VMIVME-5576
REFLECTIVE MEMORY BOARD

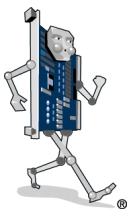
INSTRUCTION MANUAL

DOCUMENT NO. 500-005576-000 F

Revised August 4, 1998

VMIC
12090 SOUTH MEMORIAL PARKWAY
HUNTSVILLE, ALABAMA 35803-3308
(256) 880-0444
1-800-322-3616
FAX NO.: (256) 882-0859

COPYRIGHT AND TRADEMARKS



(I/O man figure)

(Instant OPC wizard logo)

(IOWorks man figure)

RECORD OF REVISIONS

REVISION LETTER	DATE	PAGES INVOLVED	CHANGE NUMBER
A	04/09/92	Release Per ECO	92-0085
B	06/29/92	Cover, Pages ii, v,vi, Section 2, Section 6, and Appendix A	92-0163
C	11/16/92	Cover, Pages ii, vi, and Section 5	92-0250
D	01/16/93	Add Appendix B	92-0535
E	11/04/93	Add Appendix C	93-0839
F	08/04/98	Cover, Pages ii, 3-1, 5-12, Notice Page, and Table of Contents	98-0498

VMIC
SAFETY SUMMARY

THE FOLLOWING GENERAL SAFETY PRECAUTIONS MUST BE OBSERVED DURING ALL PHASES OF THIS 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 THE 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.

WARNING

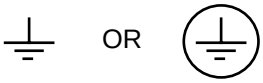
DANGEROUS VOLTAGES, CAPABLE OF CAUSING DEATH, ARE PRESENT IN THIS SYSTEM. USE EXTREME CAUTION WHEN HANDLING, TESTING, AND ADJUSTING.

SAFETY SYMBOLS

GENERAL DEFINITIONS OF SAFETY SYMBOLS USED IN THIS MANUAL

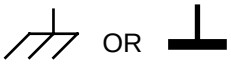
Instruction manual symbol: the product is marked with this symbol when it is necessary for the user to refer to the instruction manual in order to protect against damage to the system.

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



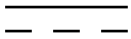
Protective conductor terminal. For protection against electrical shock in case of a fault. Used with field wiring terminals to indicate the terminal which must be connected to ground before operating equipment.

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.

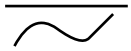


Frame or chassis terminal. A connection to the frame (chassis) of the equipment which normally includes all exposed metal structures.

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.

VMIVME-5576
REFLECTIVE MEMORY BOARD

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

TABLE OF CONTENTS (Concluded)

LIST OF FIGURES (Concluded)

LIST OF TABLES

APPENDIX

SECTION 1

INTRODUCTION

1.1 FEATURES

The VMIVME-5576 is a high performance, yet easy-to-use, method of linking from two to 256 VMEbus systems together by the use of global memory. Any data word written to a specific location in memory will show up in the same location in each of the other 255 nodes with no programming or other intervention required by the user.

The Reflective Memory Board has several unique features:

a. No software necessary to establish communications

b. On-board interrupt generation ability. Any node may generate interrupts on any or all other nodes on the system.

c. Facilitates communications over very long link lengths, i.e., up to 2,000 meters

d. Can be selected to operate in privileged or nonprivileged modes or in both modes at once

e. Memory can be configured to run in either A24 or A32 addressing schemes

f. Supports 8-, 16-, 24-, 32-bit transfers (bi-directional)

g. Attention interrupts are channeled to one of seven programmable interrupt levels

h. Double Eurocard form factor

i. Any board may be jumpered to be any Node

j. Selectable 256 K, 512 K, or 1 Mbyte of SRAM on-board

k. Options for 512 byte or 4 Kbyte FIFOs

1.2 FUNCTIONAL DESCRIPTION

The link between two nodes is established through the use of FIFO memory which is routed through fiber-optic drivers/receivers. Figure 1.2-1 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 only sees 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 SRAM will take slightly longer to DTACK when the FIFO is writing to RAM.

1.2.1 Software Requirements to Use Reflective Memory Board

The Reflective Memory Board upon power-up is able to establish the board-to-board link without any program setup. The Bus Interrupter Module (BIM) 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 registers in the BIM chip must be initialized through software control. A CSR is also present on the board which controls the Fail LED on the Reflective Memory Board. The user must Write to the CSR 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 self-test function so the LED being ON does not indicate a failure on-board unless the VME chassis software control actually has turned it ON. LED ON is a standard power-up mode for the Fail LED. If the output FIFO becomes over half full, the Reflective Memory Board will issue an interrupt if the BIM has been programmed and enabled. The user software may choose to ignore this warning. If the output FIFO becomes completely full, a Bus Error (BERR) will be issued to the user once an attempt is made to Write to a Reflective Memory Board that has a full output FIFO.

1.2.2 Hardware Requirements to Use Reflective Memory Board

Aside from the address map decoding required on the Reflective Memory Board there are a few system jumpering requirements which must be followed to allow the system to work. 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, i.e., 0,1...255. Nodes may be intermixed in any order as far as unique board IDs are concerned. There is parity and other error checking hardware on the link so the Reflective Memory Board will inform the user if an improper condition in the link exists. Each Reflective Memory Board may be mapped into a different address space. Data will appear in the same location in each node relative to the base 1 Mbyte boundry each Reflective Memory is mapped to.

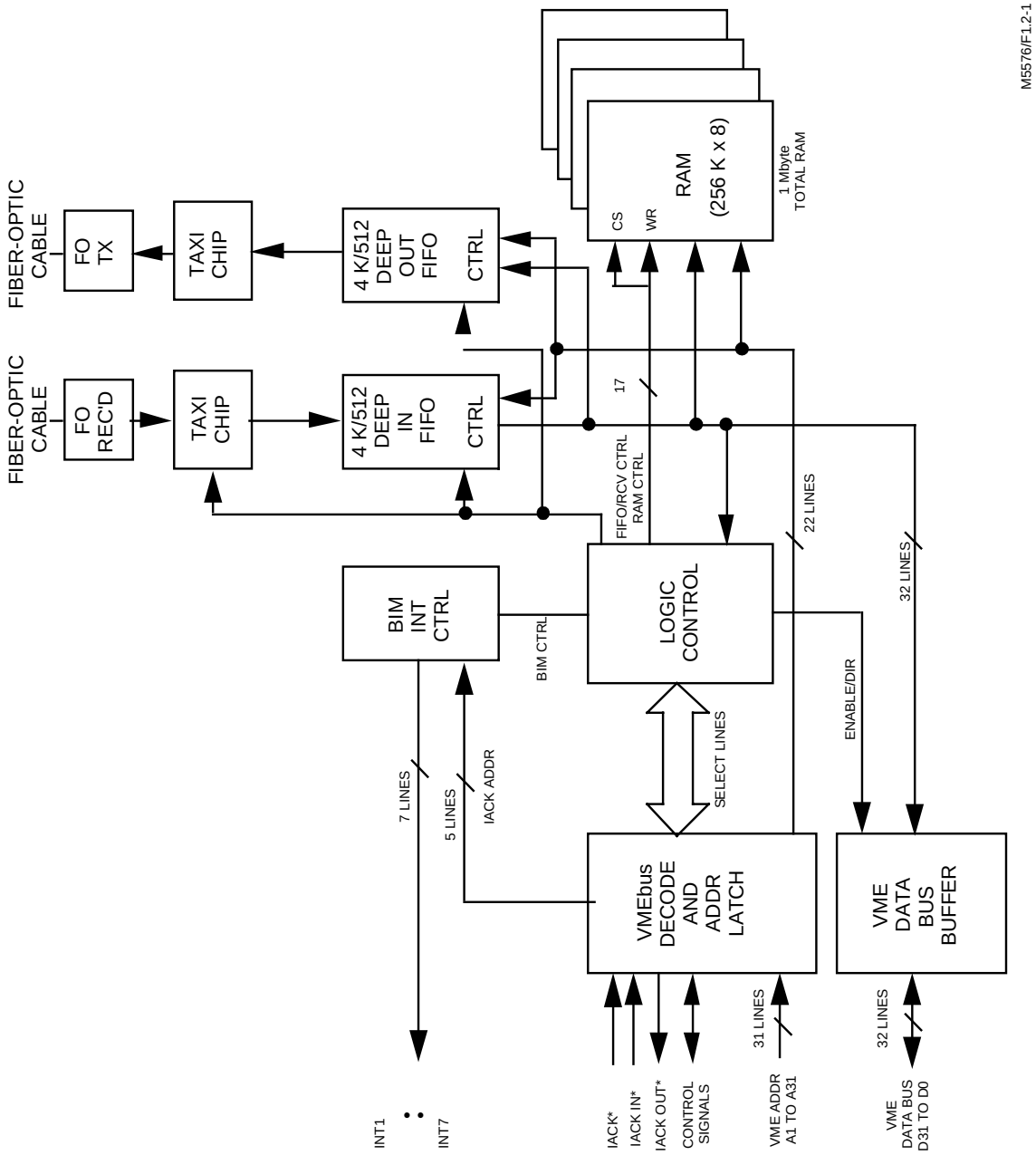


Figure 1.2-1. VMIVME-5576 Functional Block Diagram

There is a restriction on address configuration relating to each 1 Mbyte boundary for the 512 K and 256 K option Reflective Memory Boards. Even though the 512 K option may be mapped on any 512 K boundary, relative to the 1 Mbyte boundaries, there is an upper and a lower position (see Figure 3.5-1). All boards must be mapped to the same position relative to 1 Mbyte boundaries.

There are four positions possible with the 256 K Reflective Memory Boards relative to 1 Mbyte boundaries. All 256 K boards on the link must be mapped to the same position relative to the 1 Mbyte boundaries in order to communicate. There is no restriction between which 1 Mbyte boundary each board is on. Exact address matching is not required; only the position relative to the nearest 1 Mbyte boundary must be the same.

SECTION 2

PHYSICAL DESCRIPTION AND SPECIFICATIONS

REFER TO 800-005576-000 SPECIFICATION

SECTION 3

THEORY OF OPERATION

3.1 OPERATIONAL OVERVIEW

The VMIVME-5576 allows up to 256 VMEbus chassis to be linked together in a sequential fashion. Data written to any node appears in all other nodes some period of time later. The link between the nodes is two fiber-optic cables which pass address, data, and interrupt information between adjacent boards on the link.

The VMIVME-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 chassis by writing a data word into a specific location in RAM. 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 VMEbus, so if a block of data was written to the board before the interrupt command is sent, then the data will be broadcast to all boards before the interrupt command is broadcast.

NOTE:

IF THERE ARE EMPTY SLOTS TO THE LEFT OF THE VMIVME-5576, THEN IACK JUMPERS MUST BE INSTALLED FOR THE EMPTY SLOTS. OTHERWISE, THE VMIVME-5576 WILL INTERMITTENTLY FAIL TO RESPOND TO VMEbus READS AND WRITES.

3.2 BASE ADDRESS SELECTION

The base address of the VMIVME-5576 is jumper-selectable. Once the base address is established, all other writes to the VMIVME-5576 will be relative to the base address. Each node on the link may have a different base address. Only the relative offset to the base address is passed across the link. Thus, each board on the link may have a different VMEbus address and addressing mode.

3.3 FIBER-OPTIC LINK SPEED SELECTION

At full speed the link can support a 6.2 Mbyte data transfer rate. However, the high-speed serial fiber optic's bit error rate may result in erroneous data being transferred. A slowdown option has been included to allow the link to be slowed down by a factor of two. VME interface response time will not be affected by the data link slowdown. All transfer errors are detectable via parity checking and on-board receiver error detection circuitry. The slowdown mode

results from sending each data twice. The redundant transmissions statistically lowers the probability of data corruption. The probability of data transmission failure of both transmissions is once every three thousand years. Redundant transmissions guarantee that all data will arrive across the link correctly. A fiber-optic transfer error is a rare event and the board which has the receive error has the capability to notify the local VME chassis that it has occurred. In most systems it should be preferable to operate in single transmission mode to maintain high data throughput. In the case a retransmission request is too slow, the double transmission mode is available.

3.4 BUS INTERRUPTER MODULE (BIM)

To facilitate handling of interrupts, an MC68153 is used on the board. The attention interrupts (INT1, INT2, and INT3) are used to signal an interrupt from a remote chassis. An interrupt (INT0) may be generated when the transmit FIFO on the node being written to by the VMEbus becomes over half full. The transmit FIFO over half full condition occurs when the local node has received data from the local VME chassis but has not been granted permission to transmit its data on the link. All interrupts are masked off at power-up and become enabled under program control.

3.5 ADDRESSING FEATURES

Not all nodes have to be configured with the same memory size. Nodes may be configured to make optimum use of memory. An example is shown in Figure 3.5-1.

3.6 NODE LATENCY

If the fiber-optic data bandwidth has not been exceeded, data latency is typically 1.5 µs/node in single transfer mode. Longer latencies will result if data input rates exceed 6.2 Mbytes/sec for a period of time. The transmit FIFOs will back up with data until the half-full interrupts are set off or a bus error occurs in the event the FIFOs become full.

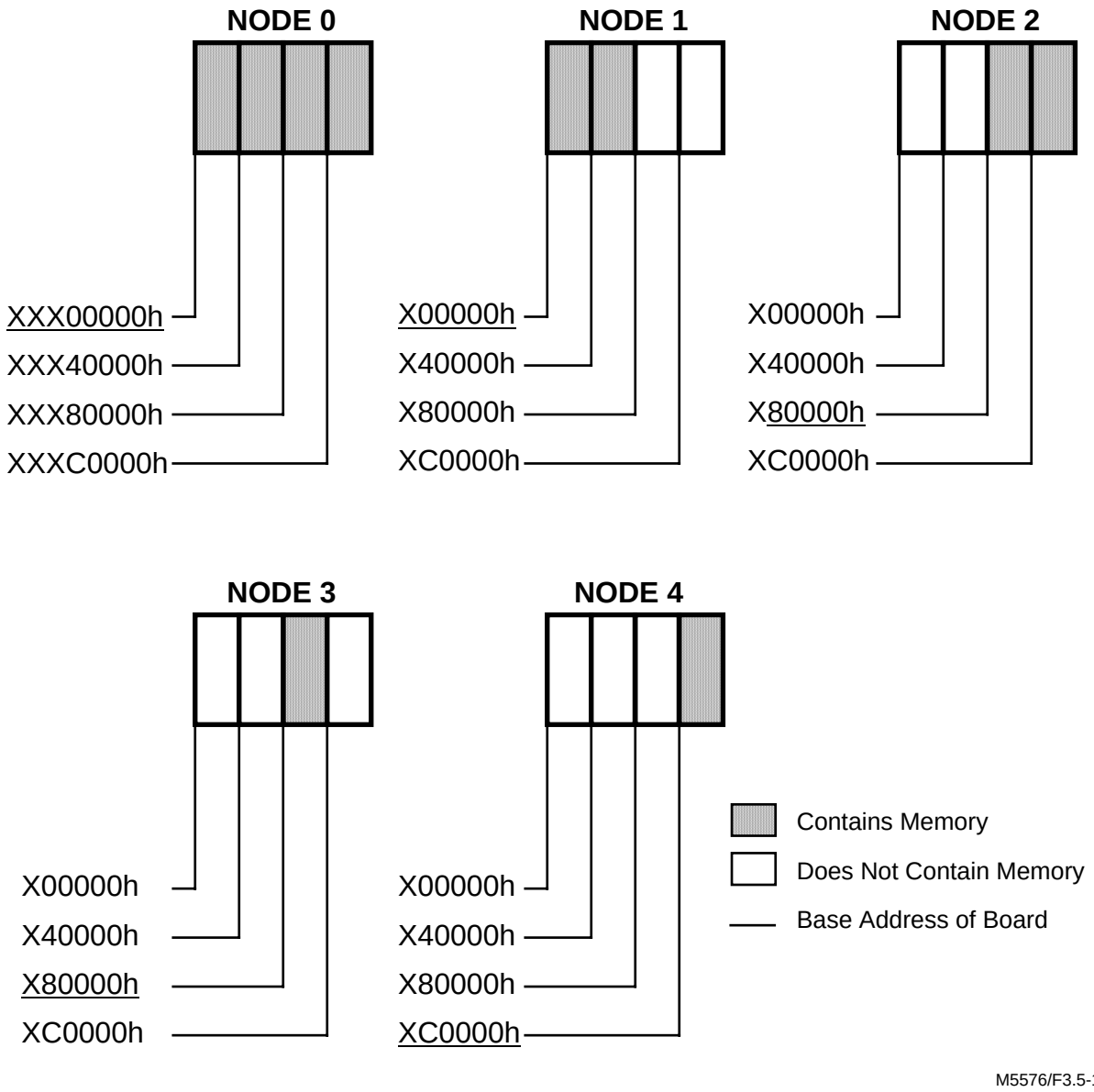


Figure 3.5-1. Example Memory Configurations

Node 0 is configured with 1 Mbyte of memory, Nodes 1 and 2 are configured with 512 K of memory, and Nodes 3 and 4 are configured with 256 K of memory. Because of the relative location of the memory, data written to address range X00000h to X7FFFFh by a processor in Node 0 will also be written into the corresponding address in Node 1. However, no data will be written into Nodes 2, 3, and 4 because there is no memory in that range in those nodes. Similarly, data written into address range X80000h to XBFFFFh in Node 2 will also be written into Nodes 0 and 3, but not Nodes 1 and 4.

SECTION 4

PROGRAMMING

4.1 PROGRAMMING

Although the VMIVME-5576 VMEbus Reflective Memory Board is software transparent on power-up, some registers are present to facilitate user information and interrupt generation. Table 4.1-1 shows the memory mapped registers used by the VMIVME-5576.

4.2 BOARD IDENTIFICATION (ID) REGISTER

The Board ID Register allows the user to verify the presence of the Reflective Memory Board at the correct address. The VMIVME-5576 Board ID number is $18 HEX.

4.3 NODE ID REGISTER

A value from 0 to 255 can be read from this byte. The value read corresponds to the node ID jumper selected for the board.

4.4 BOARD CONTROL AND STATUS REGISTER (CSR)

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

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LED	RCV HALF-FULL	TX HALF-FULL	TX EMPTY	BAD DATA	OWN DATA	MASK	FAST

BIT 0 - Fast mode (6.2 Mbytes/sec) if high (Read Only)

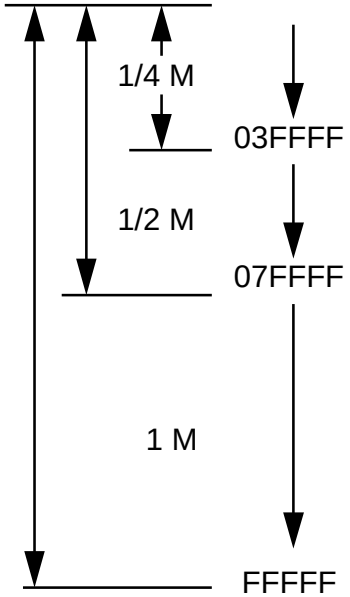
Fast mode transmits each data transfer once on the fiber-optic link. If the jumper J5 is installed, each transfer is sent twice on the fiber-optic link (3.2 Mbytes/sec).

BIT 1 - Mask Transfer Error Interrupt Masks on high (Read Only)

If the mask jumper J3 is installed and INT0 is enabled by software on the bus interrupt module, an interrupt will be generated each time a receive error is detected on the fiber-optic link.

Table 4.1-1. VMIVME-5576 Memory Map

RELATIVE ADDRESS (HEX)	DESCRIPTION	CONTENTS	WRITE-READ MODE
01	Board ID	ID (18 H)	Byte (R)
04	Node ID	Node No.	Byte (R)
05	CSR	Status Flags	Byte (R/W)
06	CMD Register	Interrupt Address	Byte (W)
07	CMD Node	Node to get Interrupt	Byte (R/W)
23	INT0 Mode Control	(MC68153 BIM)	Byte (R/W)
26	INT1 Sender ID	Node ID of INT1 Sender	Byte (R/W)
27	INT1 Mode Control	(MC68153 BIM)	Byte (R/W)
2A	INT2 Sender ID	Node ID of INT2 Sender	Byte (R/W)
2B	INT2 Mode Control	(MC68153 BIM)	Byte (R/W)
2E	INT3 Sender ID	Node ID of INT3 Sender	Byte (R/W)
2F	INT3 Mode Control	(MC68153 BIM)	Byte (R/W)
33	INT0 Vector	(MC68153 BIM)	Byte (R/W)
37	INT1 Vector	(MC68153 BIM)	Byte (R/W)
3B	INT2 Vector	(MC68153 BIM)	Byte (R/W)
3F	INT3 Vector	(MC68153 BIM)	Byte (R/W)
40	RAM	SRAM	All modes (Byte, Word, Lword) R/W



M5576/T4.1-1

BIT 2 - Own-data (high if link intact) (Read/Write)

This bit indicates that the local node has received data back that had originated on the local node. It may be reset by writing this bit to a logic "zero". Once set by receiving its own node ID from the fiber-optic receiver, the bit remains set until cleared by the local VME side.

BIT 3 - Bad-data (high if error occurred) (Read/Write)

This bit indicates that a single transfer error has occurred. It does not depend on the mask jumper to be removed. In redundant transmission mode (3.2 Mbyts/sec), it indicates only that a transfer error occurred on one of the two transfers.

BIT 4 - Transmit FIFO empty - empty when low (Read Only)

BIT 5 - Transmit FIFO half-full - over half-full when low (Read Only)

These two bits display status on the transmit FIFO on the local node. The current status of the transmit status is displayed.

BIT 6 - Receive FIFO half-full at least half-full when low (Read Only)

This bit displays current status of the receive FIFO. This bit should never be low. If this bit goes low, local VME access should be suspended or at least curtailed for a period of time or data loss could occur. This bit going low would be an indication of a local VME problem, i.e., extremely slow release of data strobes after DTACK. In a proper functioning VME system, this should never occur.

BIT 7 - Fail LED Status (Read/Write) logic 1 = LED ON, logic 0 = LED OFF

4.5 PROGRAMMING THE MC68153 BIM

The MC68153 contains one Interrupt Control Register (ICR) and one Interrupt Vector Register (IVR) for each of the four interrupt sources. All four Control Registers are identical. All four Vector Registers are also identical.

4.5.1 Register Description

The MC68153 contains eight programmable Read/Write Registers. There are four Control Registers (CRINT0 through CRINT3) that govern operation of the device. The other four (VRINT0 through VRINT3) are Vector Registers that contain the vector data used during an interrupt acknowledge cycle.

4.5.2 Control Registers (23H, 27H, 2BH, 2FH)

There is a Control Register for each interrupt source, see Table 4.9-1, i.e., CR0 controls INT0, CR1 controls INT1, etc. Each Control Register is divided into several fields:

a. Interrupt level (L2, L1, L0) - The least significant 3-bit field of the register determines the level at which an interrupt will be generated:

L2	L1	L0	IRQ LEVEL
0	0	0	DISABLED
0	0	1	IRQ1
0	1	0	IRQ2
0	1	1	IRQ3
1	0	0	IRQ4
1	0	1	IRQ5
1	1	0	IRQ6
1	1	1	IRQ7

A value of "zero" in the field disables the interrupt.

b. Interrupt Enable (IRE) - This field (Bit 4) must be set (high level) to enable the bus interrupt request associated with the Control Register. Thus, if the INTX line is asserted and IRE is cleared, **no** interrupt request (IRQX) will be asserted.

c. Interrupt Auto-Clear (IRAC) - If IRAC (Bit 3) is set , IRE (Bit 4) is cleared during an interrupt acknowledge cycle responding to this request. This action of clearing IRE disables the interrupt request. To re-enable the interrupt associated with this register, IRE must be set again by writing to the Control Register.

d. External/Internal (X/IN) - Bit 5 of the Control Register determines the response of the MC68153 during an interrupt acknowledge cycle. If the X/IN bit is clear (low level) the BIM will respond with vector data and a DTACK signal, i.e., an internal response, If X/IN is set, the vector is not supplied and no DTACK is given by the BIM, i.e., an external device should respond. Always set to "zero" for the VMIVME-5576.

e. Flag (F) - Bit 7 is a flag that can be changed without affecting chip operation.

f. Flag Auto-Clear (FAC) - If FAC (Bit 6) is set, the Flag bit is automatically cleared during an interrupt acknowledge cycle.

4.5.3 Vector Registers (33H, 37H, 3BH, 3FH)

Each interrupt input has its own associated Vector Register, see Table 4.9-2. Each register is eight bits wide and supplies a data byte during its interrupt acknowledge cycle if the associated External/Internal (X/IN) Control Register bit is clear ("zero"). This data can be status, identification, or address information depending on system usage. The information is programmed by the system user.

4.5.4 Device Reset

When the MC68153 is reset, the registers are set to a known condition. The Control Registers are set to all "zeros" (low). The Vector Registers are set to $0F. This value is the MC68000 vector for an uninitialized interrupt vector.

4.6 THE COMMAND REGISTER DEFINITION (06H)

The VMIVME-5576 may generate an interrupt in any or all other chassis through the use of the Command Register. Valid choices for interrupts are: 1,2,3. Table 4.6-1 shows all combinations possible. The interrupts are processed just like data so all words sent previous to the interrupt command will be present on receiving board's memory before the interrupt will be issued to the receiving board. The Command Register is Write only.

Table 4.6-1. VMIVME-5576 Interrupt Codes

D7	D6	D5	D4	D3	D2	D1	D0	FUNCTION
						0	0 —	not valid, no interrupt is generated
						0	1 —	interrupt 1 is generated (INT level set by CRINT1)
						1	0 —	interrupt 2 is generated (INT level set by CRINT2)
						1	1 —	interrupt 3 is generated (INT level set by CRINT3)
X	1	X	X	X	X	X	X—	interrupt is generated in all chassis
X	0	X	X	X	X	X	X—	interrupt is generated in chassis ID which was written previously at relative address 07H. A word or Lword Write can specify both interrupt type and receiving Node ID in one transfer.

M5576/T4.6-1

In the event of an external interrupt (i.e., int1, int2, int3), the VMIVME-5576 will prevent the loss of any subsequent interrupt of the same type through the use of a dedicated FIFO for each interrupt type. The interrupt handler must execute a Read of the sender ID Register in order to allow the next interrupt of the same type to be sent to the BIM. An example is that an interrupt int1 has been sent across the link immediately followed by a second int1. The receiving node must determine that the VMIVME-5576 has issued a int1 through the use of the resulting VME int level (0 through 7) and its vector. The second interrupt will remain in the int1 FIFO until the receiving node executes a Read of location 26 HEX which will allow the second interrupt int1 to be issued to the BIM. This method guarantees that all interrupts sent on the link will be serviced and the receiver knows the ID of the node which sent the interrupt it is currently servicing. Only one Read per interrupt is allowed, otherwise loss of subsequent interrupts may occur.

The three interrupt FIFOs are 512 bytes deep so up to 512 interrupts of any level may be queued in the FIFO. A clear function is executed upon a Write to the int ID Register so an interrupt level that has been masked off for some time and contains many global or local interrupts previously sent may be cleared out without servicing them. Only new interrupts received will be serviced. Since the interrupts originally go through the same receive FIFO as data, all data sent before the interrupt will be present in the local node's memory before the interrupt is issued to the local node.

4.7 COMMAND NODE (07H)

This register contains the node ID of the node to receive the interrupt sent by writing the command register. This register must be set at the same time by using a 16-bit word write to both command register and command node or prior to writing the command register. Table 4.7-1 shows node ID patterns.

Table 4.7-1. VMIVME-5576 Node IDs

D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	Node 0
0	0	0	0	0	0	0	1	Node 1
•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	1	1	Node 255

M5576/T4.7-1

4.8 INTERRUPT SENDER ID REGISTERS (26H, 2AH, 2EH)

The three interrupt sender ID registers contain the ID of the node which originated the interrupt currently being serviced. All data sent across the fiber-optic link is tagged with the ID of the originating node so it may be removed from the link once it has been passed around the link one time. The ID is stored in the appropriate register if the data word received is an interrupt. As part of the interrupt handler software, the user must read the appropriate ID register in order to re-arm the currently used interrupt. This process insures that all interrupts sent to the node will be processed. The user may or may not use the ID but it must be read as part of the interrupt handler process.

In the event that a certain interrupt has been masked off at the BIM, the FIFO for that interrupt may be cleared by writing to the ID register for the specific interrupt level before the BIM is armed. The ID write process is to be done in addition to the BIM arming process. In the interrupt handling sequence, the user should do only one read per interrupt cycle. Erroneous results will be caused by multiple reads.

4.9 LOCAL STATUS INTERRUPT

The fourth interrupt on the MC68153 (INT0) is dedicated to generate an interrupt in the event that the local FIFOs become half-full or a corrupt transfer has been received. If the interrupt is not disabled, every time the local VMIVME-5576 is written to and the transmit FIFO is over half full or a transfer error occurs, an INT0 will be generated. The half-full information flag is also available by looking at the CSR. If the transmit FIFO is allowed to become full and the FIFO half full is set, a BERR will be generated when a Write is attempted to the VMIVME-5576. Tables 4.9-1 and 4.9-2 show the architecture of MC68153 registers. The corrupt transfer interrupt may be masked off by removing the mask jumper. This is done so the user will not be bothered by interrupts in redundant transfer mode.

Table 4.9-1. BIM Register Mapping for Interrupts, 0 to 3

$XX23 CONTROL REGISTER INT0 (INT0 - TRANSMIT FIFO OVER HALF FULL)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FLAG	FLAG AUTO CLEAR	VECTOR	INT ENABLE	INT AUTO CLEAR	INTERRUPT LEVEL		
F	FAC	0=INTERNAL 1=EXTERNAL	IRE	1=AUTO 0=NO	L2	L1	L0

$XX27 CONTROL REGISTER INT1 (INT 1 - RECEIVED INTERRUPT FROM OTHER NODES)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FLAG	FLAG AUTO CLEAR	VECTOR	INT ENABLE	INT AUTO CLEAR	INTERRUPT LEVEL		
F	FAC	0=INTERNAL 1=EXTERNAL	IRE	1=AUTO 0=NO	L2	L1	L0

$XX2B CONTROL REGISTER INT2 (INT2 - RECEIVED INTERRUPT FROM OTHER NODES)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FLAG	FLAG AUTO CLEAR	VECTOR	INT ENABLE	INT AUTO CLEAR	INTERRUPT LEVEL		
F	FAC	0=INTERNAL 1=EXTERNAL	IRE	1=AUTO 0=NO	L2	L1	L0

$XX2F CONTROL REGISTER INT3 (INT3 - RECEIVED INTERRUPT FROM OTHER NODES)

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FLAG	FLAG AUTO CLEAR	VECTOR	INT ENABLE	INT AUTO CLEAR	INTERRUPT LEVEL		
F	FAC	0=INTERNAL 1=EXTERNAL	IRE	1=AUTO 0=NO	L2	L1	L0

M5576/T4.9-1

Table 4.9-2. BIM Vector Register Mapping

$XX33 VECTOR REGISTER INT0

VECTOR REGISTER							
V7	V6	V5	V4	V3	V2	V1	V0

$XX37 VECTOR REGISTER INT1

VECTOR REGISTER							
V7	V6	V5	V4	V3	V2	V1	V0

$XX3B VECTOR REGISTER INT2

VECTOR REGISTER							
V7	V6	V5	V4	V3	V2	V1	V0

$XX3F VECTOR REGISTER INT3

VECTOR REGISTER							
V7	V6	V5	V4	V3	V2	V1	V0

M5576/T4.9-2

SECTION 5

CONFIGURATION AND INSTALLATION

5.1 UNPACKING PROCEDURES

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

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

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

5.2 PHYSICAL INSTALLATION

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

DO NOT INSTALL OR REMOVE 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 card guides, slide the board smoothly forward against the mating connector until firmly seated.

5.3 JUMPER INSTALLATIONS

Figures 5.3-1 shows the layout of jumpers on the Reflective Memory board.

J-4 and J-7 are address jumpers. These jumpers must be set to the desired base address of the board as described in paragraphs 5.3.1 through 5.3.3. J3, J5, J8, and J9 must be set to configure other functions of the board as described in paragraphs 5.3.4 through 5.3.7.

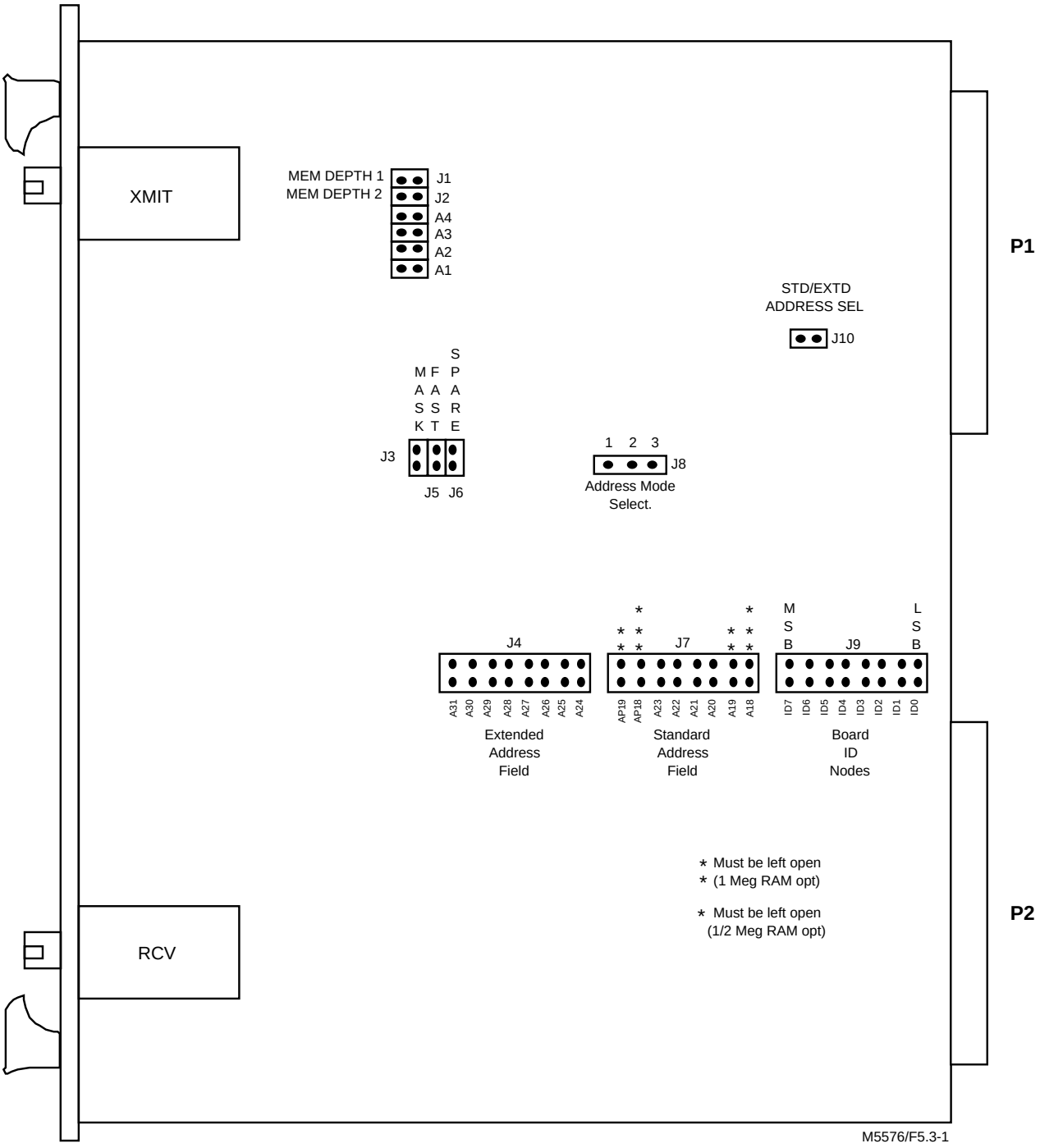


Figure 5.3-1. VMIVME-5576 Jumper Fields

5.3.1 Extended Address Field (J4)

If the extended address jumper, J10, is installed, the extended address match field becomes active. The extended address field selects A31 through A24. An installed jumper is a logic low and no jumper is a logic "one". Refer to Figures 5.3.1-1 and 5.3.1-2.

5.3.2 Address Pass Through Field (J7)

J1 and J2 are configured at the factory and should not need configuration by the user. The information in the rest of this paragraph is for reference only. Refer to Figure 5.3.2-1.

Since the Reflective Memory Board may operate at several different SRAM sizes, the Address field must be able to accommodate the different memory sizes. The Address Pass Through Field performs this function.

If the 1 Meg SRAM option is used, the address field A23 through A20 are valid options for standard address jumper selections. The 512 K SRAM option is the same as the 1 Meg option with the addition of the AP19 pass through jumper. The A19 standard address field jumper may also be used with this option. Refer to Figures 5.3.2-2 and 5.3.2-3.

Since the address pass through jumper pattern is fixed by board memory depth, the field is hard-wired at the factory for the optional memory depth used on the board.

5.3.3 Standard Address Match Field (J7)

The standard address match field is always active. An installed jumper indicates a logic "zero" address. Some jumpers must be left open (no jumper) depending on the board memory options. The Active fields are as follows:

*1 Meg: (A23 through A20) A19 and A18 must be left open (Both Pass Through and Address Field)
512 K: (A23 through A19) A18 must be left open (Both Pass Through and Address Field)
256 K: (A23 through A18)

Jumper fields which must be left open due to memory options specified are left unpopulated on the PCB, i.e., for 512 Kbyte SRAM option, A18 is not used for Standard Address decode and thus, it must be left open. In the 1 Mbyte version, both A18 and A19 address jumpers must be left open.

* No jumper is logic "one".

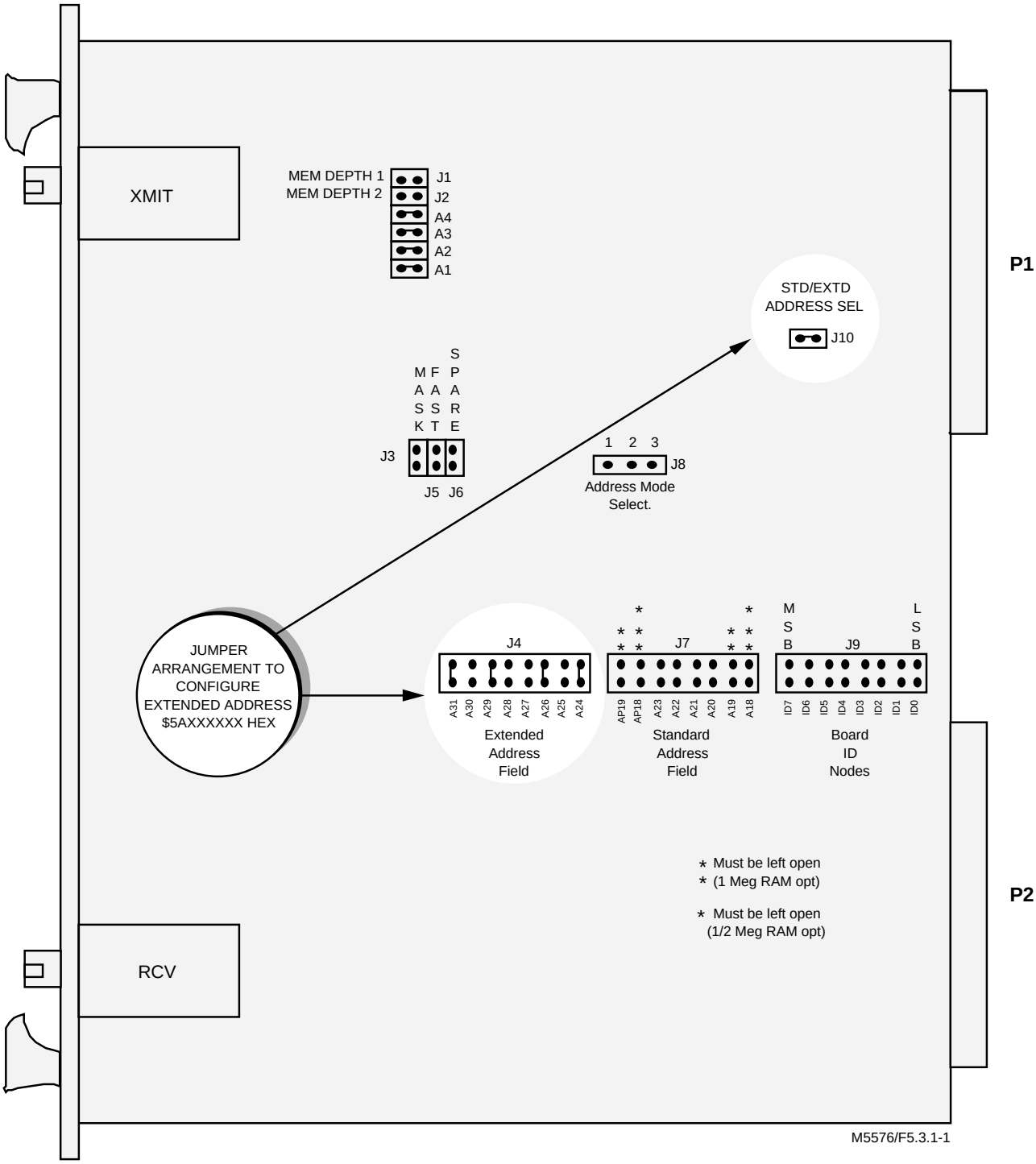


Figure 5.3.1-1. VMIVME-5576 Extended Address Jumper Fields

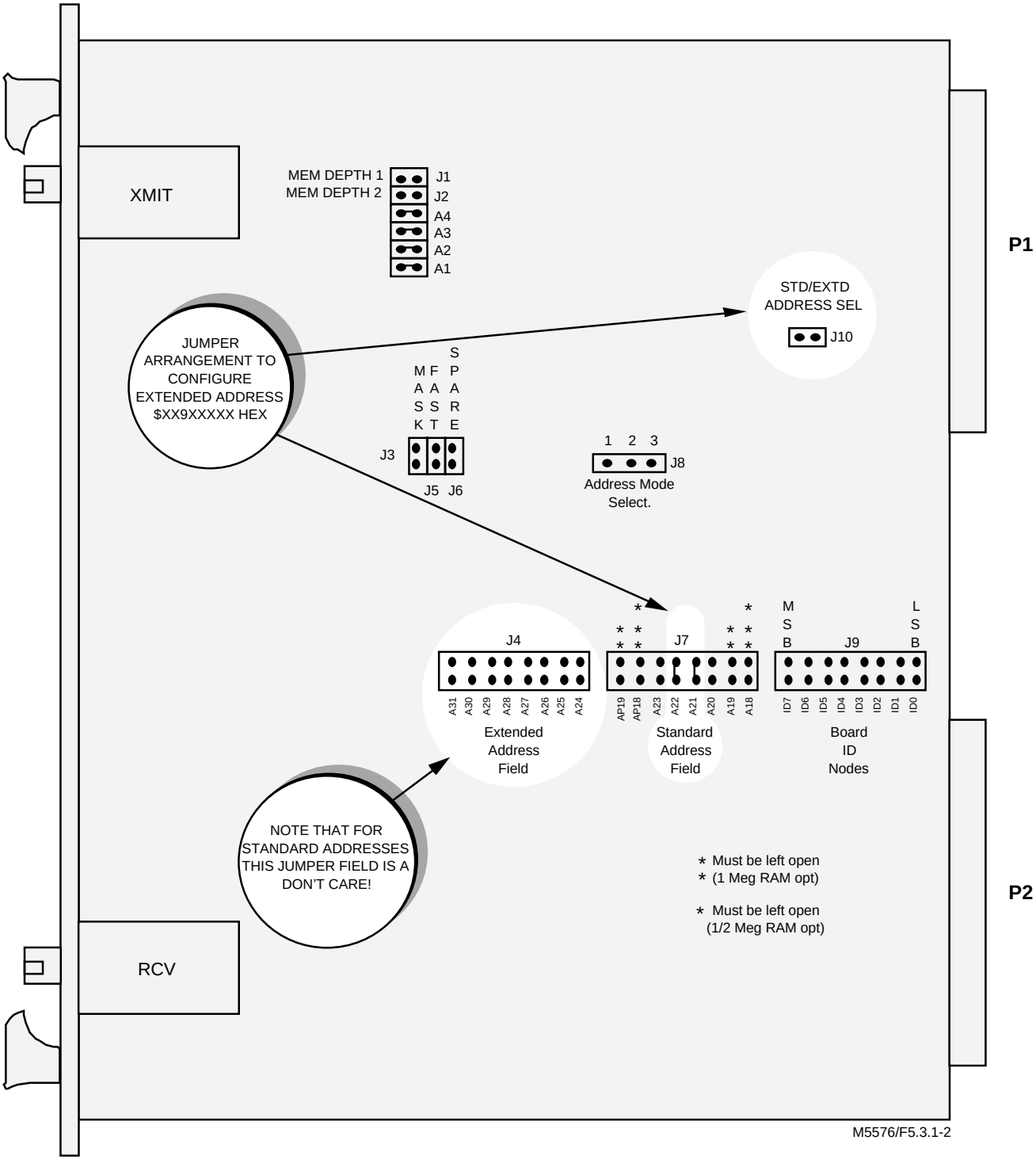


Figure 5.3.1-2. VMIVME-5576 Standard Address Jumper Fields

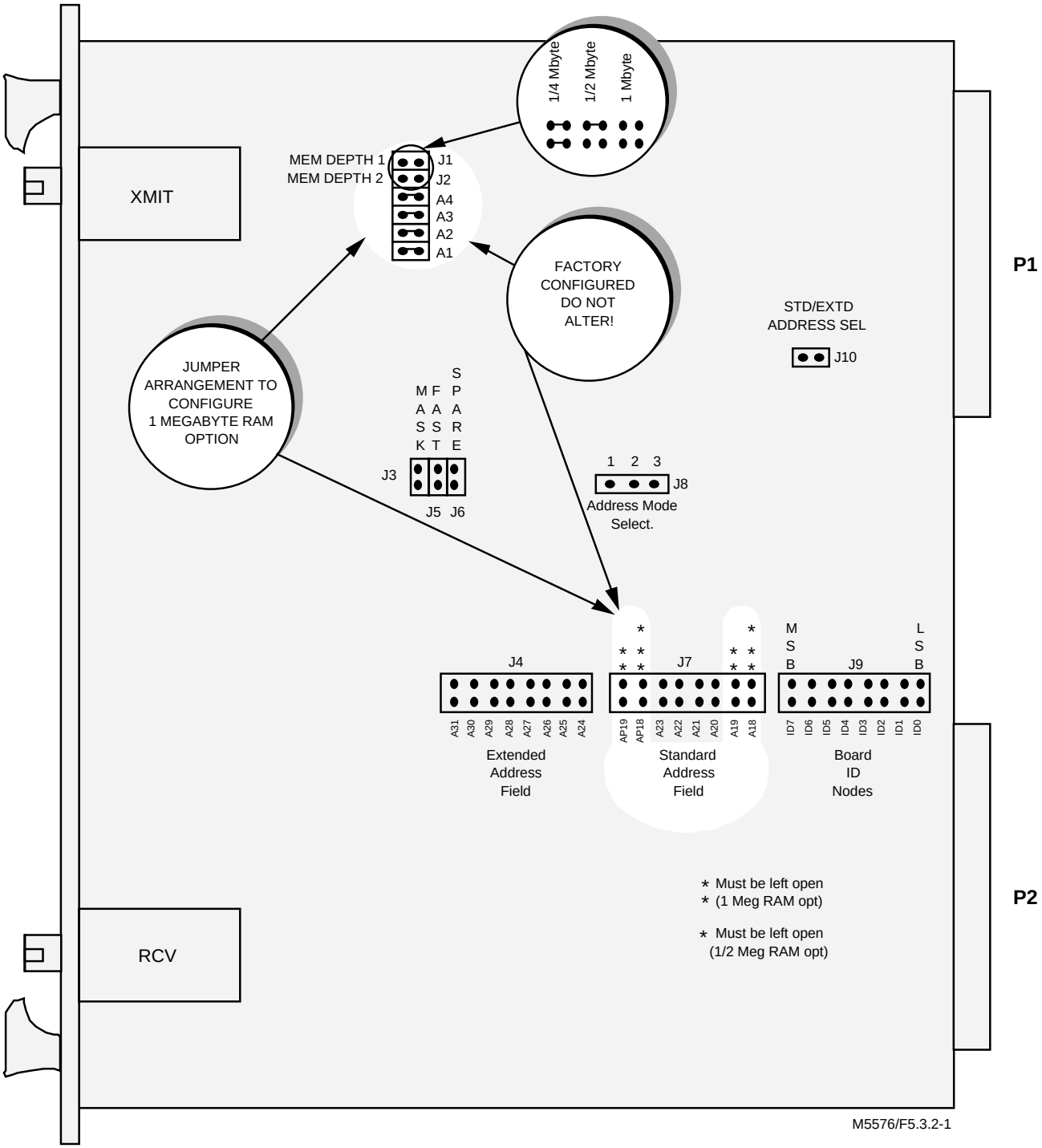


Figure 5.3.2-1. VMIVME-5576 RAM Option Jumper Fields

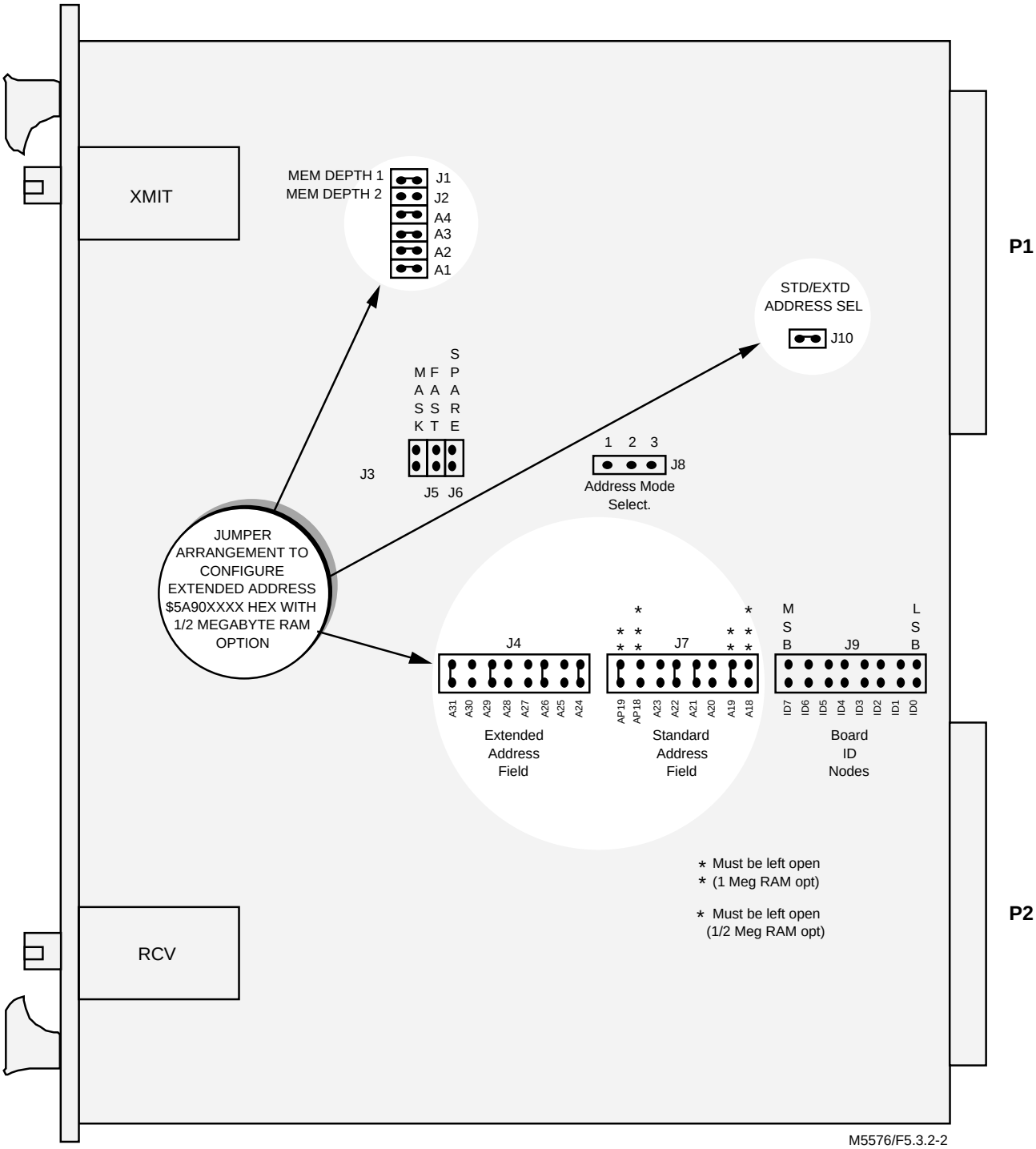


Figure 5.3.2-2. VMIVME-5576 Extended Address with 1/2 Mbyte Option Example

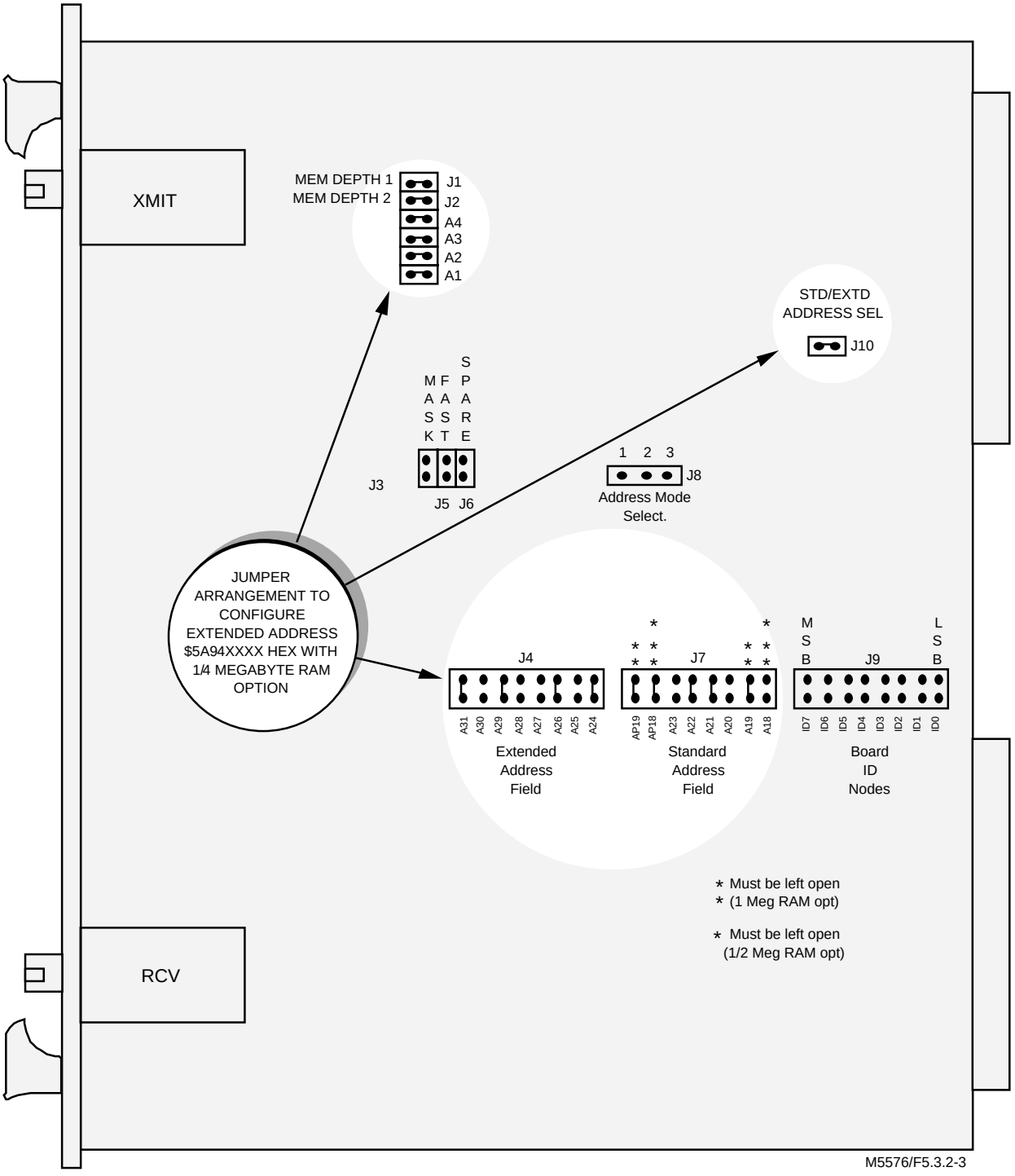


Figure 5.3.2-3. VMIVME-5576 Extended Address with 1/4 Mbyte Option Example

5.3.4 Board Node ID Field (J9)

The Board Node ID field identifies each board on the Memory link. No ID may be used more than once on the link. All nodes on the link should be sequentially numbered starting with 0. Jumper field J9, provides a double hexidecimal digit defining the board node ID. An installed jumper sets the corresponding bit to 0. There is no relation between node number and physical position on the link. Nodes may be physically located in any order. Refer to Figure 5.3.4-1.

5.3.5 Address Modifier Select (J8)

The Reflective Memory Board may operate in one of three modes, supervisory data access, nonprivileged data access, or both. The different options are shown in Figures 5.3.5-1 and 5.3.5-2.

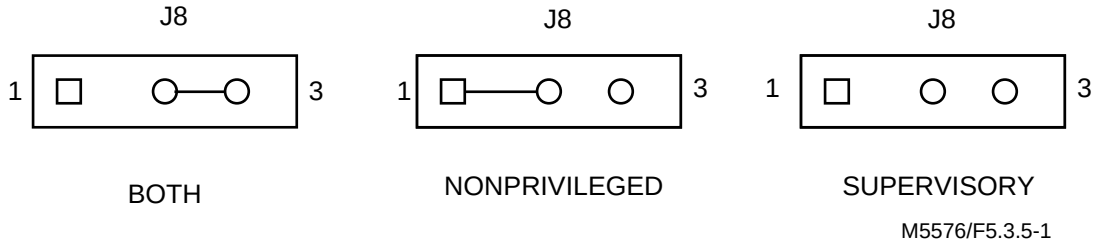


Figure 5.3.5-1. Address Modifier Jumper Options

5.3.6 Fast Field (J5)

If the J5 jumper is left off the fiber-optic link rate is 6.2 Mbytes. If the jumper is present, the link rate is 3.2 Mbytes. The 3.2 Mbytes rate results from transmitting every data word 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.

5.3.7 Mask Field (J3)

If the mask jumper is present, the INT0 on the MC68153 interrupt IC is set when a data error is detected. The CSR can be read to determine if the INT0 was set by the transmitter becoming half-full or by a transfer error. If the mask jumper is removed then no interrupt is generated. In redundant transfer mode, the user does not care if a transfer error occurs on a single transfer since the second transfer statistically is certain to be received correctly. The CSR bit 3 is always set if a single transfer error is detected regardless of the state of the mask jumper.

5.3.8 Spare Field (J6)

Not used.

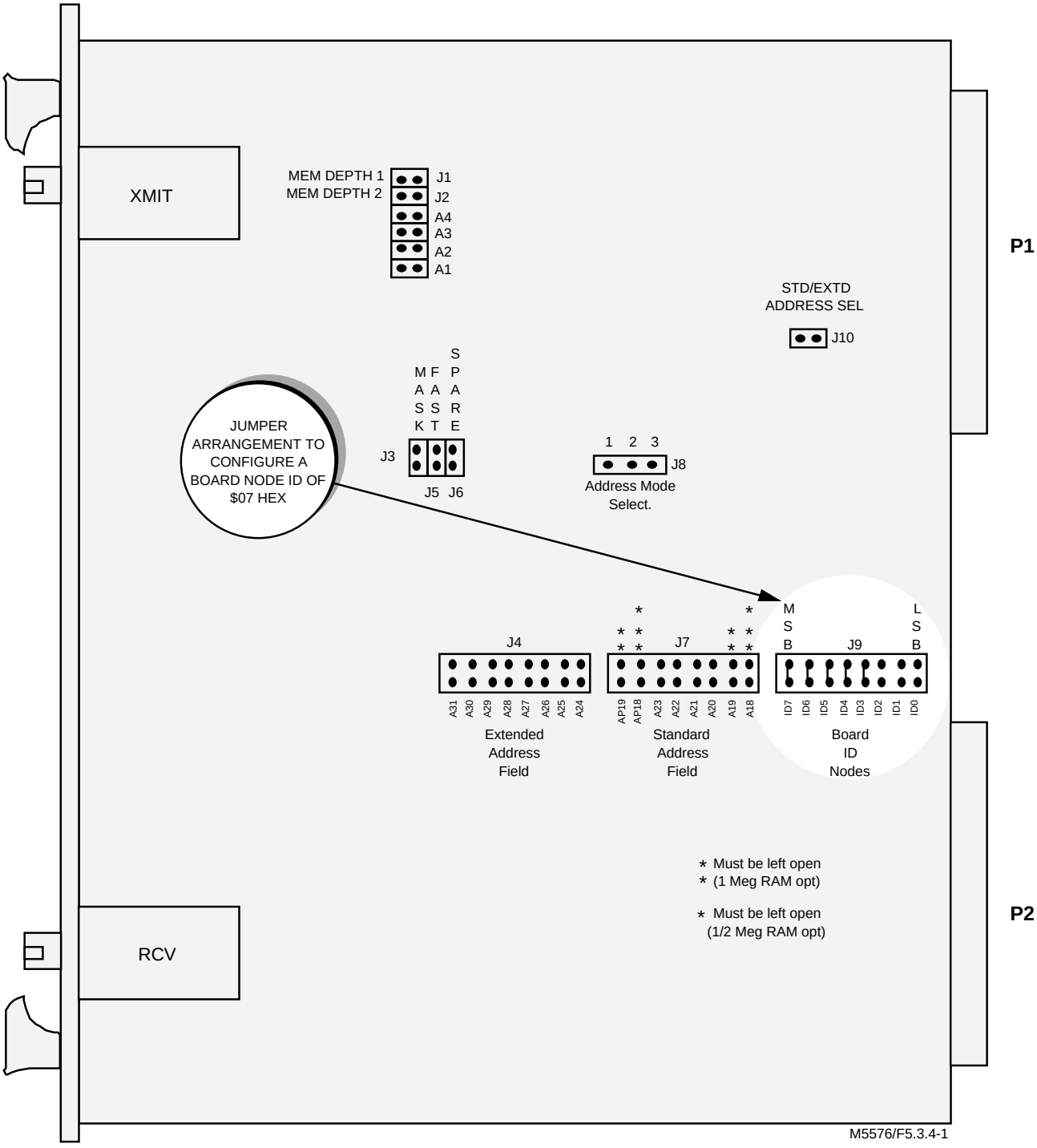


Figure 5.3.4-1. VMIVME-5576 Board Node ID Jumper Field

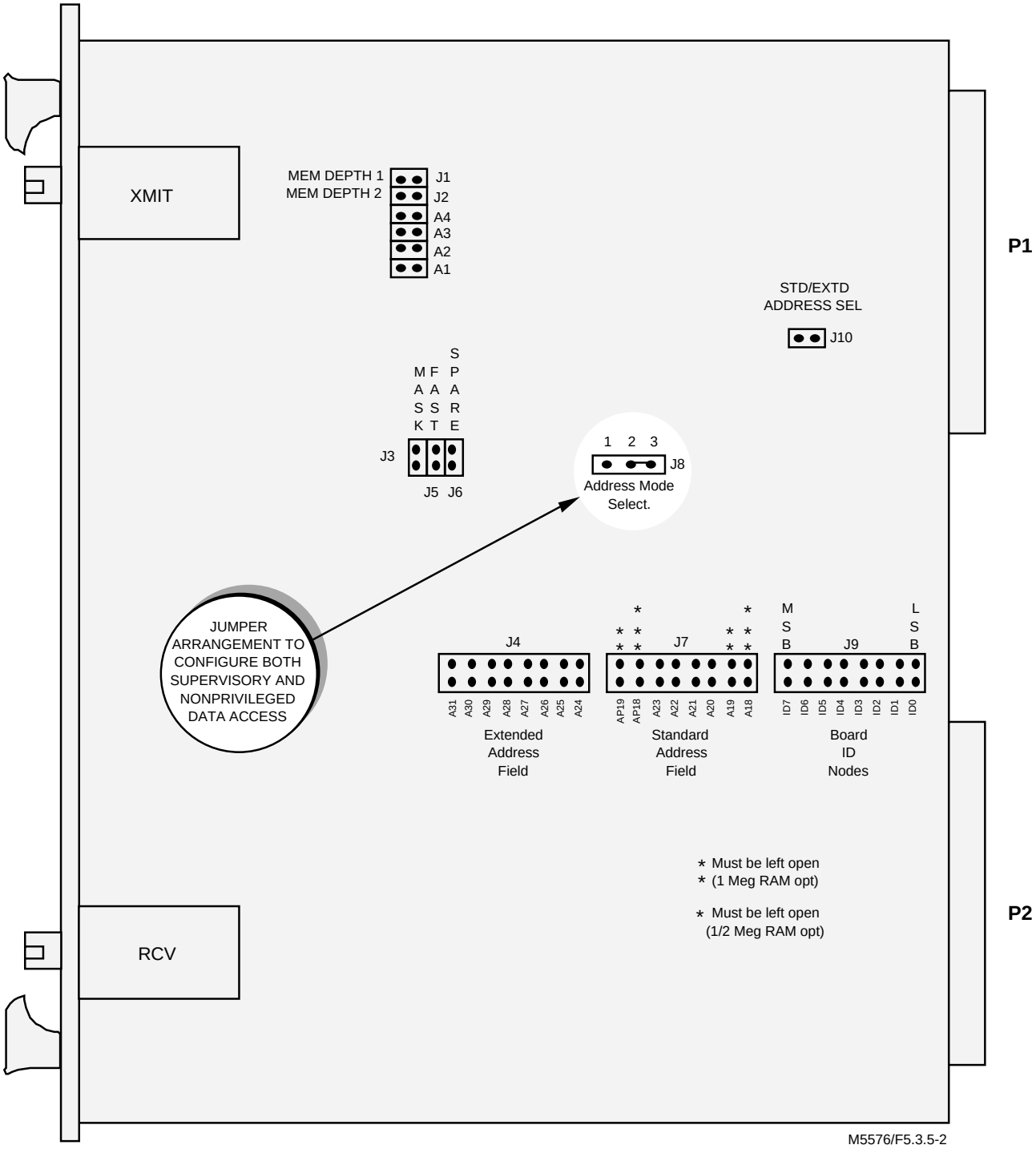


Figure 5.3.5-2. VMIVME-5576 Address Modifier Jumper Field

5.4 FIBER-OPTIC LINK CONFIGURATION

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

When data has been sent around the link and returns to the originating node, two things happen. One, the data is removed from the link. This is done by comparing an ID tag sent with data to the local node ID. If a match is determined, the data is removed. The second event which occurs is that the OWN-ID bit is set in the CSR. Loop data latency can be measured by writing a "zero" to the OWN-ID bit in the CSR and polling until it returns to a "one" 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 will not affect the status of any other nodes CSR. The CSR data write provides a means to measure latency without giving up any memory which may be in use in order to measure data transfer latency.

If data was not generated by the local node, it is placed in the receive FIFO. The receive FIFO then places data in RAM and into the transmit FIFO to be sent to the next node on the link. Data can be mixed from the local VME and the link based on which arrives at what time. Just because two data transfers arrive in one node back-to-back does not guarantee a transfer cannot be inserted between them by the local VME card.

Priority is given to the local VME card in case of a simultaneous access to the RAM and the transmit FIFO by the local VME and the receive FIFO. In any other case, the other must wait for the current cycle to finish to gain access to the RAM.

5.5 IACK DAISY CHAIN

If there are empty slots to the left of the VMIVME-5576, then IACK Jumpers must be installed for the empty slots. Otherwise, the VMIVME-5576 will intermittently fail to respond to VMEbus reads and writes.

SECTION 6

MAINTENANCE

6.1 MAINTENANCE

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

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

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

6.2 MAINTENANCE PRINTS

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

APPENDIX A

ASSEMBLY DRAWING, PARTS LIST, AND SCHEMATIC

APPENDIX B

EXAMPLE CODE

```
/*

     This code written to be down loaded into a Force CPU-33 on a VME bus.
     Compiled using Cross code C compiler, using the VMIC test code library.
     This file should be loaded into the 1st VME chassis CPU, setup as node
     0. This routine will setup and process interrupts then report those
     interrupts received. The second thing this software will do is to
     attempt to fill the output fifo over half full and then process and
     report that interrupt 0 occurred.

*/

#include <stdio.h>
#include <test.h>
#include "5576int.h"

/*
Declare a global pointer to the 5576 board.
*/

Vmic5576 * uut = (( Vmic5576 *)( VME_STANDARD )); /* "VME_STANDARD" defined in
                                                     VMIC Library */
/*
Declare external functions
*/
void isr_int0( void );
void isr_int1( void );
void isr_int2( void );
void isr_int3( void );

/*
global variables for the interrupt routines
*/
int  int0status,int1status,int2status,int3status;

main()
{
   int aa;

   int0status = 0xff;
   int1status = 0;
   int2status = 0;
   int3status = 0;


   printf("\r\n\n Receive Interrupts from remote VME chassis test");

   /*
   **  initialize intr vector table to point to timer ISR
   **
   **  System dependent initialization where our system
   **  is a Force CPU-33 Single Board Computer.
   **
   **  Our method of installation is via a setvect() function
   **  that installs the interrupt service routine address in
   **  the vector table based on the vector chosen.
   **
```

```
**  USER_VECTOR() is a macro in our test library that
**  adds a passed value to the first available
**  user vector for the Force CPU-33 interrupt table.
*/

setvect( USER_VECTOR( 0 ), &isr_int0 );
setvect( USER_VECTOR( 1 ), &isr_int1 );
setvect( USER_VECTOR( 2 ), &isr_int2 );
setvect( USER_VECTOR( 3 ), &isr_int3 );

/*
**  initialize intr vector register to installed ISR
*/

uut->int0vr = USER_VECTOR( 0 );
uut->int1vr = USER_VECTOR( 1 );
uut->int2vr = USER_VECTOR( 2 );
uut->int3vr = USER_VECTOR( 3 );

/*
    Clear interrupt registers of all previous interrupts
*/
uut->int1sid = 0; /* data doesn't matter */
uut->int2sid = 0;
uut->int3sid = 0;

/*
Setup interrupt mode control registers
*/
uut->int0mc = IRQ_LEVEL_7 : INT_ENABLE; /* interrupt 0 enabled at level 7
                                           Auto clear bit 3 low - off,
                                           X/IN bit 5 low - internal  */
uut->int1mc = IRQ_LEVEL_5 : INT_ENABLE; /* interrupt 1 enabled at level 5
                                           Auto clear bit 3 low - off,
                                           X/IN bit 5 low - internal  */
uut->int2mc = IRQ_LEVEL_3 : INT_ENABLE; /* interrupt 2 enabled at level 3
                                           Auto clear bit 3 low - off,
                                           X/IN bit 5 low - internal  */
uut->int3mc = IRQ_LEVEL_1 : INT_ENABLE; /* interrupt 3 enabled at level 1
                                           Auto clear bit 3 low - off,
                                           X/IN bit 5 low - internal  */

printf("\r\n**** Setup waiting on interrupts from node 1."
       "\r\nHit any key to continue after sending interrupts\r\n");

getc();

printf("int0 status = %x, int1 = %x, int2 = %x,"
       "int3 = %x",int0status,int1status,int2status,int3status);

printf("\r\n\n int1 - int3 should = number of node sending interrupt");

printf("\r\n\n ***** sending data to Reflective Memory"
       "\r\n        to generate transmit half full interrupt 0");


    for(aa = 0;aa < 0xfaaaaa;aa += 1)
    {
```

```
        uut->mem5576 = 0xaa;
        uut->mem5576 = 0xaa;
        uut->mem5576 = 0xaa;
        uut->mem5576 = 0xaa;
        }

        printf("\r\nint0 status = %x, int1 = %x, int2 = %x,"
               "int3 = %x",int0status,int1status,int2status,int3status);

   printf("\r\nint0 should be f4 hex, Hit any key to continue");
   getc();

   printf("\r\n\n ***** sending interrupts 1,2,3 to node 1.");
   for (aa = 1;aa < 4;aa += 1)
   {
        uut->cmdnd = 0x01;  /* node one into register 0x07 */
        uut->cmdreg = aa;   /* Send interrupts 1,2,3 to node in reg. 0x07 */
   }
   printf("\r\n\nProgram complete");
}

#pragma interrupt()
void isr_int0( void )
{

   int0status = uut->csr;

   /* wait for transmit fifo to clear. */
   do{
   }while( ( uut->csr & TXFIFO_UHF) == 0 );

}

#pragma interrupt()
void isr_int1( void )
{
   int1status = uut->int1sid; /* read id reg. to clear the interrupt. */
}

#pragma interrupt()
void isr_int2( void )
{
   int2status = uut->int2sid; /* read id register to clear the interrupt. */

}

#pragma interrupt()
void isr_int3( void )
{
   int3status = uut->int3sid; /* read id register to clear the interrupt. */

}
```

```
/*
This is the header file for the VMIVME-5576 Reflective Memory Board
Interrupt test. This file assumes the 2 boards are -200 option.
Both VMIVME-5576 boards are jumpered for Standard Either Mode access.
*/

struct vmivme_5576 {
    union reg_5576 {
        unsigned char  reg_5576_b[64];  /* register space */
        unsigned short reg_5576_w[32];
        unsigned int   reg_5576_l[16];
    } r5576u;
    unsigned char vmivme_5576_mem[1048512];
};

typedef struct vmivme_5576 Vmic5576;     /* 5576 type define */

/* register definitions */

#define     bid       r5576u.reg_5576_b[0x01] /* Board ID */
#define     nid       r5576u.reg_5576_b[0x04] /* Node ID */
#define     csr       r5576u.reg_5576_b[0x05] /* Control & Status Register */
#define     cmdreg    r5576u.reg_5576_b[0x06] /* Command register */
#define     cmdnd     r5576u.reg_5576_b[0x07] /* Command node */
#define     int0mc    r5576u.reg_5576_b[0x23] /* interrupt 0 mode
                                                 control register */
#define     int0vr    r5576u.reg_5576_b[0x33] /* interrupt 0
                                                 vector register */
#define     int1sid   r5576u.reg_5576_b[0x26] /* interrupt 1 sender
                                                 ID register */
#define     int1mc    r5576u.reg_5576_b[0x27] /* interrupt 1 mode
                                                 control register */
#define     int1vr    r5576u.reg_5576_b[0x37] /* interrupt 1
                                                 vector register */
#define     int2sid   r5576u.reg_5576_b[0x2a] /* interrupt 2 sender
                                                 ID register */
#define     int2mc    r5576u.reg_5576_b[0x2b] /* interrupt 2 mode
                                                 control register */
#define     int2vr    r5576u.reg_5576_b[0x3b] /* interrupt 2
                                                 vector register */
#define     int3sid   r5576u.reg_5576_b[0x2e] /* interrupt 3 sender
                                                 ID register */
#define     int3mc    r5576u.reg_5576_b[0x2f] /* interrupt 3 mode
                                                 control register */
#define     int3vr    r5576u.reg_5576_b[0x3f] /* interrupt 3
                                                 vector register */
#define     mem5576   vmivme_5576_mem[0x0]    /* memory */


/* Control Status Register bit define's */


#define  FAIL_LED                0x80
#define  RXFIFO_UHF              0x40  /* receive fifo under half full */
#define  TXFIFO_UHF              0x20  /* Transmit fifo under half full */
#define  TXFIFO_NOT_EMPTY        0x10
#define  BAD_DATA                0x08
#define  OWN_DATA                0x04
```

```
#define  ERR_INT_MASK            0x02
#define  FAST                    0x01

/*  command register bit def's  */

#define  GLOBAL_INT              0x40
#define  INT_1                   0x01
#define  INT_2                   0x02
#define  INT_3                   0x03

/*  bim control bits (for regs. int0mc-int3mc) */

#define  FLAG_BIT                0x80
#define  FLAG_AUTO_CLR           0x40
#define  EXT_VECTOR              0x20  /* set to 0 for internal operation */
#define  INT_ENABLE              0x10
#define  INT_AUTO_CLR            0x08
#define  IRQ_LEVEL_7             0x07
#define  IRQ_LEVEL_6             0x06
#define  IRQ_LEVEL_5             0x05
#define  IRQ_LEVEL_4             0x04
#define  IRQ_LEVEL_3             0x03
#define  IRQ_LEVEL_2             0x02
#define  IRQ_LEVEL_1             0x01
#define  INT_DISABLE             0x00

#define  MEM_OFF                 0x40


/* Board ID register value */

#define     ID_5576      0x18

/* Interrupt FIFO depth in bytes */

#define  INT_FIFO_DEPTH  512

/* Base address pointers */
/*
Reg5578 * regbase_5578 = ((Reg5578 *)(VME_STANDARD + reg_off));*/ /* registers *
/

/*
Memory * membase_5578 = ((Memory *)(VME_STANDARD + mem_off));*/    /* memory *
/
```

APPENDIX C

TROUBLESHOOTING GUIDE

SYMPTOM	POSSIBLE CAUSE
Card/Cards bus error when accessed from VMEbus after multiple writes.	1) Iackin/Iackout daisy chain not in place on one or more nodes on 5550 network. (Iackin must be in place on the backplane even if interrupts are not used on the node) 2) The local node ID is higher than the maximum node ID strapped on Node 0. 3) System throughput has reached maximum and FIFO's have filled up with data to be sent and the transmit FIFO half-full signal has been ignored. 4) P3/P4 cable swapped or lines are open between cable and local node. 5) No Node 0 present to pass token.
Communications lost after power down and up of one or more Nodes on link	1) Receive FIFO's on one or more nodes have been 'glitched' by an out of spec txclk on link. Reset all nodes or issue link reset to nodes strapped to listen to the link reset signal.
Erratic communications on data link	1) Link rate too high for cable length i.e., RATE MAX LENGTH 1 (20 MBYTE/SEC) 50' 2 (10 MBYTE/SEC) 100' 3 (5 MBYTE/SEC) 250' 4 (2.5 MBYTE/SEC) 1000' 2) Open cable at one or more pins on link. 3) Terminator resistors not installed on end nodes of link or extra terminators have been left on center nodes. 4) P3/P4 pins pushed back on Node connector. Inspect pins for damage. 5) P3/P4 cable has crushed pin in Panduit connector. Try swapping P3 cable to P4 and P4 cable to P3 on all nodes. If symptoms of problem change, at least one of the cables is bad.

Node does not answer at expected Address	1) Address pass through for specific memory size not strapped correctly. (See Configuration section of manual.) 2) Address modifier Incorrectly strapped on card 3) Address strapped wrong (Certain LSB address jumpers must be left off because of memory size options).
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 cards are not mapped in the same relative address in relation to the 1 MBYTE boundary then they will not communicate appear to communicate since they contain no common space in RAM. The 2 and 4 MBYTE cards pass a 4 MBYTE relative address and thus must be mapped in the same address relative to 4 MBYTE boundaries. (Note 4 and 2 MBYTE nodes may not be used on the same link as 1 MBYTE or smaller nodes due to the fact that they look at different relative address sizes) 2) The sending node ID is higher than the maximum node ID strapped on Node 0. 3) Open or defective connection to cable P3 or P4 4) Defective FIFO module on either receiving node or transmitting node. (The bad card may be found by checking memory on other nodes to see if they received data correctly.)
Receive FIFO fills up and never empties out even after link traffic stops.	1) Iackin has not been connected to the interrupt Arbiter card. (VME Slot 0 card.)
Data Bits dropped or data appears in wrong address in memory.	1) Damaged or defective FIFO on node 2) Link rate set too high for cable length 3) Damaged cable or connector to link
If data is wrong in a node and is correct in all other nodes	1) Possible write to same memory location in two or more cards at the same time. (This must be prevented in software.)

	2) Damaged local node 3) Link rate set too high for cable length 4) Damaged cable or connection on node with data error.
When interrupts are being used a spurious interrupt is issued from the CPU	1) Iack-in is not daisy chained from the CPU to the reflective memory card on the backplane. VME spec requires all empty VME slots to have Iack-in jumpered to Iack-out. Reflective memory cards expect Iack-in to be driven even if interrupts are not being used. 2) The Iack-in/Iack-out jumper has been placed on backplane for the slot the reflective memory is residing in. This shorts the Iackout driver in the previous slot to the reflective memory Iack-out driver.