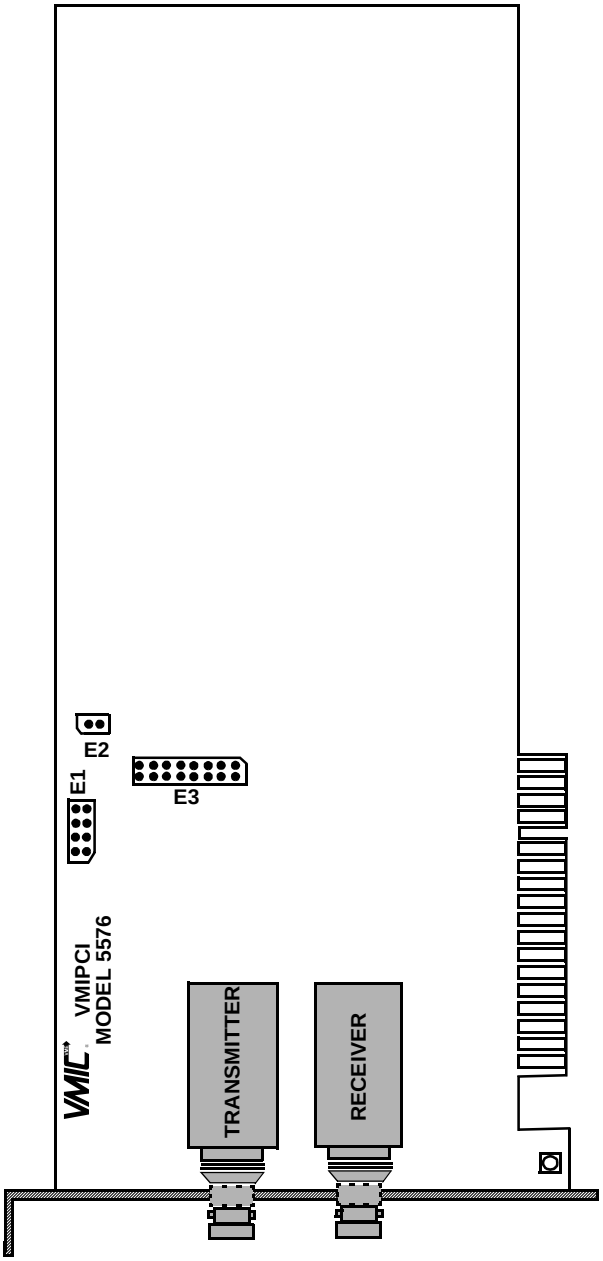
CAUTION

DO NOT INSTALL OR REMOVE THE BOARD WHILE POWER IS APPLIED.

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

Jumper Installation

Figure 2-1 shows the layout of user-configurable jumpers on the reflective memory board.



E1 NODE BYPASS JUMPER	
Pin No.	**Function**
1	NODE BYPASS
2	FUSED 5 V
3	N/C
4	GND
5	LOGIC LOW
6	FAIL LED (D2)
7	LOGIC LOW
8	+5 V (NOT FUSED)
DEFAULT = NO JUMPERS INSTALLED	

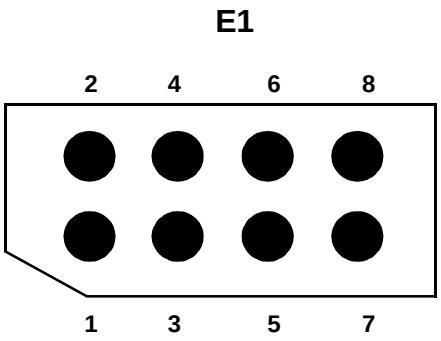
E3 NODE ID JUMPER	
Pin No.	**Function**
1 TO 2	LSB
15 TO 16	MSB
DEFAULT = ALL JUMPERS INSTALLED	

E2 FAST JUMPER
DEFAULT = ALL JUMPERS INSTALLED

Figure 2-1. VMIPCI-5576 Jumper Locations

Node Bypass Jumper (E1)

The Node Bypass jumper allows the user sofware control of a bit in the Control Registers to be used externally from the board. This bit can be used to enable a node bypass switch that causes the optical signal to bypass this board. Power and GND are also provided for use with an external switch. The 5 V line is fused with a 1 Amp fuse. The Node Bypass signal is driven with a 74FCT2244. See Figure 2-2 for the overview of the Node Bypass jumper. The LED signals are available also for use by an external board. Pin 8 is a nonfused 5 V signal that should only be used by VMIC-designed hardware.



PIN#	FUNCTION
1	NODE BYPASS
2	FUSED +5 V (1 AMP MAX)
3	N/C
4	GND
5	LOGIC LOW
6	FAIL LED (D2)
7	LOGIC LOW
8	+5 V (NOT FUSED)

Figure 2-2. Node Bypass Jumper E1

Board Node ID Field (E3)

The Board Node ID Field identifies each board on the memory link. No ID may be used more than once on the link. Jumper field E3 provides 8 bits to define the board node ID. An installed jumper sets the corresponding bit to zero. There is no relation between the node number and the physical position on the link. Nodes may be physically located in any order.

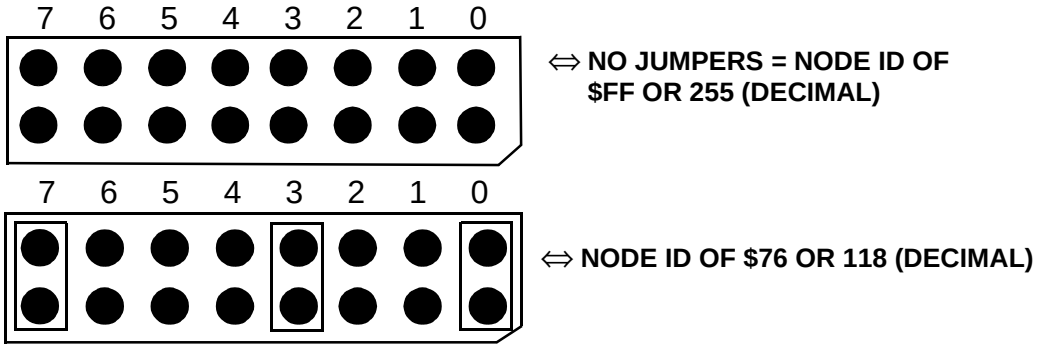


Figure 2-3. Example of Node ID Field Jumper E3

Fast Field Jumper (E2)

If the E2 jumper is omitted, the fiber-optic link rate is 6.2 Mbyte / sec. If the jumper is installed, the link rate is 3.2 Mbyte / sec. The 3.2 Mbyte / sec rate results from transmitting every data point twice. If an error is detected in the first transmission, it is thrown away and the second transmission of data is used. If the first transmission is okay, the second is ignored. The state of this jumper can be read in CSR1 register.

Fiber-Optic Link Configuration

A VMIPCI-5576 link is formed by connecting the transmitter of board A into the receiver of board B. Board B's transmitter is then connected to the receiver of board C and so on. The last board in the link has its transmitter connected back to board A's receiver to close the link.

When data has been sent around the link and has returned to the originating node, two things happen. First, the data is removed from the link by comparing the data packet's ID to the local node's ID. If they match, the data packet is removed from the link. Second, the OWN DATA bit is set in the CSR. Loop data latency can be measured by writing a zero (0) to the OWN DATA bit in the CSR and polling until it returns to a one (1) state. This test assumes there is no other data originated by the local node on the link before the latency test is initiated. The data write to the CSR is passed around the link as regular traffic but does not affect the status of any other node's CSR.

CHAPTER

3

PROGRAMMING

CONTENTS

NOTE: 

FOLLOWING ESTABLISHED CONVENTIONS, A GROUP OF DATA BITS 16 BITS WIDE IS SIMPLY CALLED A "WORD" WHILE 32-BIT WIDE DATA IS CALLED A "DOUBLE WORD OR DWORD."

Programming

The VMIPCI-5576 reflective memory board has three sets of registers associated with it. The first is the PCI Configuration registers that can only be accessed with configuration cycles over the PCI bus. The PCI Configuration registers are automatically configured by the PC BIOS on power up. These registers are defined by the PCI specifications and will be initialized automatically (by an on-board state machine, no software is required) during the power up sequence from an on-board EPROM. The address map of the PCI Configuration Space Header is shown below. Detailed bit information is in Section 1 under the title "Configuration Space" on page 1-9.

<table>
<tr><td colspan="2">31 24</td><td colspan="2">23 16</td><td colspan="2">15 8</td><td colspan="2">7 00</td></tr>
<tr><td colspan="4">DEVICE ID = 5576</td><td colspan="4">114A = VENDOR ID</td></tr>
<tr><td colspan="4">STATUS 00 80</td><td colspan="4">0000 COMMAND</td></tr>
<tr><td colspan="2">02 80</td><td colspan="4">CLASS CODE 00</td><td colspan="2">REV ID = 00</td></tr>
<tr><td>BIST = 00</td><td>HEADER TYPE = 00</td><td colspan="4">LATENCY TIMER=00</td><td colspan="2">CACHE SIZE = 00</td></tr>
<tr><td colspan="8">BASE ADDRESS REGISTER #1 (I/O - PCI I/F IC) 64 BYTES REQUESTED
BASE ADDRESS REGISTER #2 (Memory and Board Control) 1 MBYTES REQUESTED
BASE ADDRESS REGISTER #3
BASE ADDRESS REGISTER #4
BASE ADDRESS REGISTER #5
BASE ADDRESS REGISTER #6 } NOT USED</td></tr>
<tr><td colspan="8">RESERVED = 0</td></tr>
<tr><td colspan="8">RESERVED = 0</td></tr>
<tr><td colspan="8">EXPANSION ROM BASE ADDRESS</td></tr>
<tr><td colspan="8">RESERVED = 0</td></tr>
<tr><td colspan="8">RESERVED = 0</td></tr>
<tr><td colspan="2">MAX_LAT</td><td colspan="2">MIN_GNT</td><td colspan="2">INTERRUPT PIN</td><td colspan="2">INTERRUPT LINE</td></tr>
</table>

Figure 3-1. PCI Configuration Space Header

The second set of Control Registers are those associated with the PCI Interface Chip which is used on the board. This set of Control Registers is only used to enable and configure the PCI interrupts. This address area is mapped as 16 consecutive DWORD registers located at the address space (I/O) specified by the Base Address register 1 (as shown in Figure 3-1.) The register map for this address space is shown in Table 3-1 on page 3-3. These registers are accessed with PCI I/O cycles. The third set of registers are those associated with the internal board operations and are located in the lower $40 bytes of memory. These registers must be accessed with PCI memory cycles.

PCI bus Operation Registers

The PCI bus operation registers are mapped as 16 consecutive DWORD registers located at the address space (I/O) specified by the Base Address Register 0. Table 3-1 lists the PCI bus Operation Registers.

Table 3-1: PCI bus Operation Register Map

Offset	Descriptions	Mnemonic	Access Mode
$00	I/O Mailbox	OMB1	Lword (R/W)
$04	I/O Mailbox	OMB2	Lword (R/W)
$08	I/O Mailbox	OMB3	Lword (R/W)
$0C	I/O Mailbox	OMB4	Lword (R/W)
$10	Incoming Mailbox	IMB1	Lword (Read Only)
$24	Master Write Address Register	MWAR	Lword (R/W)
$28	Master Write Transfer Count Register	MWTC	Lword (R/W)
$2C	Master Read Address Register	MRAR	Lword (R/W)
$30	Master Read Transfer Count Register	MRTC	Lword (R/W)
$38	Interrupt Control and Status	INTCSR	Lword (R/W)
$3C	Bus Master Control/Status	MCSR	Lword (R/W)

The PCI I/F IC has other functions that are not utilized in this implementation. Reserved bits should be zero (0) when written to and ignored when read.

User I/O Outgoing Mailboxes (OMB1, OMB2, OMB3, OMB4)

32-bit, R/W, located at I/O offset $0, $4, $8, and $C.

There are four user mailboxes mapped at the lower 4 longwords of I/O space on the board. These mailboxes are 32-bit R/W locations that are accessible to the user for any purpose. The access times for these registers are faster than a single memory-mapped access. These registers are not reflected across the network.

Incoming Mailbox (IMB1)

32-bit, Read only, located at I/O offset $10.

IMB1 will never contain useful information, but due to the architecture of the PCI I/F and the way the PCI interrupts are generated this register must be read during some of the interrupt service routines. When to read this register is defined in the section , titled "Using Interrupts", on page 3-31.

Master Write Address Register (MWAR)

This register is used to establish the PCI address for data moving from the VMIPCI-5576 to the PCI bus during DMA write operations. It consists of a 30-bit counter with the low-order two bits hardwired as zeros. (See Figure 3-2, below.)

The Master Write Address Register is continually updated during the transfer process and will always be pointing to the next unwritten location. Reading of this register during a transfer process is permitted and can be used to monitor the progress of the transfer. During the address phase for bus master write transfers, the two least significant bits presented on the PCI bus pins AD[31:0] will always be zero. This identifies to the target memory that the burst address sequence will be in a linear order rather than in an Intel 486 or Pentium cache line fill sequence. Also, the PCI bus address bit A1 will always be zero when this controller is the bus master. This signifies to the target that the S593X controller is burst capable and that the target should not arbitrarily disconnect after the first data phase of this operation.

Register Name: Master Write Address

Address Offset 24h

Power up Value: 00000000h

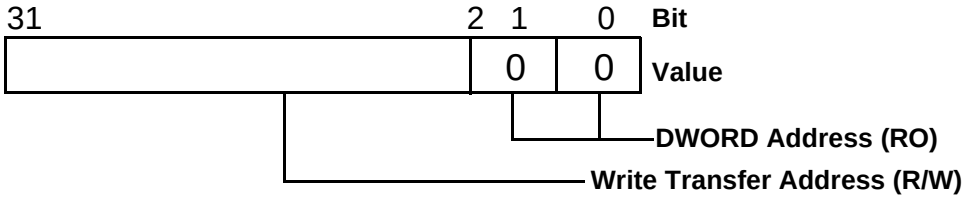


Figure 3-2. Master Write Address Register

Master Write Transfer Count Register (MWTC)

The Master Write Transfer Count Register is used to convey to the PCI DMA controller the actual number of bytes that are to be transferred. The value in this register is decremented with each bus master PCI write operation until the transfer count reaches zero. Upon reaching zero, the transfer operation ceases and an interrupt may be optionally generated. Figure 3-3 is an illustration of the MWTC register.

Register Name: Master Write Transfer Count

Address Offset 28h

Power up Value: 00000000h

Attribute: Read/Write

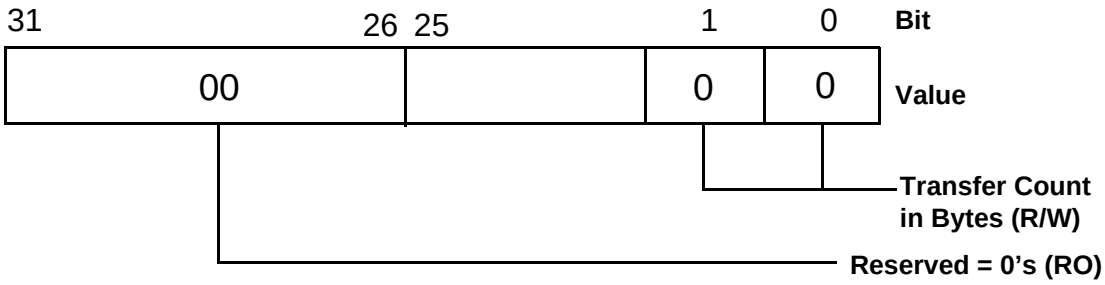


Figure 3-3. Master Write Transfer Count Register

Master Read Address Register (MRAR)

This register is used to establish the PCI address for data moving to the VMIPCI-5576 from the PCI bus during DMA read operations. It consists of a 30-bit counter with the low-order two bits hardwired as zeros.

The Master Read Address Register is continually updated during the transfer process and will always be pointing to the next unread location. Reading of this register during a transfer process is permitted and may be used to monitor the progress of the transfer. During the address phase for bus master read transfers, the two least significant bits presented on the PCI bus AD[31:0] will always be zero. This identifies to the target memory that the burst address sequence will be in a linear order rather than in an Intel 486 or Pentium cache line fill sequence. Also, the PCI bus address bit A1 will always be zero when this controller is the bus master. This signifies to the target that the controller is burst capable and that the target should not arbitrarily disconnect after the first data phase of this operation. Figure 3-4 is an illustration of this register.

Register Name: Master Read Address

Address Offset 2Ch

Power up Value: 00000000h

Attribute: Read/Write

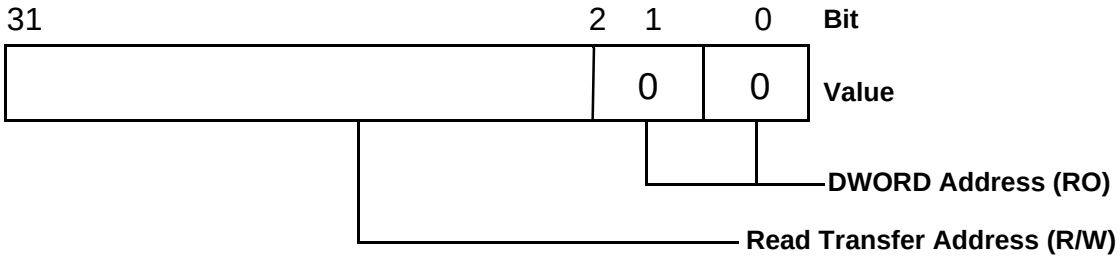


Figure 3-4. Master Read Address Register

Master Read Transfer Count Register (MRTC)

The Master Read Transfer Count Register is used to convey to the PCI DMA controller the actual number of bytes that are to be transferred. The value in this register is decremented with each bus master PCI read operation until the transfer count reaches zero. Upon reaching zero, the transfer operation ceases and an interrupt may be optionally generated. Figure 3-5 is an illustration of this register.

Register Name: Master Read Transfer Count

Address Offset 30h

Power up Value: 00000000h

Attribute: Read/Write

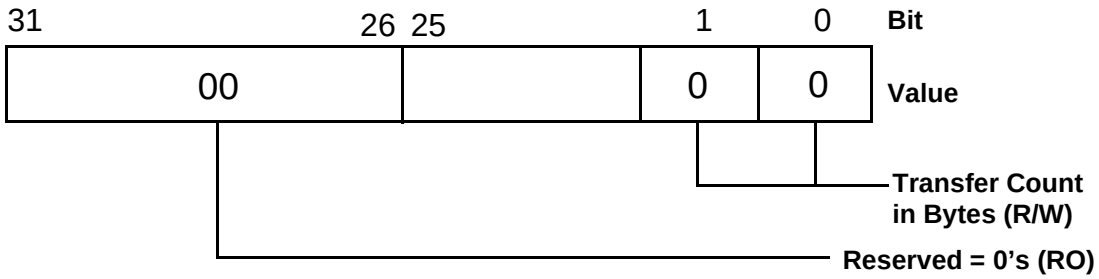


Figure 3-5. Master Read Transfer Count Register

Interrupt Control/Status Register (INTCSR)

This register provides the method for choosing which conditions are to produce an interrupt on the PCI bus interface, a method for viewing the cause of the interrupt, and a method for acknowledging (removing) the interrupt's assertion.

Table 3-2: Interrupt Control/Status Register Bit Map

INTCSR: RELATIVE OFFSET $38 READ/WRITE, LWORD							
Bit 31	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
0	0	0	0	0	0	0	0
Bit 23	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
INT Asserted	0	Target Abort	Master Abort	Read Transfer	Write Transfer	Incoming Mailbox INT	0
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 09**	**Bit 08**
Interrupt on Read COMP	Interrupt on Write COMP	0	Incoming Mailbox Enable	Reserved			
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
0	0	0	0	0	0	0	0

Interrupt Control/Status Register Bit Definitions

Bits 31 through 24: **Reserved** -- Always zero.

Bit 23: **Interrupt Asserted** -- This read-only status bit indicates that one or more of the four possible interrupt conditions is present. This bit is nothing more than the ORing of the interrupt conditions described by bits 19 through 16 of this register.

Bit 22: **Reserved** -- Always zero.

Bit 21: **Target Abort** -- This bit signifies that an interrupt has been generated due to the VMIPCI-5576 encountering a target abort during a PCI bus cycle while the VMIPCI-5576 was the current bus master. This bit operates as read or write one clear. A write to this bit with the data of "one" will cause this bit to be reset; a write to this bit with the data of "zero" will not change the state of this bit.

Bit 20: **Master Abort** -- This bit signifies that an interrupt has been generated due to the VMIPCI-5576 encountering a Master Abort on the PCI bus. A master abort occurs when there is no target response to a PCI bus cycle. This bit operates as read or write one clear. A write to this bit with the data of one (1) will cause this bit be reset; a write to this bit with the data of zero (0) will not change the state of this bit.

Bit 19: **Read Transfer Complete** -- This bit signifies that an interrupt has been generated due to the completion of a PCI bus master operation involving the transfer of data from the PCI bus to the add-on. This interrupt will occur when the Master Read Transfer Count register reaches zero. This bit operates as read or write one clear. A write to this bit with the data of one (1) will cause this bit to be reset; a write to this bit with the data of zero (0) will not change the state of this bit.

Bit 18: **Write Transfer Complete** -- This bit signifies that an interrupt has been generated due to the completion of a PCI bus master operation involving the transfer of data to the PCI bus from the add-on. This interrupt will occur when the Master Write Transfer Count register reaches zero. This bit operates as read or write one clear. A write to this bit with the data of one (1) will cause this bit to be reset; a write to this bit with the data of zero (0) will not change the state of this bit.

Bit 17: **Incoming Mailbox Interrupt** -- This bit is set when the mailbox selected by bits 12 through 8 of this register are written by the add-on interface. This bit operates as read or write one clear. A write to this bit with the data of one (1) will cause this bit to be reset; a write to this bit with the data as zero (0) will not change the state of this bit.

Bit 16: **Reserved** -- Write to zero.

Bit 15: **Interrupt on Read Transfer Complete** -- This bit enables the occurrence of an interrupt when the read transfer count reaches zero. This bit is read / write.

Bit 14: **Interrupt on Write Transfer Complete** -- This bit enables the occurrence of an interrupt when the write transfer count reaches zero. This bit is read / write.

Bit 13: **Reserved** -- Write to zero.

Bit 12: **Incoming Mailbox Enable** -- This bit allows the VMIPCI-5576 to produce a PCI interrupt when one is received over the reflective memory network. This bit is read / write.

Bits 11 Through 0: Reserved - Write to zero.

Bus Master Control/Status Register (MCSR)

This register provides for overall control of the VMIPCI-5576 PCI I / F. It is used to enable bus mastering for both data directions as well as providing a method to perform software resets. Table 3-3 on page 3-11 is the bit definitions for this register. The following PCI bus controls are available:

- Write Transfer Enable
- Write master requests on four or more DMA FIFO words available (full)
- Read transfer enable
- Read master requests on four or more DMA FIFO available (empty)
- Assert reset to VMIPCI-5576
- Reset VMIPCI-5576 to PCI DMA FIFO flags
- Reset PCI to VMIPCI-5576 DMA FIFO flags
- Reset mailbox empty full status flags

The following PCI interface status flags are provided:

- PCI to VMIPCI-5576 DMA FIFO FULL
- PCI to VMIPCI-5576 DMA FIFO has four or more empty locations
- PCI to VMIPCI-5576 DMA FIFO EMPTY
- VMIPCI-5576 to PCI DMA FIFO FULL
- VMIPCI-5576 to PCI DMA FIFO has four or more words loaded
- VMIPCI-5576 to PCI DMA FIFO EMPTY
- PCI to VMIPCI-5576 Terminal Count = Zero
- VMIPCI-5576 to PCI Terminal Count = Zero

Table 3-3: Bus Master Control/Status Register Bit Map

Relative Offset $3C MCSR Read/Write, Lword							
Bit 31	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
0				Reset Control			
Bit 23	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
Reserved 0000 0000							
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 08**
0	Read Transfer Enable	Read FIFO Management Scheme	0	0	Write Transfer Enable	Write FIFO Management Scheme	0
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
PCI-5576 to PCI Transfer Count	PCI to PCI-5576 Transfer Count	PCI-5576 to PCI FIFO Empty	PCI-5576 to PCI 4+ Space	PCI-5576 to PCI FIFO Full	PCI to PCI-5576 FIFO Empty	PCI to PCI-5576 FIFO 4+ Spaces	PCI to PCI-5576 FIFO Full

Bus Master Control/Status Register Bit Definitions

Bits 31 through 28: **Reserved**: Write to zero.

Bit 27: **Mailbox Flag (Reset Control):** Writing a one to this bit causes all mailbox status flags to become reset (EMPTY). It is not necessary to write this bit as zero because it is used internally to produce a reset pulse. Since reading of this bit will always produce zeros, this bit is write only.

Bit 26: **VMIPCI-5576 to PCI FIFO Status (Reset Control)** -- Writing a one to this bit causes the VMIPCI-5576 to PCI (Bus master memory writes) FIFO empty flag to set indicating empty and the FIFO FULL flag to reset and the FIFO Four Plus word flag to reset. It is not necessary to write this bit as zero because it is used internally to produce a reset pulse. Since reading of this bit will always produce zeros, this bit is write only.

Bit 25: **PCI to VMIPCI-5576 FIFO Status (Reset Control):** Writing a one to this bit causes the PCI to VMIPCI-5576 (Bus master memory reads) FIFO empty flag to set indicating empty and the FIFO FULL flag to reset and the FIFO Four Plus words available flag to set. It is not necessary to write this bit as zero because it is used internally to produce a reset pulse. Since reading of this bit will always produce zeros, this bit is write only.

Bit 24: **VMIPCI-5576 Pin (Reset Control)** -- Writing a one (1) to this bit resets the entire VMIPCI-5576 reflective memory board. All registers will be put in their default condition. Writing a zero (0) to this bit is necessary to remove the assertion of the reset. This register bit is read / write.

Bits 23 through 15: **Reserved:** Write to zero.

Bit 14: **Read Transfer Enable** -- This bit must be set to a one (1) for VMIPCI-5576 PCI bus master read transfers to take place. Writing a zero (0) to this location will suspend an active transfer. An active transfer is one in which the terminal count is not zero.

Bit 13: **Read FIFO Management Scheme** -- When set to a one (1), this bit causes the controller to refrain from requesting the PCI bus unless it has four or more vacant FIFO locations to fill. Once the controller is granted the PCI bus or is in possession of the bus due to the write channel, this constraint is not meaningful. When this bit is zero (0) the controller will request the PCI bus if it has at least one vacant FIFO word.

Bits 12 and 11: **Reserved** -- Write to zero.

Bit 10: **Write Transfer Enable** -- This bit must be set to a one (1) for PCI bus master write transfers to take place. Writing a zero (0) to this location will suspend an active transfer. An active transfer is one in which the terminal count is not zero.

Bit 9: **Write FIFO Management Scheme** -- When set to a one (1) this bit causes the controller to refrain from requesting the PCI bus unless it has four or more FIFO locations filled. Once the VMIPCI-5576 controller is granted the PCI bus or is in possession of the bus due to the write channel, this constraint is not meaningful. When this bit is zero (0), the controller will request the PCI bus if it has at least one valid FIFO word.

Bit 8: **Reserved** -- Write to zero.

Bit 7: **VMIPCI-5576 to PCI Transfer Count Equal Zero (RO)** -- This bit is a one (1) to signify that the write transfer count is all zeros.

Bit 6: **PCI to VMIPCI-5576 Transfer Count Equals Zero (RO)** -- This bit is a one (1) to signify that the read transfer count is all zeros.

Bit 5: **VMIPCI-5576 to PCI DMA FIFO Empty** -- This bit is a one (1) when the VMIPCI-5576 to PCI bus DMA FIFO is completely empty.

Bit 4: **VMIPCI-5576 to PCI 4+ Words** -- This bit is a one (1) when there are four or more FIFO words valid within the VMIPCI-5576 to PCI bus DMA FIFO.

Bit 3: **VMIPCI-5576 to PCI DMA FIFO Full** -- This bit is a one (1) when the VMIPCI-5576 to PCI bus DMA FIFO is completely full.

Bit 2: **PCI to VMIPCI-5576 DMA FIFO Empty** -- This bit is a one (1) when the PCI bus to VMIPCI-5576 DMA FIFO is completely empty.

Bit 1: **PCI to VMIPCI-5576 DMA FIFO 4+ Space** -- This bit signifies that there are at least four empty words within the PCI to VMIPCI-5576 DMA FIFO.

Bit 0: **PCI to VMIPCI-5576 DMA FIFO Full** -- This bit is a one (1) when the PCI bus to VMIPCI-5576 DMA FIFO is completely full.

VMIPCI-5576 Reflective Memory-Mapped Control Registers

The third set of control registers are those that have more direct control over the reflective memory board. Although the VMIPCI-5576 reflective memory board is software transparent on power up, some registers are present to facilitate user information and interrupt generation. Table 3-4 shows the memory map for the VMIPCI-5576.

Table 3-4: VMIPCI-5576 Register Map

Offset	Descriptions	Mnemonic	Access Mode
$1	Board ID	BID	Byte (R)
$4	Node ID	NID	Byte (R)
$8	INT and Receiver Status	IRS	Byte (R/W)
$9	Control and Status 1	CSR 1	Byte/Word (R/W)
$C	Control and Status 2	CSR 2	Byte/Word (R/W)
$D	Control and Status 3	CSR 3	Byte/Word (R/W)
$10	Command Node	CMDN	Byte/Word (R/W)
$11	Command	CMD	Byte/Word (W)
$14	INT Control and Status	ICSR	Byte/Word (R/W)
$18	INT1 Sender ID	SID1	Byte (R/W)*
$1C	INT2 Sender ID	SID2	Byte (R/W)*
$20	INT3 Sender ID	SID3	Byte (R/W)*
$24	DMA start ADDRESS	DADD	LWORD (W)
$40	Start of RAM		Byte, Word, Lword (R/W)
$03FFFF	End of 1/4 Mbyte RAM		Byte, Word, Lword (R/W)
$07FFFF	End of 1/2 Mbyte RAM		Byte, Word, Lword (R/W)
$0FFFFF	End of 1 Mbyte RAM		Byte, Word, Lword (R/W)

Board Identification (BID) Register

The Board Identification (BID) Register is read-only and will be assigned a fixed value. This will allow the user to verify the presence of the reflective memory board. This register may be considered redundant on the VMIPCI-5576 since a Device ID is present in the PCI configuration memory on this board. The board ID for the VMIPCI-5576 is $53.

Table 3-5: Board ID Register Bit Map

BID: Relative Offset $01, Read-Only, Byte							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
0	1	0	1	0	0	1	1

Node ID (NID) Register

The Node ID Register contains the 8-bit node ID (0 to 255) which is used over the network. This is set by the Node ID jumper (E3) on the board.

Table 3-6: Node ID Register Bit Map

NID: Relative Offset $04, Read-only, Byte							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
Node ID							

Interrupt and Receiver Status (IRS) Register

Table 3-7: Interrupt and Receiver Status Register Bit Map

IRS: RELATIVE OFFSET $08, READ/WRITE TO CLEAR BITS 3, 5, AND 6							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
Reserved		BAD DATA RECEIVED	RCVR FIFO HF LOW	Reserved	INT3 PENDING	INT2 PENDING	INT1 PENDING

Interrupt and Receiver Status Register Bit Definitions

Bits 07 and 06: **RESERVED** -- Write to zero.

Bit 05: **BAD DATA RECEIVED** -- A transfer error has occurred. This bit is cleared by writing a zero (0) to this register. 1 = bad data received.

Bit 04: **RCVR FIFO HALF-FULL LOW** -- The receiver FIFO is not half-full. Writes have no effect on this register (immediate status only). 1 = FIFO not more than half-full.

Bit 03: **RESERVED** -- Write to zero.

Bit 02: **INT3 PENDING** -- A one (1) in this bit location indicates that at least one level 3 interrupt has been received (Sender FIFO3 not empty). Writes have no effect on this register (Immediate status only).

Bit 01: **INT2 PENDING** -- A one (1) in this bit location indicates that at least one level 2 interrupt has been received (Sender FIFO2 not empty). Writes have no effect on this register (immediate status only).

Bit 00: **INT1 PENDING** -- A one (1) in this bit location indicates that at least one level 1 interrupt has been received (Sender FIFO1 not empty). Writes have no effect on this register (immediate status only).

Control and Status Register 1 (CSR1)

The CSR contains local node state information and is mapped as follows.

Table 3-8: Control and Status Register 1 Bit Map

CSR1: RELATIVE OFFSET $09 READ/WRITE, BYTE/WORD							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
Reserved	RCVR FIFO HALF FULL LOW	TX FIFO HALF FULL LOW	TX FIFO EMPTY LOW	DMA WRITE ENABLE	OWN DATA	DMA READ ENABLE	FAST

Control and Status Register 1 Bit Definitions

Bit 07: **RESERVED** -- Write to zero, when read, this bit should be a zero (0.)

Bit 06: **RCVR FIFO HALF-FULL, LOW** -- When this bit is one (1), the receive FIFO is less than half-full (Read-Only).

Bit 05: **TRANSMIT FIFO HALF-FULL, LOW** -- When this bit is one (1), the transmit FIFO on the local node is less than half-full (Read-Only).

Bit 04: **TRANSMIT FIFO EMPTY** -- When this bit is one (1), the transmit FIFO on the local node is not empty (Read-Only).

Bit 03: **DMA WRITE ENABLE** -- When this bit is set to a one (1), internal data transfers are allowed between memory and PCI DMA FIFO. This bit should not be set if bit 1 (Read DMA Enable bit) is set.

Bit 02: **OWN-DATA** -- When this bit is set to one (1), a data packet that this node sent has been received. Writing a zero (0) to this bit will clear it. The act of writing a zero (0) will generate a packet on the network. Therefore, it will be set back to one (1) as soon as the packet goes around the ring.

Bit 01: **DMA READ ENABLE** -- When this bit is set to a one (1), internal data transfers are allowed between the PCI DMA FIFO and memory. This bit should not be set if bit 3 (DMA Write Enable bit) is set.

Bit 00: **FAST MODE** -- When this bit is set to one (1), Fast mode is enabled. Fast mode transmits each data transfer once on the fiber-optic link, if the jumper E7 is omitted. This reads the status of the Fast jumper and is read-only.

1= Jumper omitted (Fast Mode)

0= Jumper Installed (Redundant Mode)

NOTE: 

BITS THAT ARE LABELED "SPARE" ARE BITS THAT CAN HAVE SOME FUTURE USE. BITS THAT ARE LABELED "RESERVED" ARE BITS THAT CANNOT BE USED AND MUST BE FORCED TO THE STATE INDICATED.

Control and Status Register 2 (CSR2)

CSR2 is mainly a board control register. The base address bits are utilized when the board is configured with less than the full amount of memory allowed on the network.

Table 3-9: Control and Status Register 2 Bit Map

CSR2: RELATIVE OFFSET $0C, READ/WRITE, BYTE/WORD							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
FAIL LED STATUS	NODE BYPASS	RESERVED				ADDRESS OFFSET OA19	ADDRESS OFFSET OA18

Control and Status Register 2 Bit Definitions

Bit 07: **FAIL LED STATUS** -- When this bit is set to one (1), the LED is OFF. The LED is ON after power up. The user must write a one (1) to this bit to turn the Fail LED OFF.

Bit 06: **NODE BYPASS** -- Writing a one (1) to this bit causes the node to be removed from the network (node is bypassed). This feature requires additional external hardware. This bit is buffered and made available to the user on jumper E1 for any purpose.

Bits 5 through 2: **RESERVED** -- Write to zero.

Bits 1 and 0: **ADDRESS OFFSET [D1=OA19, D0=OA18]** -- Bits 1 and 0 are used to add an offset value to the PCI address before it is put on the reflective memory network. This is used to put memory boards that have less memory than the network allows on any legal boundary (for example, a 256 K board can be put at 0, 256, 512, or 768 K boundaries). Both bits 1 and 0 are used to offset the 256 K boards, while bit 1 alone is used to offset 512 K boards. For 512 K boards, bit 0 should always be written to a zero.

The address offset bits default to zero (0) and all unused bits must be written as zeros when modifying other bits. Not all bits have an effect on all boards, Table 3-10 on page 3-21 shows which bits effect which boards.

Table 3-10: Address Offset Bits

256 Kbyte	Bits 1 and 0 (all other bits must be loaded with zero (0))
512 Kbyte	Bit 1 (all other bits must be loaded with zero (0))
1 Mbyte	No offset allowed

Example. A: A 256 Kbyte board will request 1 Mbyte of address space during the PCI bus initialization process. An address such as $FFC00000 will be assigned as the base address for the reflective memory during initialization. All the reflective memory boards on the network will pass the lower 20 address bits around the ring since it is considered a 1 Mbyte ring. The 256 Kbyte board can then be configured with CSR3 address offset registers in the conditions below. CSR3 bits 2, 3, 4, and 5 must be zero (0).

CSR3 (1)	CSR3 (0)	PCI Address	Network Address
0	0	$FFC00000 to $FFC3FFFF	$00000 to $3FFFF (*Default condition*)
0	1	$FFC00000 to $FFC3FFFF	$40000 to $7FFFF
1	0	$FFC00000 to $FFC3FFFF	$80000 to $BFFFF
1	1	$FFC00000 to $FFC3FFFF	$C0000 to $FFFFF

This would allow different boards the ability to write to a 1 Mbyte board in different sections of memory if desired.

The offset is similar for the 512 K cards.

Bits 4 and 2: **RESERVED**-- Not currently used, but must be zero (0).

Control and Status Register 3 (CSR3)

CSR3 is mainly a board control register.

Table 3-11: Control and Status Register 3 Bit Map

CSR3: RELATIVE OFFSET $0D, READ/WRITE, BYTE/WORD							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
USER FLAG A	CONFIG 2	CONFIG 1	CONFIG 0	USER FLAG B	Endian Control EC2	Endian Control EC1	Endian Control EC0

Control and Status Register 3 Bit Definitions

Bits 7 and 3: **USER FLAG[A..B]** -- These bits are read / write user-defined flags.

Bits 6, 5, and 4: **CONFIG[2..0]** -- These bits show the memory configuration of the board, see the table below for memory configuration.

000	256 Kbyte	101	reserved
001	512 Kbyte	110	reserved
010	1 Mbyte	111	No Memory
011	Reserved		
100	reserved		

Bits 2, 1, and 0: **Endian Control EC[2..0]** -- These bits are for big-endian / little-endian conversions between PCI and any VMEbus machines. Because of endian swaps between VMEbus and PCI bus, swap on size function swaps bytes and words for 32-bit access. Swapping bytes for 16-bit access does nothing for byte accesses. The simplest way of using the VMIPCI-5576 hardware to take care of the byte-ordering problem is to configure it to swap according to size (see Table 3-12 "Size Swap for Endian Conversion" on page 3-23).

Table 3-12: Size Swap for Endian Conversions

Bits 2 through 0: Size Swap	000	NO SWAP STRAIGHT THROUGH, NO DATA CHANGE (DEFAULT).
	100	Byte Swap
	101	Word Swap
	110	Byte and Word Swap
	111	Swap on Size

Swap on size

a. Dword (32 bits), swap bytes and words

b. Word (16 bits), swap bytes

c. Byte (8 bits), no swap

NOTE: 

DWORD IS THE NAME CONVENTION USED IN PCI MANUALS FOR 32-BIT WIDE DATA TRANSFERS. OTHER PLACES IN THIS AND OTHER VMIxxx-5576 MANUALS USE THE NAME CONVENTION LWORD OR LONG WORD FOR 32-BIT WIDE DATA.

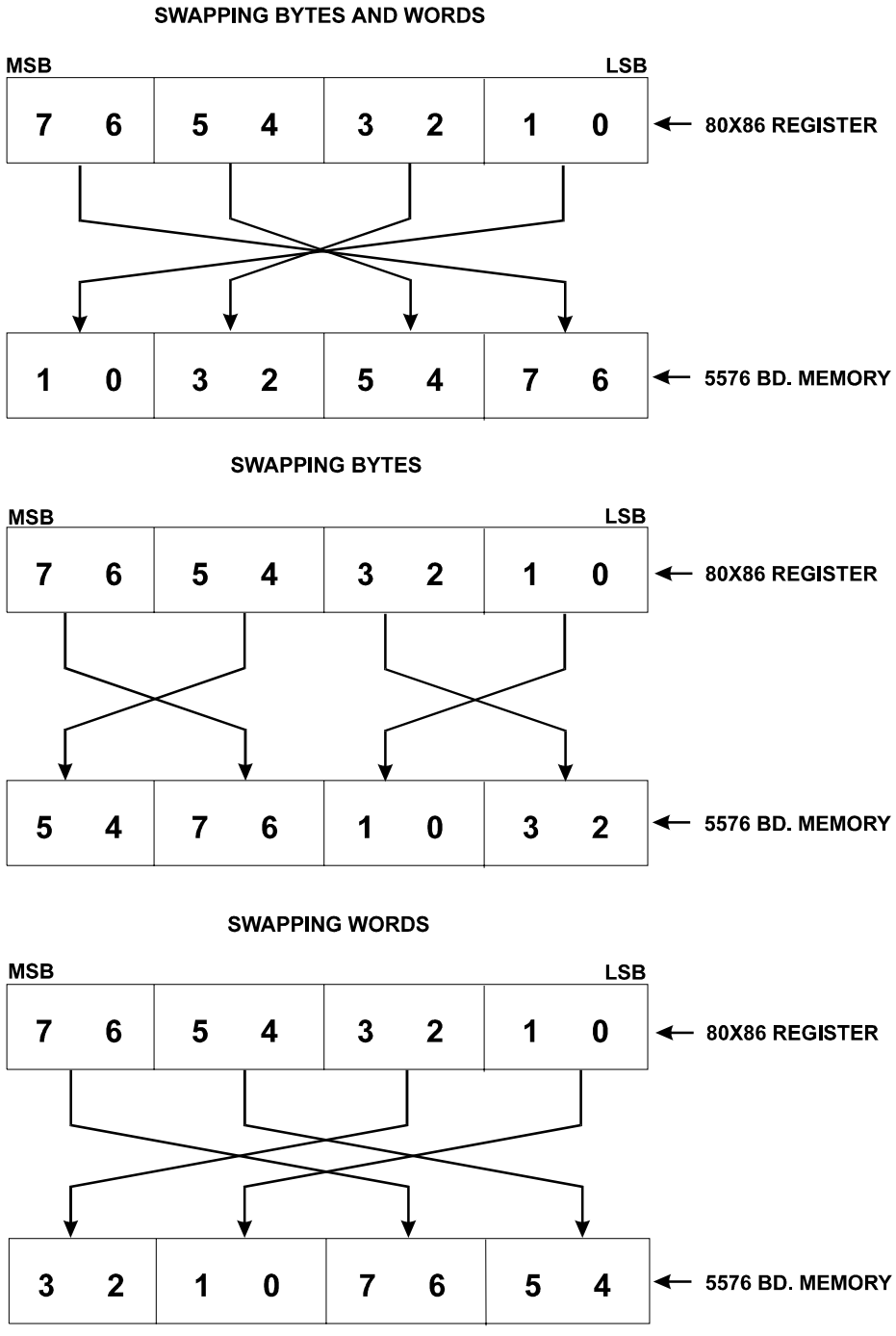


Figure 3-6. Example of Size Swap Endian Conversion

Command Node (CMDN) Register

The CMDN register contains the Node ID of the node that is to receive the command in the Command (CMD) Register. See Table 3-14 on page 3-26 for the CMD bit map.

Table 3-13: Command Node Register Bit Map

CMDN: RELATIVE OFFSET $10, READ/WRITE, BYTE/WORD							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
CMD TARGET NODE ID (0 TO 255)							

Command Register (CMD)

The VMIPCI-5576 can generate an interrupt to one or all other chassis through the use of the Command Register. Valid choices for interrupts are 1, 2, and 3. Table 3-15 shows all possible combinations. The interrupts are processed just like data so all words sent previous to the interrupt command are present on the receiving board's memory before the interrupt is issued to the receiving board.

Table 3-14: Command Register Bit Map

CMD: RELATIVE OFFSET $11, READ/WRITE, BYTE/WORD							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
X	ALL NODES	X	X	X	X	L1	L2

Command Register Bit Definitions

Bits 7, 5, 4, 3, and 2: -- X = Don't care.

Bit 6: **ALL NODES** -- Writing a one (1) to this bit location allows all nodes on the ring to receive commands. A zero (0) in this bit position allows a specific node to receive commands, identified in the CMDN.

Bits 1 and 0: **Interrupt Levels[1..0]** -- When these bits are set to one (1), an interrupt has been received intended for the local node.

Table 3-15: Interrupt Levels

D7	D6	D5	D4	D3	D2	D1	D0	Function
						0	0	NONE (NOT VALID)
						0	1	INTERRUPT 1 IS GENERATED
						1	0	INTERRUPT 2 IS GENERATED
						1	1	INTERRUPT 3 IS GENERATED

The act of writing to the CMD Register actually sends the INT CMD to the fiber-optic link FIFO queue.

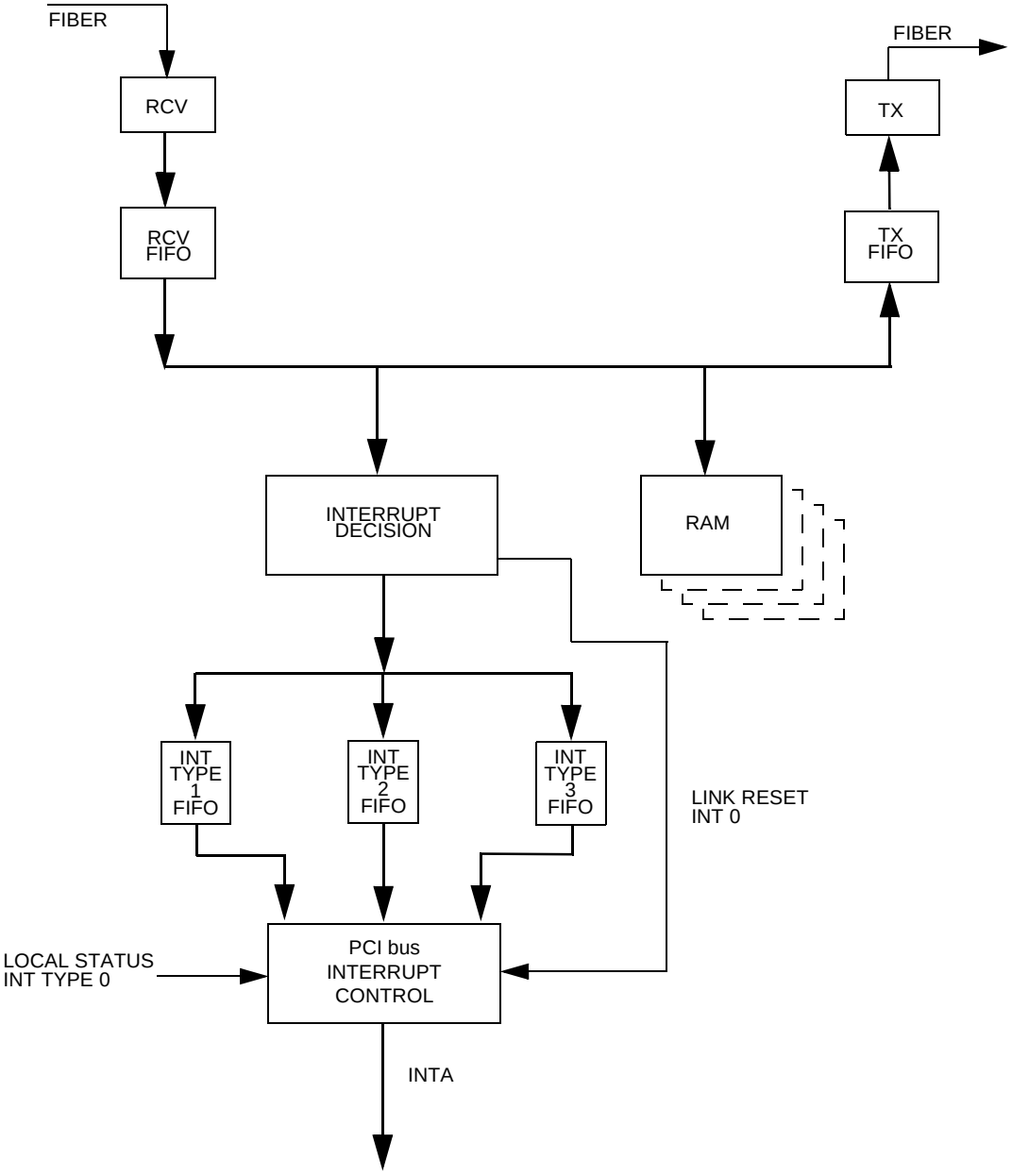


Figure 3-7. Interrupt/Data Receive Path

Interrupt Control and Status Register (ICSR)

The ICSR is used to enable the different interrupts.

Table 3-16: Interrupt Control and Status Register Bit Map

ICSR: RELATIVE OFFSET $14, READ/WRITE, BYTE/WORD							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
FLAG 1	GLOBAL INT EN	EN INT BAD DATA RECEIVED	EN INT RCVR FIFO HF	Reserved	EN INT INT3 PENDING	EN INT INT2 PENDING	EN INT INT1 PENDING

Interrupt Control and Status Register Bit Definitions

Bit 7: **FLAG 1** -- User-defined read / write flag. No specific purpose.

Bit 6: **GLOBAL INT ENABLE** -- Writing a one (1) to this bit location enables individual bits 5 through 0. This bit is cleared after each time the PCI bus is interrupted and must be rearmed if more interrupts are desired.

Bit 5: **EN INT, BAD DATA RECEIVED** -- This bit is an enable bit which allows interrupts to be generated if bad data is received. A one (1) equals Bad Data Interrupt Enabled. Bit 6 must be set to enable interrupts.

Bit 4: **EN INT, RCVR FIFO HALF-FULL** -- This bit is an enable bit, that when set to a one (1) allows the RCVR FIFO half-full flag to generate an interrupt. A one (1) equals RCVR FIFO half-full interrupt enabled. Bit 6 must be set to enable interrupts.

Bit 3: **RESERVED** -- Write to zero.

Bit 2: **EN INT, INT3 PENDING** -- When this bit is set to one (1), an interrupt is generated if there is a level 3 INT pending and the global INT EN is active.

Bit 1: **EN INT, INT2 PENDING** -- When this bit is one (1), an interrupt is generated if there is a level 2 INT pending and the global INT EN is active.

Bit 0: **EN INT, INT1 PENDING** -- When this bit is set to one (1), an interrupt is generated if there is a level 1 INT pending and the global INT EN is active.

NOTE: **THE IRS REGISTER MUST BE READ TO DETERMINE WHO GENERATED AN INTERRUPT IF MORE THAN ONE SOURCE IS ENABLED.**

Interrupt Sender ID Registers

These registers contain the ID of the node that send the local node interrupt types 1 through 3.

Table 3-17: Interrupt Sender ID Registers Bit Maps

INT1 Sender ID: Relative Offset $18 Read/Write, Byte							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
INT1 Sender ID							

INT2 Sender ID: Relative Offset $1C Read/Write, Byte							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
INT2 Sender ID							

INT3 Sender ID: Relative Offset $20 Read/Write, Byte							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
INT3 Sender ID							

Interrupt Sender ID Registers, Bit Definitions

Bits 7 through 0: **INT1 Sender ID** -- Contains the ID of the node that sent the local node INT Type 1. A write to this location clears all pending INT1 interrupts.

Bits 7 through 0: **INT2 Sender ID** -- Contains the ID of the node that sent the local node INT Type 2. A write to this location clears all pending INT2 interrupts.

Bits 7 through 0: **INT3 Sender ID** -- Contains the ID of the node that sent the local node INT Type 3. A write to this location clears all pending INT3 interrupts.

In the interrupt handling sequence, the user should perform only one read per interrupt cycle. Erroneous results are caused by multiple reads.

Each Sender ID register is implemented with a FIFO. When the user reads a sender ID, that particular Node ID is removed from the FIFO. If the user attempts to read the sender ID again, the output will be the next Node ID in the FIFO. However, if the FIFO is empty, the output is indeterminent. Therefore, do not key on the presence of any value in the Sender ID register to indicate a pending interrupt. Instead, use the Interrupt and Receive Status register.

DMA Start Address Register (DADD)

This register is initialized with the internal address pointer for DMA transfers. Once loaded it will count every DMA transfer.

Register Name: DMA Start Address

Address Offset 24h

Power up Value: 00000000h

Attribute: Write Only

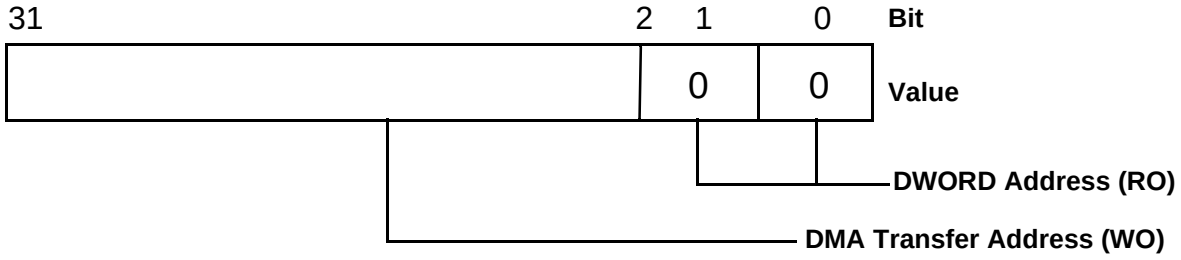


Figure 3-8. DMA Start Address Register

Using Interrupts

Due to the architecture of the VMIPCI-5576 board, the interrupts are enabled and handled in a tiered fashion. The PCI interface chip is used to generate the PCI interrupts and other logic is used to tell the PCI interface chip when and what to do. The control registers in the PCI interface chip are different than the main board control registers. Therefore, both PCI I/O access and memory accesses are required to use interrupts. The proper sequence of events is outlined below:

Enabling Interrupts:

1. Select what sources can enable interrupts and enable the interrupt process by writing to the ICSR (memory base + $14) as outlined in the programming section (for example, 43 would enable interrupts for INT1 or INT2 pending).

2. Enable the PCI Interface Chip by writing H"0000:1000" to the PCI INTCSR control register (I/O base + $38).

Disabling Interrupts:

Either disable the PCI Interface Chip from generating interrupts by writing H"0000:0000" to the PCI Interface Chip Control register or disable the board control logic from generating interrupts by writing a "0" to bit 6, global interrupt enable, of the ICSR.

Servicing Interrupts:

1. To verify the PCI interrupt, the PCI Interface Chip INTCSR will have bit 17 set (logical one) (optional).

2. Read PCI I/F (I/O base + $10) and discard data - this clears the cause of the interrupt.

3. A write to the PCI Interface Chip's INTCSR with a H"0002:0000" will clear the interrupt off the PCI bus. Bit 17 is set when the interrupt is generated by the board's control logic.

4. To see why the interrupt was generated, read IRS (memory base + $8). All the conditions that cause interrupts are in this register. It is possible for multiple events to occur while an interrupt is being generated and as such multiple flags could be set. The host may service as many pending events as it desires during an interrupt sequence (if all flags that can generate interrupts are not serviced, the logic will generate a new interrupt after they are rearmed). If INT1, INT2, or INT3 are used, the appropriate SID Register must be read to clear the cause of the interrupt.

5. To rearm the interrupts, the Global interrupt enable must be reset in the ICSR control register (memory + $14). This bit was cleared when the interrupt was sent to the PCI Interface Chip.

6. The PCI Interface Chip must be rearmed by writing H"0000:1000" to the INTCSR.

Generating Interrupts on other Nodes:

1. Write the node to be interrupted to the Command Node register. (This can be done simultaneous with the write to the CMD register in step 2.)

2. Write the desired INT level to the Command register. See the examples below:

EX. 1) Write byte 22 to CMDN

2) Write byte 01 to CMD

This would generate a level 1 interrupt at node 22.

EX. Write byte 43 to CMD

This would generate a level 3 interrupt in all nodes.

DMA Features

The VMIPCI-5576 is capable of being set up as a DMA Initiator on the PCI bus. This feature is initialized by another PCI Initiator who wishes to off-load the process of moving blocks of data to or from the VMIPCI-5576 to another memory location. This is very useful for moving large blocks of data to or from system memory and relieving the processor from this time consuming activity. The VMIPCI-5576 can move data blocks up to 1 Mbyte at burst rates over 33 Mbyte/sec. Large blocks of data are broken into smaller blocks by the VMIPCI-5576 so that the PCI bus is not monopolized. The VMIPCI-5576 will request the PCI bus when it needs to move data. The period of time, in PCI clock cycles, that the VMIPCI-5576 will use the bus once the bus is granted is programmable. The VMIPCI-5576 will move as much data as possible during this time and then relinquish the bus and request the bus again if more data needs to be moved. The VMIPCI-5576 can be configured to generate a PCI interrupt upon completion of a DMA cycle or the board can be polled to see if the DMA process is complete or how many more bytes are remaining to be moved. The VMIPCI-5576 PCI interface contains FIFOs that are utilized during the DMA process to quicken the data movement. During a DMA write, these FlFOs are being filled even before the VMIPCI-5576 is granted the bus. This allows some burst to be at the maximum PCI data rate of 132 Mbyte/sec but only about ten longwords (40 bytes). The maximum sustained rate can drop to 33 Mbytes/sec for longer burst. During DMA reads, these FlFOs can be filled at the maximum of 132 Mbyte/sec but once filled the data rate will be slowed to 33 Mbyte/sec. The 33 Mbyte/second will be further throttled to 6.2 Mbyte once the reflective memory transmit FIFOs are filled. An illustration of the DMA data path is shown in Figure 3-9 on page 3-35.

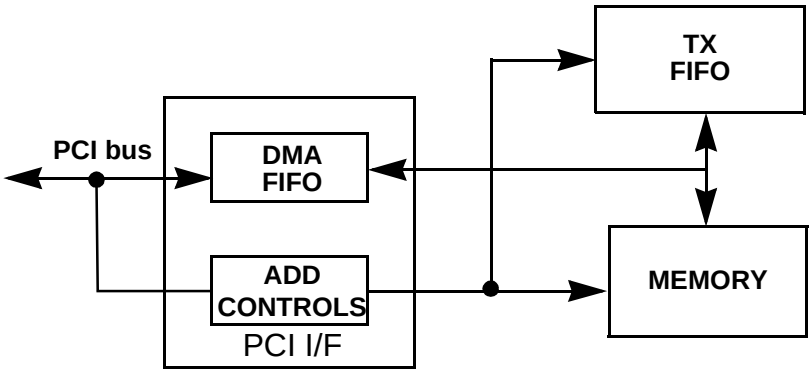


Figure 3-9. DMA Data Path

DMA Write Procedure

The following is a list of procedures that must be followed to perform DMA Writes (from the VMIPCI-5576 to a PCI target device):

1. Clear PCI I/F DMA Write FIFO, I/O write to offset 3c with value 04000000.

2. Initialize Target address, I/O write to offset 24 the target address (R/W 32-bit).

3. Initialize Transfer Byte Count, I/O Write to offset 28 total bytes (R/W 32-bit).

4. Initialize VMIPCI-5576 Initiator address, Memory-Mapped control register at offset 24.

5. Enable the VMIPCI-5576 DMA Write bit, Memory-Mapped CSR1 bit 3.

6. Enable VMIPCI-5576 PCI bus Master, I/O Write long to offset 3c value of 400.

7. After DMA cycle is complete, disable bus mastership, offset 3c value of 0.

8. Disable internal DMA control CSR1 bit 3.

Example DMA Write

The above procedure is done by the following instructions to perform a DMA write from a VMIPCI-5576 to system memory. The VMIPCI-5576 board is mapped at F000 0000 for memory and 6100 for I/O. The following procedure will move 40(H) bytes, 10(H) longwords from the block F000 0040 to F000 007C to the block 100000 to 10003C.

1. IWL 613C 04000000 ;Clear DMA FIFO
2. IWL 6124 100000 ;Initialize Target Address
3. IWL 6128 40 ;Initialize Transfer Byte Count
4. MWL F0000024 F0000040 ;Initialize Initiator Address
5. MWB F0000009 8 ;Enable VMIPCI-5576 capability
6. IWL 613C 400 ;Enable VMIPCI-5576 to become PCI Initiator
7. IRL 6128 ;Poll to verify the DMA process is complete

   = 0 ;(Interrupts can also flag the DMA termination)

   IWL 613C 0 ;Disable VMIPCI-5576 from requesting bus
8. MWB F0000009 0 ;Disable internal DMA data movement.

DMA Read Procedure

The following is a list of procedures that must be followed to perform DMA Read (from the PCI target Device to VMIPCI-5576):

1. Clear DMA FIFO, I/O Write to offset 3c value 02000000.
2. Initialize Target address, I/O Write to offset 2C the target address (R/W 32-bit).
3. Initialize Transfer Byte Count, I/O Write to offset 30 total bytes (R/W 32-bit).
4. Initialize VMIPCI-5576 Initiator address, Memory-Mapped control register at offset 24.
5. Enable the VMIPCI-5576 DMA Read bit, Memory-Mapped CSR1 bit 1.
6. Enable VMIPCI-5576 PCI bus Master, I/O Write long to offset 3c value of 4000.
7. After DMA cycle is complete, disable bus mastership, offset 3c value of 0.
8. After DMA cycle is complete, disable internal movements by clearing CSR1 bit 1.

Example DMA Read

The above procedure is done by the following instructions to perform a DMA read from system memory to a VMIPCI-5576. The VMIPCI-5576 board is mapped at F000 0000 for memory and 6100 for I/O. The following procedure will move 40(H) bytes, 10(H) longwords from the block 100000 to 10003C to the VMIPCI-5576 at the block F000 0040 to F000 007C.

1. IWL 613C 02000000 ;Clear DMA FIFO
2. IWL 612C 100000 ;Initialize Target Address
3. IWL 6130 40 ;Initialize Transfer Byte Count
4. MWL F0000024 F0000040 ; Initialize Initiator Address
5. MWB F0000009 2 ;Enable VMIPCI-5576 DMA capability
6. IWL 613C 4000 ;Enable VMIPCI-5576 to become PCI Initiator

   IRL 6130 ; Poll to verify the DMA process is complete

   = 0 ;(Interrupts can also flag the DMA termination)

   IRL 613C ;Make sure internal data movements are complete

   bit 2 = 1
7. IWL 613C 0 ;Disable VMIPCI-5576 from requesting bus
8. MWB F0000009 0 ;Disable internal DMA movements.

DMA Interrupts

The VMIPCI-5576 can generate a PCI interrupt upon completion of a DMA transfer. The procedure is slightly different for reads and writes, but it is generally the same. The I/O mapped register INTCSR at offset 38 is used to enable interrupts. This is the same register that is used by other reflective memory interrupts. A method to determine why the interrupt has occurred is also in this register. The bit map for this register is below.

Bit 23 Interrupt Asserted (From any cause, Read Only)

Bit 21 Target Abort (A Target Abort occurred while VMIPCI-5576 was Initiator)

Bit 20 Master Abort (A Master Abort occurred while VMIPCI-5576 was Initiator)

Bit 19 DMA Read Complete (Offset 30 = 0, end of DMA read cycle)

Bit 18 DMA Write Complete (Offset 28 = 0, end of DMA write cycle)

Bit 17 Mailbox Int (Standard reflective memory interrupt pending)

Bits 17 through 21 are Read/Write One to Clear type.

Bit 15 Interrupt on Read Transfer Complete Enable (enable set to 1)

Bit 14 Interrupt on Write Transfer Complete Enable (enable set to 1)

Bit 12 Interrupt on Incoming Mailbox Enable (enable set to 1)

To arm the interrupts for use with the DMA procedure, the transfer count must first be loaded into the appropriate register before the interrupt is enabled. This would normally be done immediately before the VMIPCI-5576 is enabled to become a PCI initiator (step 5 in the procedures section).

DMA Restrictions

The following restrictions apply to VMIPCI-5576 activities. There are no restrictions on target accesses while the VMIPCI-5576 is enabled as a PCI Initiator in DMA mode.

1. All DMA cycles move 32-bit wide data words.

2. The maximum byte transfer count is 1 Mbyte (the maximum reflective memory size).

3. The DMA capability can be used for both DMA reads and DMA writes, but not both simultaneously. The user should only enable one at a time in the control registers.

4. Data rates. The VMIPCI-5576 can do a DMA read at rates of 132 Mbyte/sec maximum and 33 Mbyte/sec sustained. This data rate will be further throttled down to the 6.2 Mbyte/sec, the reflective memory network speed, after the transmit FIFOs are full. This allows all data written to the board to be broadcast out on the reflective memory network. This ensures no data will be lost.

5. To abort any DMA process, the transfer count should be written to four so that the PCI I/F will terminate normally after the next transfer. This is due to an errata in the PCI I/F. After the next generation chips are here, this restriction will no longer be valid and the user can just abort the DMA process by disabling the PCI Initiator enable bit.

CHAPTER

4

Maintenance

Maintenance

This section provides information relative to the care and maintenance of VMIC's products. If the products malfunction, verify the following:

a. Software

b. System configuration

c. Electrical connections

d. Jumper or configuration options

e. Boards are 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 chassis

h. Quality of cables and I/O connections

If products must be returned, contact VMIC for a Return Material Authorization (RMA) Number. **This RMA Number must be obtained prior to any return.**

Maintenance Prints

User level repairs are not recommended. The appendix to this manual contains drawings and diagrams for reference purposes only.

Troubleshooting Hints

The following pages list problems you can have when using the VMIPCI-5576. The list also contains the possible causes and solutions for those problems.

SYMPTOM	POSSIBLE CAUSES
Communications lost after power down of one or more nodes on the link.	1) The VMIPCI-5576 is a ring architecture network and all data must be repeated at each node. If a node is powered down, the ring is opened at that point. The fiber-optic cable may be moved to complete the ring again or a fiber-optic bypass switch may be used to automatically complete the ring.
Data written to one node does not appear in other nodes.	1) For nodes of memory size 1 Mbyte or smaller, a 1 Mbyte relative address is passed. If two 0.25 Mbyte boards are not mapped in the same relative address in relation to the 1 Mbyte boundary, then they will not appear to communicate since they contain no common space in RAM. 2) The sending node ID is the same as another node on the ring. Each node must have a unique node ID, otherwise data originated on one node will be removed by the other node of the same ID. 3) Open or defective connection to fiber-optic cable. 4) Defective FIFO module on either receiving node or transmitting node (the bad board may be found by checking memory on other nodes to see if they received data correctly). 5) Mixed nodes on the link (redundant and single transfer nodes on link). All nodes must be in either redundant or single transmission mode.
Data bits dropped or data appears in wrong address in memory.	1) Damaged or defective FIFO on node. 2) Damaged cable or connector to link.

SYMPTOM	POSSIBLE CAUSES
Data is wrong in one node and is correct in all other nodes.	1) Possible write to same memory location in two or more boards (in separate nodes) at the same time (this must be prevented in software). 2) Damaged local node.
Fiber-optic errors commonly seen on local node.	1) Defective fiber-optic cable. It may have insufficient length-bandwidth or simply be broken. Light loss error budget has been exceeded. 2) Damaged fiber-optic transmitter or receiver. 3) Damaged node on ring (if parity errors are seen on a node, the failing node is either the local node or the node sending data to the local node).