MM-6290

VMEbus D64 High Speed Memory Module

Users Guide

9540 Vassar Ave.
Chatsworth, Ca. 91311-4141

Tel:(818) 998-0070
Fax:(818) 998-4459
e-mail: Tech@umem.com

The information in this document is subject to change without notice.

Micro Memory Inc. makes no warranty of any kind with regard to this material. However the information in this document has been carefully checked and is believed to be entirely correct and no responsibility is assumed for inaccuracies.

Any information regarding this product or any other Micro Memory products may be obtained from:

9549 VASSAR AVE.
CHATSWORTH, CA. 91311

TEL: 818/998-0070
FAX. 818/998-4459

CONVENTIONS

This manual uses the following notational conventions.

NOTE	A Note focuses attention on supplementary or unique information and should be read before any action is taken
* or /	Used after a signal names (for example DS0* or DS0/) to indicate that the signal is active low and a signal without an asterisk or slash indicates that the signal is active high.

TABLE OF CONTENTS

FIGURES

TABLES

SECTION I - GENERAL INFORMATION

1.1 INTRODUCTION

This manual describes the elements of operation and installation procedures of the MM-6290D High Speed VMEbus Dynamic Random Access Memory Module, with D64 capability.

1.2 GENERAL DESCRIPTION

The MM-6290D High Speed Memory is compatible with the VMEbus D specification. Inherent to the module are VMEbus options D64, D32, D16 and D8 block transfers and D32, D16 & D8 (32, 16 & 8 bit data path width), and A32, A24 (32, 24 bit address path width) transfers. The module also generates and stores a parity bit for each byte memory location on write cycles and checks this bit for each byte location read on read cycles. If a parity error is detected on a read access, the module sets a Control Status Register (CSR) bit, and may be programmed to assert the bus error signal.

For Extremely High Performance operation, the MM-6290D High Speed Memory is equipped with a High Speed BLOCK Mode Transfer capability. This allows the use of High Performance Block Mode Controllers to transfer up to 256 bytes on the VMEbus, using standard VME block transfer address modifiers, $0B, $0F, $3B, or $3F; or to transfer up to 2048 bytes ($800) using D64 block mode address modifiers, $08, $0C, $38, or $3C.

While the MM-6290D High Speed Memory was designed as a 64 bit wide (data path) memory board, and complies with VMEbus Specifications (Revision D) it can be addressed as 8 bit bytes, 16 bit words, 16 bit or 24 bit unaligned transfers or 32 bit longwords. The MM-6290D Memory can be configured for 8M, and 16M bytes capacity by populating it with industry standard 1 Megabit CMOS DRAM SIP Modules Additional capacity of 32M, and 64M bytes may be obtained by populating it with industry standard 4 Megabit CMOS DRAM SIP Modules. Future expansion is provided for 128Mbytes, and 256Mbytes, using custom 16 Megabit CMOS DRAM SIP Modules.

The MM-6290D High Speed Memory modules are available in several options depending on the capacity required. Table 1 lists the optional part numbers for ordering purposes, the total memory capacity provided by each version.

Table 1 - Memory Capacity Options

Option	Capacity
MM-6290D/8M	8.0M Bytes
MM-6290D/16M	16.0M Bytes
MM-6290D/32M	32.0M Bytes
MM-6290D/64M	64.0M Bytes
MM-6290D/128M	128.0M Bytes
MM-6290D/256M	256.0M Bytes

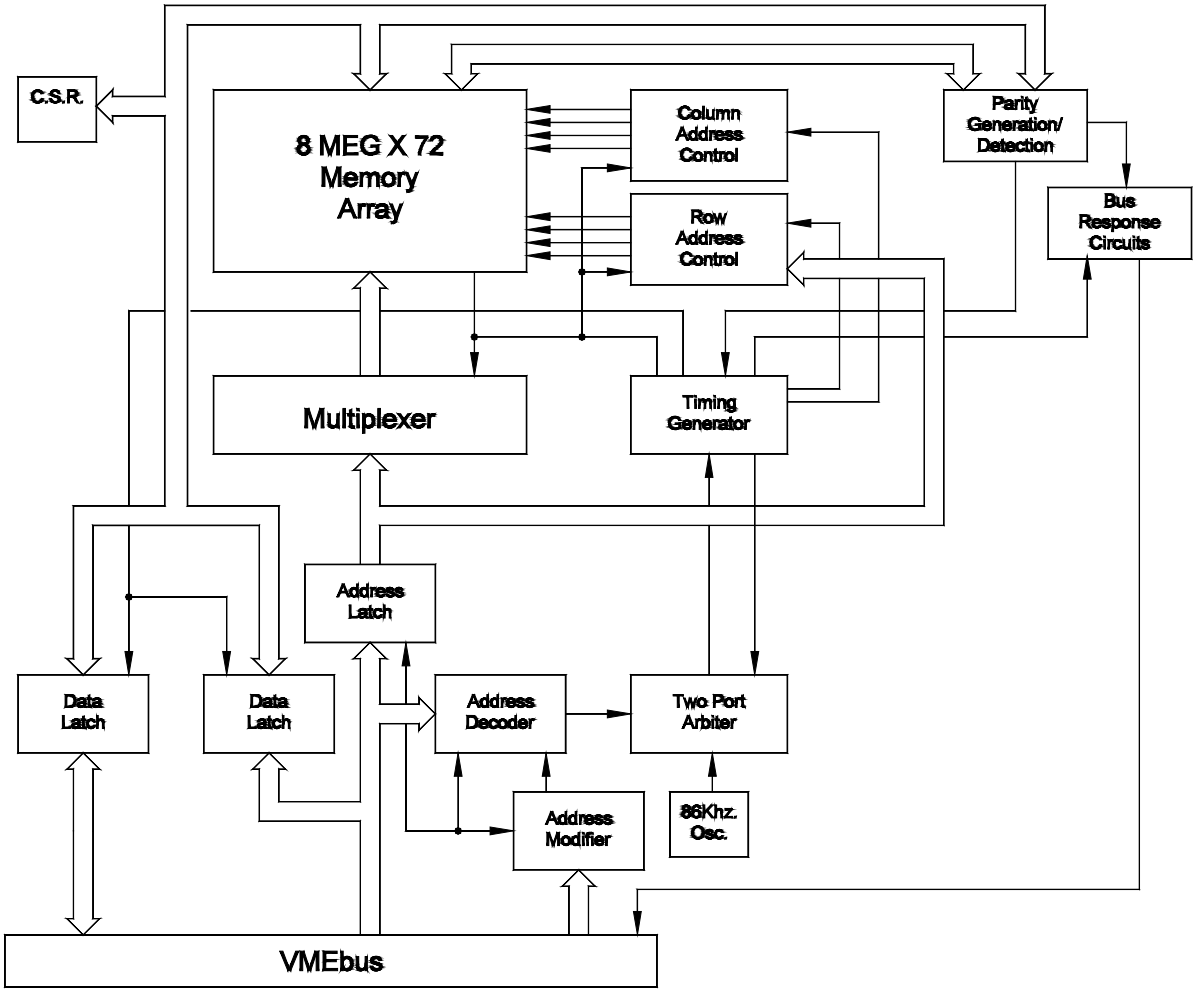


Figure 1 - Functional Block Diagram

1.2.2 Operational Features

The MM-6290D High Speed Memory module contains its own address register and data buffers for total compatibility with the VMEbus Specification (Rev. D).

Module selection for the MM-6290D resides on any 1 Mbyte (100000 hex) boundary (A24/A32). The starting and ending addresses of The MM6290 may be set on any 1 Mbyte boundary within the 4 gigabytes of physical address space, for A32 address modifiers, and on any 1 Mbyte boundary within the 16 megabytes of A24 address space. The A24 ending address is separately selectable so that a portion of the MM6290 may be mapped into the A24 space, for boards which are required to respond to both A24 and A32 address modifiers on the VMEbus.

The MM-6290D High Speed Memory array can be addressed as 8 bit bytes, 16 bit words, or as 32 bit longwords, using conventional Read/Write, Read/Modify/Write (RMW), or 8 bit bytes, 16bit words, 32 bit longwords, or as 64 bit double long words using HIGH Performance BLOCK Transfers (BLT). Unaligned read or write transfers are also supported.

The MM-6290D High Speed Memory responds To various Address Modifier Codes. These include $09, $0A, $0B, $0D, $0E, and $0F for A32 accesses; $08, and $0C for D64/A32 accesses; $29 and $2D for CSR accesses; $39, $3A, $3B, $3D, $3E, $3F for A24 accesses; and $38, and $3C for D64/A24 accesses. The codes are decoded by U1 (a socketed PAL). The user may select independently, for the board to respond to A32, or A24 address modifiers, by means of jumpers E3, and E4. (see 2.5 JUMPER OPTIONS).

The user may select to create his/her own Address Modifier code<s>, in this event the Address Modifier evaluation PAL (U1) can be replaced to accommodate the systems' requirements (see APPENDEX A).

A Control status Register (CSR) is available to allow various dynamic options to be utilized by the software programmer. Included in the CSR is a bit to enable the MM-6290D High Speed Memory to respond to parity error by asserting BERR* instead of DTACK*. Another bit may be read or written to give the parity error status since the last write to this bit or the last bus reset. A jumper option (E1) allows BERR* to be returned on a parity error always. This eliminates the need for the software to manage the CSR. The CSR is an A16 slave and has separate address selection switches and may be addressed on any 2 byte boundary in the VMEbus I/O address space. Valid data is only driven on the odd byte of this two byte location. (See SECTION III - PROGRAMMING INFORMATION for programming information)

1.3 GENERAL SPECIFICATIONS

Table 2 - General Specifications

Characteristics	Specifications	
Capacity	8M, 16M, 32M or 64M bytes	
Address	32 bits A32/A24/BLT/UAT VMEbus |	
Data In/Data Out	8/16/24/32/64 bits bidirectional with three-state output D08/D16/D32/D64 BLT	
Modes of Operation	Read, Write, Read/Modify/Write (RMW), Block Transfers (BLT), Un-aligned (UAT)	
Address Modifiers	6 bits, jumper selectable, or user programmable PAL	
Error Detection	Even Parity, one parity bit generated/checked for each byte.	
Module Selection	Memory selected on 1 Megabyte	
VMEbus	($100000) boundary within the 4G byte address space, for A32 address modifiers; and any 1 Megabyte ($100000) boundary within the 16M byte address space for A24 address modifiers.	
Control Status Register	CSR Selected on any 2 byte Selection in the VMEbus 'A16' I/O Address space.	
Interface:		
Inputs	TTL-compatible	
Outputs	48ma. Three-state, TTL-compatible	
Operating Temperature	0 to 60 degrees C	
Storage Temperature	-40 to +85 degrees C	
Relative Humidity	95% without condensation	
Power Requirements:	Standby	Operate
	+5V (fully-populated)	2.6A 4.0A

Table 3 - Timing Specifications of the Dynamic Ram Memory Module

Cycle Type	Cycle Time	Access Time
Write	200 nsec.	60 nsec.
Write	130 nsec.	50 nsec.
(Block Transfer)		
Write	150 nsec.	70 nsec.
(First Cycle of Block Transfer)		
Read	250 nsec.	150 nsec.
Read	180 nsec.	80 nsec.
(Block Transfer)		
Read	240 nsec.	140 nsec.
(First Cycle of Block Transfer)		

Table 4 - P1 Connector VMEbus Pin Assignments

PAGE NUMBER	ROWa SIGNAL MNEMONIC	ROWb SIGNAL MNEMONIC	ROWc SIGNAL MNEMONIC
1	D00	** BBSY*	D08
2	D01	** BCLR*	D09
3	D02	** ACFAIL	D10
4	D03	BG0IN*	D11
5	D04	BG0OUT*	D12
6	D05	BG1IN*	D13
7	D06	BG1OUT*	D14
8	D07	BG2IN*	D15
9	GND	BG2OUT*	GND
10	** SYSCLK	BG3IN*	** SYSFAIL*
11	GND	BG3OUT*	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	GND	BR3*	A23
16	DTACK*	AM0	A22
17	GND	AM1	A21
18	AS*	AM2	A20
19	GND	AM3	A19
20	IACK*	GND	A18
21	IACKIN*	** SERCLK(1)	A17
22	IACKOUT*	** SERDAT*(1)	A16
23	AM4	GND	A15
24	A07	** IRQ7*	A14
25	A06	** IRQ6*	A13
26	A05	** IRQ5*	A12
27	A04	** IRQ4*	A11
28	A03	** IRQ3*	A10
29	A02	** IRQ2*	A09
30	A01	** IRQ1*	A08
31	** -12v	** +5vSTDBY	** +12v
32	+5v	+5v	+5v

** - No Connection on MM6290D High Speed Memory

Table 5 - P2 Connector VMEbus Pin Assignments

PAGE NUMBER	ROWa SIGNAL MNEMONIC	ROWb SIGNAL MNEMONIC	ROWc SIGNAL MNEMONIC
1	N/C	+5v	N/C
2	N/C	GND	N/C
3	N/C	** RESERVED	N/C
4	N/C	A24	N/C
5	N/C	A25	N/C
6	N/C	A26	N/C
7	N/C	A27	N/C
8	N/C	A28	N/C
9	N/C	A29	N/C
10	N/C	A30	N/C
11	N/C	A31	N/C
12	N/C	GND	N/C
13	N/C	+5v	N/C
14	N/C	D16	N/C
15	N/C	D17	N/C
16	N/C	D18	N/C
17	N/C	D19	N/C
18	N/C	D20	N/C
19	N/C	D21	N/C
20	N/C	D22	N/C
21	N/C	D23	N/C
22	N/C	GND	N/C
23	N/C	D24	N/C
24	N/C	D25	N/C
25	N/C	D26	N/C
26	N/C	D27	N/C
27	N/C	D28	N/C
28	N/C	D29	N/C
29	N/C	D30	N/C
30	N/C	D31	N/C
31	N/C	GND	N/C
32	N/C	+5v	N/C

** - No Connection on MM6290D High Speed Memory

1.4 INTERFACE TIMING

The VMEbus Interface Timing Diagrams (Figure 2 and Figure 3), illustrate the timing relationships on the Data Transfer bus, for Write, and Read Cycles. Access delay time is measured from the leading edge of the data strobes, to the leading edge of DTACK*.

The refresh cycle timing is the same as read cycles. Refresh cycles are repeated once every 10usec. The cycle interval timing is generated by a free running oscillator which provides a refresh transparent to memory access cycles.

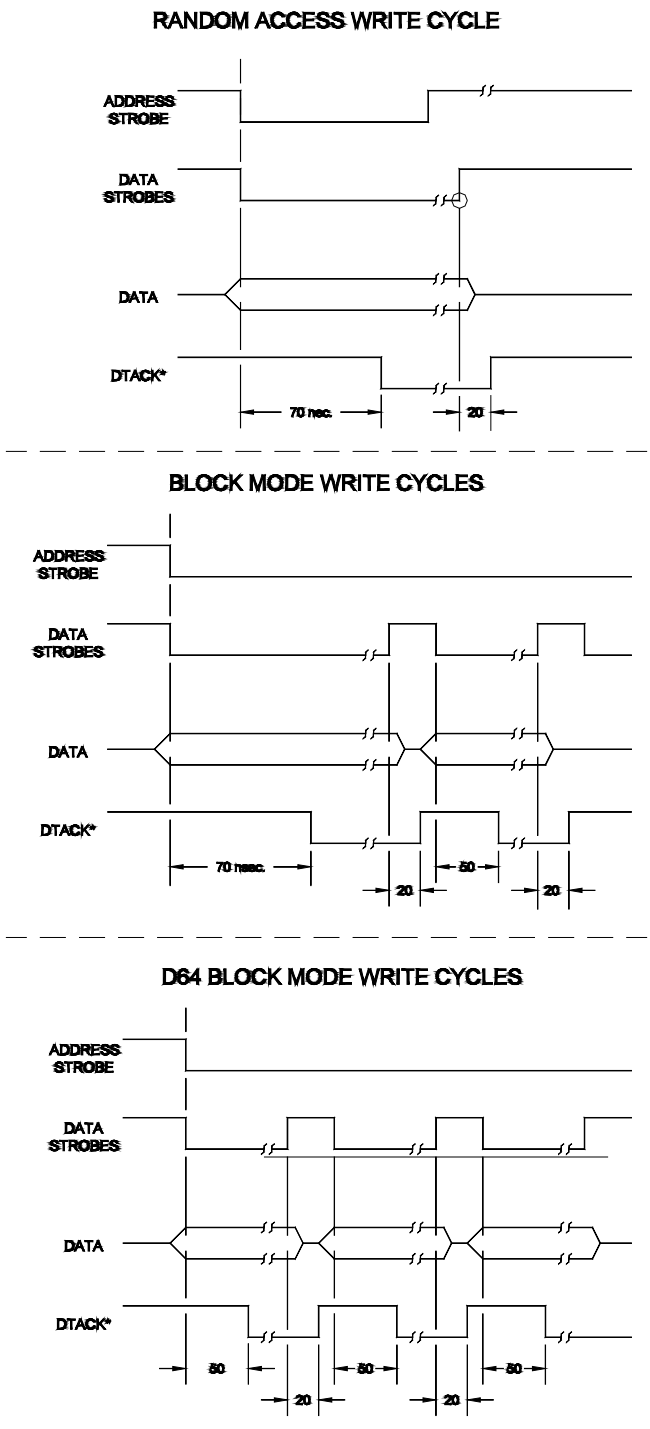


Figure 2 - VMEbus Write Timing Diagram

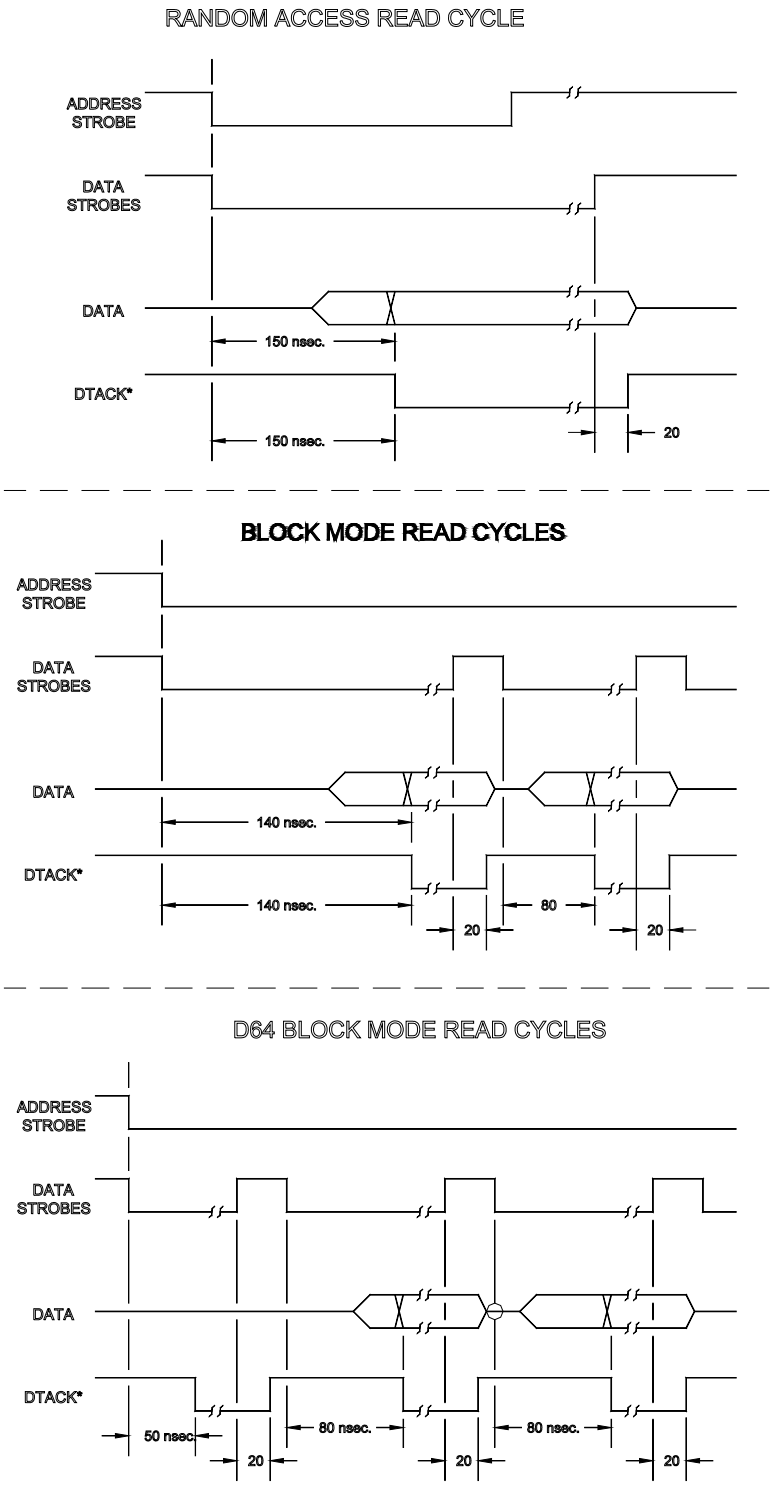


Figure 3 - VMEbus Read Timing Diagram

1.5 MEMORY ARRAY ORGANIZATION

The memory array is partitioned into two 4096K x 72 bit segments. The minimum memory configuration would be 1024K x 72 bits, including the eight parity bit DRAMs in segment zero. This would translate to 8 Megabytes minimum configuration for the MM-6290D Memory.

Although the MM-6290D Memory is partitioned as two blocks of 72 bits each (64 plus 8 parity), The system will only see 64 bits of data during D64 block memory references, since The parity bits are generated or checked transparent to the user. The 64 bits of data are divided into 8 bytes (8 bits each). Memory reads and writes may be 8, 16, 24, 32 or 64 bits wide.

Determining which bits constitute what byte may be somewhat confusing. This can be cleared up by seeing the same data in the various formats used in most systems (IE: byte, word, longword, and Double longword). If the contents of the first 8 locations in memory are $01 $23 $45 $67 $89 $AB $CD $EF they would be seen in various formats as shown below.

Figure 4 - Memory Array Organization

BYTE Format		WORD Format		LONGWORD Format		DOUBLE LONGWORD Format	
Location	Data	Location	Data	Location	Data	Location	Data
$00000000	$01	$00000000	$0123	$00000000	$01234567	$00000000	$0123456789ABCDEF
$00000001	$23	$00000002	$4567	$00000004	$89ABCDEF		
$00000002	$45	$00000004	$89AB				
$00000003	$67	$00000006	$CDEF				
$00000004	$89						
$00000005	$AB						
$00000006	$CD						
$00000007	$EF						

BIT Organizations			
DOUBLE LONGWORD			
6 4	4 3	3 1	1 0
3.....8	7.....2	1.....6	5.....0
LONGWORD			
3 1	1 0	3 1	1 0
1.....6	5.....0	1.....6	5.....0
WORD			
1 0	1 0	1 0	1 0
5.....0	5.....0	5.....0	5.....0
BYTE			
0 00 0	0 00 0	0 00 0	0 00 0
7.....07.....0	7.....07.....0	7.....07.....0	7.....07.....0
0000000100100011	0100010101100111	1000100110101011	1100110111101111
$0 $1 $2 $3	$4 $5 $6 $7	$8 $9 $A $B	$C $D $E $F

SECTION II - INSTALLATION

2.1 INTRODUCTION

This section details a step by step procedure to interface the MM-6290D Memory to the VMEbus.

2.2 UNPACKING INSTRUCTIONS

Unpack module from shipping carton. If carton is damaged upon receipt, request that the carrier's agent be present during unpacking and inspection of the equipment. Refer to packing list and verify that all items are present. Save packing material for storing or reshipping the equipment. For repairs or replacement of a board damaged during shipment, contact MICRO MEMORY, Inc. to obtain a Return Authorization number and further instructions.

2.3 HARDWARE PREPARATION

The memory module should be inspected and prepared for jumper placement prior to system installation. The following sections describe the proper jumper options necessary for system operation (see 2.5 JUMPER OPTIONS, for jumper options).

2.3.1 Handling Procedure

The MM-6290D High speed Memory uses cMOS components that are susceptible to damage if exposed to static electrical charges. To avoid damage of these components during handling, testing or operation, the following procedure should be used.

A. Device leads should contact conductive material to avoid building of any static charge, except during testing or operation.

B. Soldering iron tips, metal fixtures, tools, and handling facilities used in preparing the module for operation should be grounded.

C. Devices should never be removed or inserted while power is applied to the module because voltage transients may permanently damage the devices and/or module.

D. The memory module should never be plugged in or out of the cardcage while power is applied.

E. External signals should not be applied to device inputs while power is removed.

F. Any memory module removed from the system should be transferred to either non-conductive foam or an anti-static plastic bag for storage or shipment.

2.4 OPERATING ENVIRONMENT

The MM-6290D High Speed Memory is an extremely High density memory product. Containing, 64 bits of high current (48ma.) data bus drivers, numerous high speed components. And as a result, the environment into which the MM-6290D Memory is placed, must be carefully considered.

2.4.1 Airflow and Cooling

Due to the extremely high component density of the MM-6290D Memory, an adequate airflow is required to maintain the operating temperature within specifications of the memory module. Failure to adhere to these requirements may result in poor long term reliability.

2.4.2 Power Requirements

The power requirements of MM-6290D High Speed Memory are such that the module should never be operated while powered only by The P1 connector. Hence power should be applied to the appropriate pins (See Table 4 and Table 5) of both the P1 and the P2 connector prior to operation of the MM-6290D Memory.

2.5 JUMPER OPTIONS

There are several jumper options available on the MM-6290D Memory module, depending on the user application. All of the available options are selected by installing or removing mini-jumpers on wire-wrap pins on 0.100 inch centers. (see **Error! Reference source not found.**)

2.5.1 Address Modifier Evaluation/Selection

The MM-6290D High Speed Memory evaluates the Address Modifiers from the VMEbus to determine if they are to be responded to. The MM-6290D Memory may be configured by jumpers to respond to EXTENDED A32 address modifier codes (where all 32 bits are decoded for module selection) or to STANDARD A24 address modifier codes (where only The lower 24 address bits are decoded for module selection), or to both sets of address modifier codes. When jumper E4 is removed, the module will respond to A32 address modifier codes, and when jumper E3 is removed, the module will respond to A24 address modifier codes. As shipped from the factory, the MM-6290D Memory E4 is removed and E3 is installed, causing it to respond to EXTENDED A32 address Modifier codes. These address Modifier codes are described in appendix A. The user may select to replace the Address Modifier selection PAL (U1), to allow for some other combination or the creation of a special (user defined) code or codes. The MM6290 does not respond to A64 address modifier codes.

Table 6 Address Modifier Enable Jumpers

E4	E3	MM6290 Responds to
Installed	Installed	No memory accesses
Installed	Removed	A24 address modifiers only
Removed	Installed	A32 address modifiers only
Removed	Removed	A24 & A32 address modifiers

2.5.2 Memory Device Size Selection (Factory, 4Mbit)

The MM-6290D High Speed Memory is designed to accommodate either 1 Mbit dRAMs, 4Mbit DRAMs, or (when available) 16Mbit DRAMs. If the board is only half populated, regardless of the size of DRAMs used, E5 should be jumpered between positions 2 and 3, and E6 should not be jumpered. When using 1 Mbit DRAMs, jumpers should be installed between positions 2 and 3 of jumper block E6 and E5 should not be jumpered.

When using 4 Mbit DRAMs, E6 should be jumpered between positions 1 and 2, and E5 should not be jumpered. When using 16 Mbit DRAMs, E5 should be jumpered between positions 1 and 2, and E6 should not be jumpered.

2.5.3 Parity Error LED Latch on Error (Factory installed)

The MM-6290D High speed Memory has a parity error indicator lED, which will be turned on by a parity error during a read access. If jumper E2 is installed, it will remain on until a bus reset, or reset by writing a 0 to the CSR, bit 7, if the CSR is enabled. (see section 2.5.4, below) If jumper E2 is removed, The LED will be turned off by the next write access or next read access without a parity error. See Section 2.6.4 for cSR selection and Section 3.1 for cSR bit descriptions.

2.5.4 CSR Enable (Factory, CSR enabled)

If jumper E1 is removed, the CSR is enabled. A bus reset causes all bits in the CSR to be reset. Writing a 1 to bit 0 of the CSR causes a BERR* to be generated for any read cycle where a parity error is detected. Bits 1 through 6 may be written and read back, but have no effect on the MM6290 Memory Board. Bit 7 is set whenever a parity error is detected during a read cycle, regardless of the condition of bit 0. Writing a 0 to bit 7 resets it, and if the CSR is enabled, also resets the parity error LED. If jumper E1 is installed: BERR* is always returned for read accesses where a parity error is detected, and the parity error LED may only be reset by a bus reset.

NOTE: See 2.6.2 Control / Status Register Address Selection for CSR selection.

2.6 ADDRESS SWITCH SETTINGS

The MM-6290D High Speed Memory contains seven (7) major switches that must be configured prior to installation before proper operation can be obtained. (See Figure 5 - Board Layout for switch locations) These are sW1 through sW7. A closed switch position represents a 0, and an open switch position represents a 1.

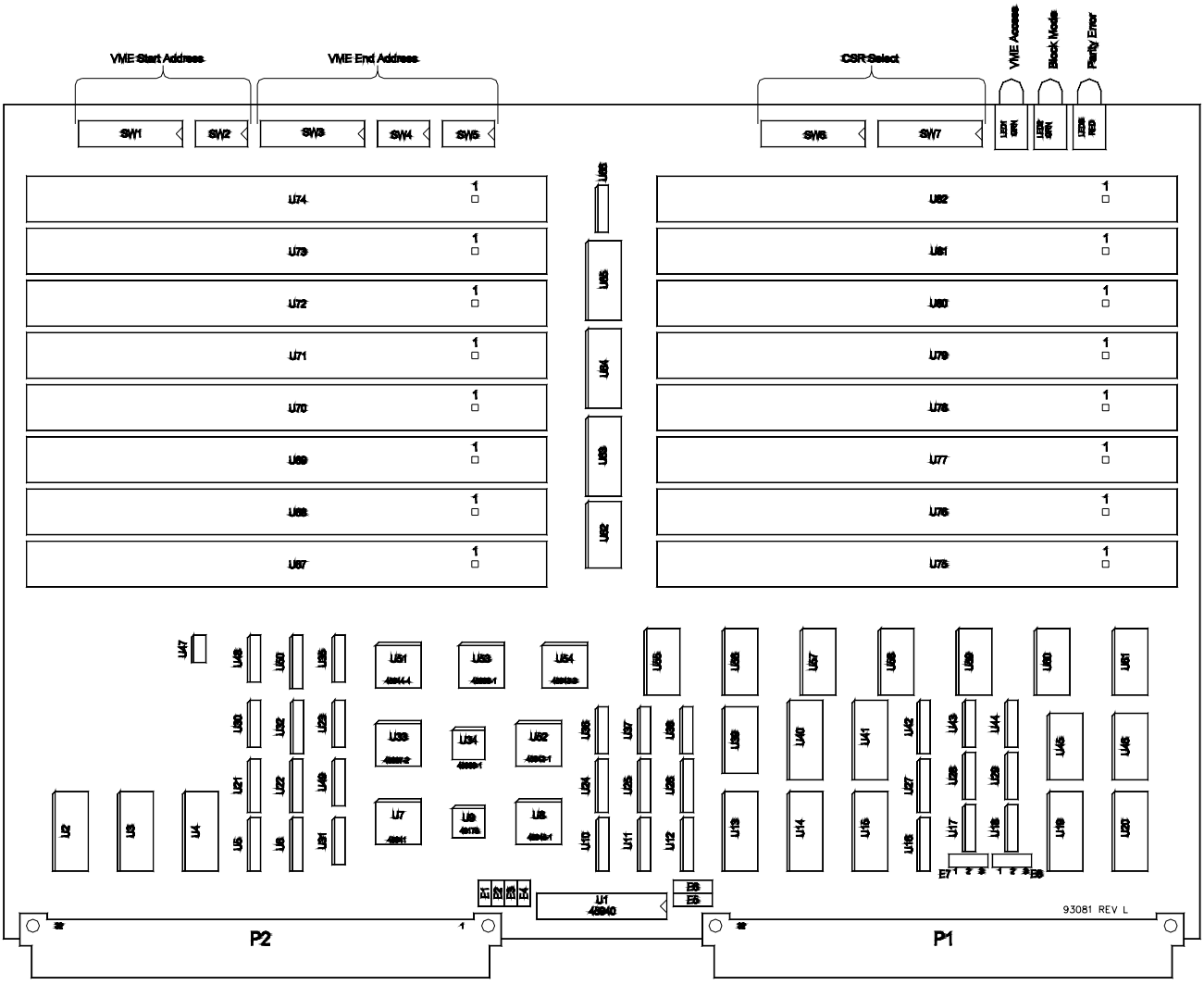


Figure 5 - Board Layout

2.6.1 VMEbus Module Selection

Module selection for the MM-6290D Memory resides on 1024K ($100000) byte boundaries. selection of the VMEbus lower limit and upper limit on a particular 1024 kilobyte ($100000) boundary within the 4 GigaByte memory space is accomplished by closing or opening the appropriate switch positions in switches 1 through 5.

2.6.1.1 Address Selection for A32 Accesses

For A32 address modifiers, the starting address is set by the twelve bits of switches 1 and 2. Position 8 of switch 1 is the most significant bit, and corresponds to A31; position 1 of switch 2 is the least significant bit, and corresponds to A20. The ending address is set by the twelve bits of switches 3 and 4; position 8 of switch 3 is the most significant bit and corresponds to A31; position 1 of switch 4 is the least significant bit, and corresponds to A20. On all switch positions, an open corresponds to a 1, while a closed position corresponds to a 0.

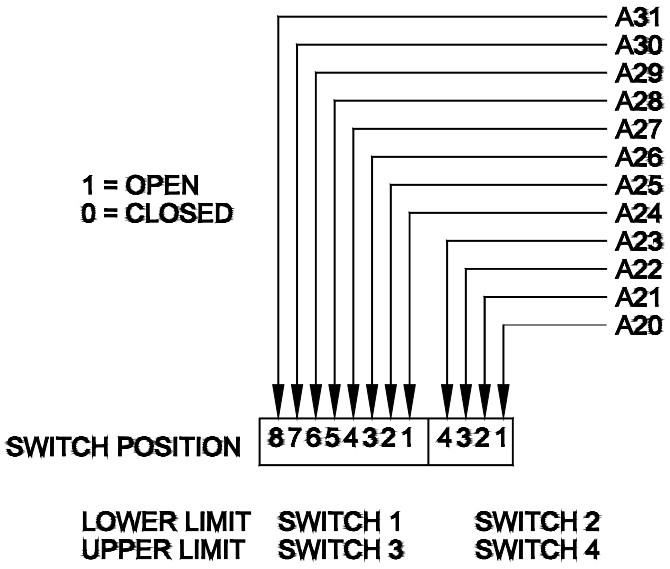


Figure 6 - Upper & Lower Address Selection for A32 Address Modifiers

The above address switch arrangement can be thought of as three hex digits for the starting address, with the most significant digit being switch 1, positions 8-5; the next most significant digit being switch 1, positions 4-1; and the least significant digit being switch 2, positions 4-1. The three hex digits for the ending address are: The most significant digit is switch 3, positions 8-5; The next most significant digit is witch 3, positions 4-1; and the least significant digit is switch 4, positions 4-1.

2.6.1.2 Address Selection for A24 Accesses

For A24 address modifiers, The starting address is set by the same four bits of switch 2 which form the lower four bits of the A32 starting address. Position 4 is the most significant bit, and corresponds To A23; position 1 is the least significant bit, and corresponds to A20. In order to accommodate some processors which have on-board memory in the upper portion of the A24 address space, the ending address for A24 accesses is set by a separate switch from the
A32 ending address. Switch 5 sets the A24 ending address. Position 4 is the most significant bit, and corresponds to A23. Position 1 is the least significant bit, and corresponds to A20.

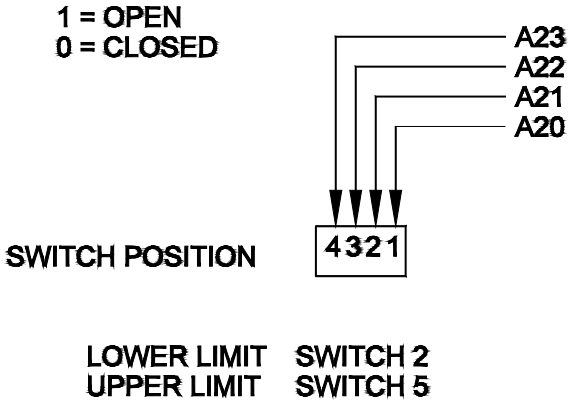


Figure 7 - Upper & Lower Address Selection - A24

This may be thought of as a single hex digit for the starting address, which is switch 2, positions 4-1; and a single hex digit for the ending address, which is switch 5, positions 4-1.

The fact that the ending address for A24 accesses is separate from the ending address for A32 accesses allows A24 and A32 bus masters to access a common portion of memory of the MM6290 memory. For example, if an A24 disk controller, and an A32 processor were accessing an MM6290, where the starting address was $00800000; the A24 ending address was $FFFFFF; and the A32 ending address was $017FFFFF, Then both masters could access the first eight Mbytes of the memory, but only the processor could access the last eight Mbytes of the memory.

2.6.2 Control / Status Register Address Selection

Selection of the CSR address may be on any 2 byte boundary, (ie: $0000, $0002, ..., $FFFE), within the VMEbus I/O space, by closing or opening the appropriate positions on switches 6 and 7. Switch 6, position 8 corresponds to A15, and position 1 corresponds to A08. Switch 7, position 8 corresponds to A07, and position 2 to A01. This arrangement could be viewed as four hex digits.

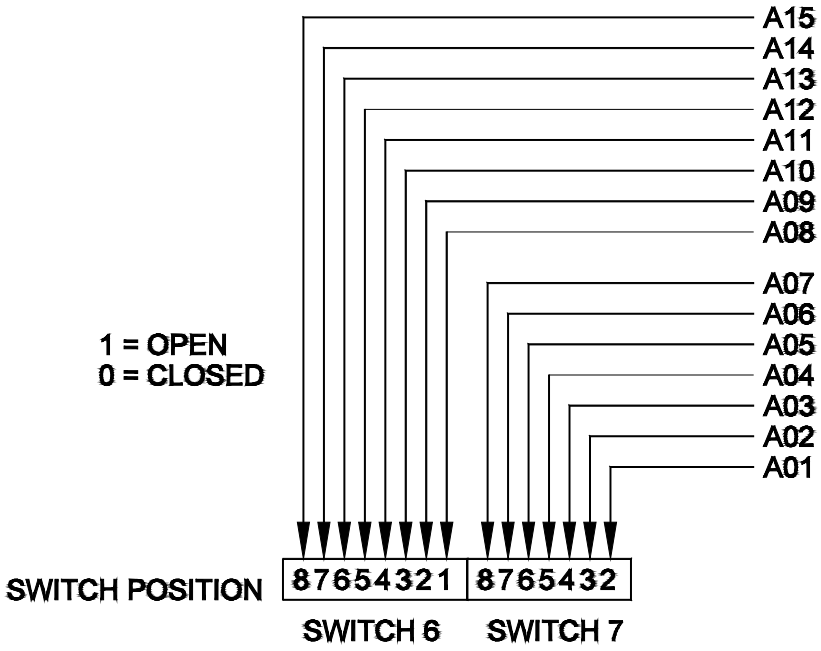


Figure 8 - Control/Status Register Address Selection

2.6.3 Example Addressing

To address the MM-6290D High Speed Memory, the user must determine first where the MM-6290D Memory is to be located. In this example I have chosen from 00100000 to $010FFFFF for the VMEbus A32 accesses and from $100000 to $EFFFFF for the VMEbus A24 accesses. And an address of $0404 (VMEbus I/O space) for the CSR.

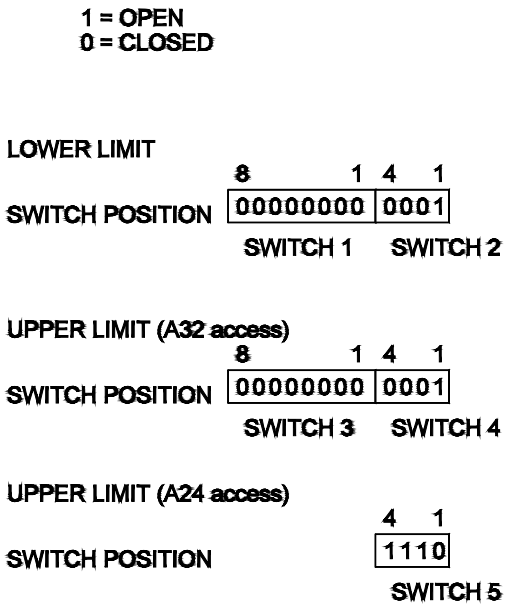


Figure 9 - Example of VMEbus Address Selection

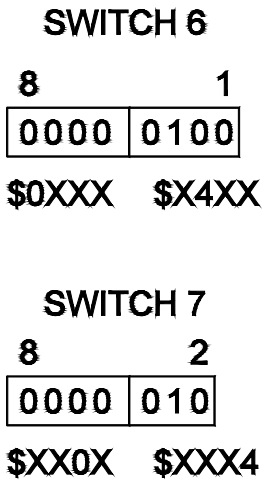


Figure 10 - Example of CSR Addressing

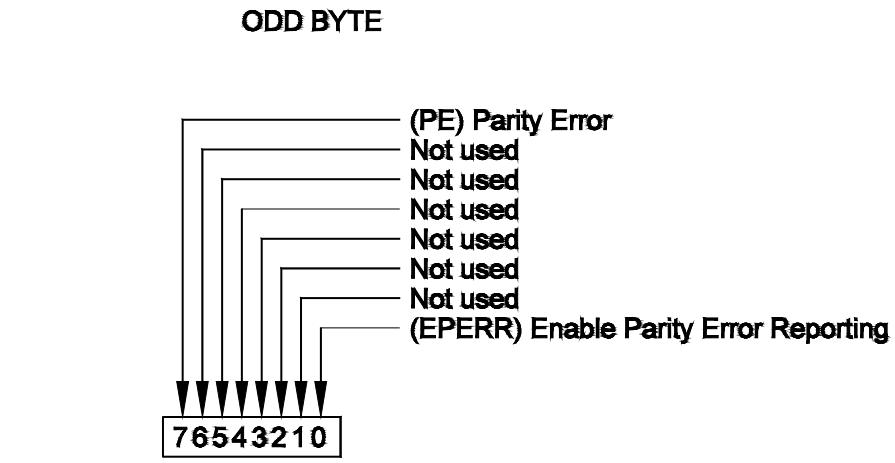
2.7 INSTALLATION IN THE SYSTEM

After selecting the memory for the proper address locations by setting the appropriate switches (see section 2.6 ADDRESS SWITCH SETTINGS), installing/removing the user defined option jumpers (see Section 2.5 JUMPER OPTIONS), and after following the guidelines set forth in Section 2.3 HARDWARE PREPARATION; remove power, install the MM-6290D High Speed Memory into the system, reapply power to the system.

SECTION III - PROGRAMMING INFORMATION

3.1 CONTROL STATUS REGISTER

The Control status register (CSR) provides a VMEbus accessible resister to control and determine the status of the MM-6290D High Speed Memory. In the following sections, The CSR bits are defined and their use is explained.



Note: The even byte, if read, will riturn an indeterminate value.

Figure 11 - C.S.R. Bit Definitions

3.1.1. Parity Error Detection

The MM-6290D High Speed Memory generates even parity for each byte written during a write access; and checks even parity for each byte read during a read access. If a parity error is detected during a read access, The CSR PE bit will be set. It will be reset by a system bus reset, and may also be reset by writing a 0 to bit 7 of the CSR. If bit 0 is set, (EPERR), and a parity error is detected during a VMEbus read access, The BERR* lead is asserted instead of the DTACK* lead. EPERR will be reset by a system bus reset, and may also be reset by writing a 0 to bit 0 of the CSR.

3.2 INITIALIZATION OF MEMORY

Since the memory will power up with random data, each byte location should be written to initialize its associated parity bit, before it is read, or before EPERR in the CSR is set to enable *BERR to be asserted for parity errors on read accesses.

APPENDEX A

```
TITLE           MM6290 Address Modifier Decoder
PATTERN         45940B
REVISION        V1.03

CHIP    AM_DECODE       PAL16L8

AM5 AM4 AM3 AM2 AM1 AM0 /IACK RESET BAS GND
NC NC NC /LAM24 BLK AM16 AM24 AM32 AMD64 VCC

EQUATIONS

/AMD64 = /(AM5 * AM4 * AM3 * AM2 * /AM1 * /AM0 * /RESET * /IACK
         + AM5 * AM4 * AM3 * /AM2 * /AM1 * /AM0 * /RESET * /IACK
         + /AM5 * /AM4 * AM3 * AM2 * /AM1 * /AM0 * /RESET * /IACK
         + /AM5 * /AM4 * AM3 * /AM2 * /AM1 * /AM0 * /RESET * /IACK)

/AM32 = /(/AM5 * /AM4 * AM3 * AM2 * AM1 * AM0 * /RESET * /IACK
         + /AM5 * /AM4 * AM3 * AM2 * AM1 * /AM0 * /RESET * /IACK
         + /AM5 * /AM4 * AM3 * AM2 * /AM1 * AM0 * /RESET * /IACK
         + /AM5 * /AM4 * AM3 * AM2 * /AM1 * /AM0 * /RESET * /IACK
         + /AM5 * /AM4 * AM3 * /AM2 * AM1 * AM0 * /RESET * /IACK
         + /AM5 * /AM4 * AM3 * /AM2 * AM1 * /AM0 * /RESET * /IACK
         + /AM5 * /AM4 * AM3 * /AM2 * /AM1 * AM0 * /RESET * /IACK
         + /AM5 * /AM4 * AM3 * /AM2 * /AM1 * /AM0 * /RESET * /IACK)

/AM24 = /(AM5 * AM4 * AM3 * AM2 * AM1 * AM0 * /RESET * /IACK
         + AM5 * AM4 * AM3 * AM2 * AM1 * /AM0 * /RESET * /IACK
         + AM5 * AM4 * AM3 * AM2 * /AM1 * AM0 * /RESET * /IACK
         + AM5 * AM4 * AM3 * AM2 * /AM1 * /AM0 * /RESET * /IACK
         + AM5 * AM4 * AM3 * /AM2 * AM1 * AM0 * /RESET * /IACK
         + AM5 * AM4 * AM3 * /AM2 * AM1 * /AM0 * /RESET * /IACK
         + AM5 * AM4 * AM3 * /AM2 * /AM1 * AM0 * /RESET * /IACK
         + AM5 * AM4 * AM3 * /AM2 * /AM1 * /AM0 * /RESET * /IACK)

/AM16 = /( AM5 * /AM4 * AM3 * AM2 * /AM1 * AM0 * /RESET * /IACK
         + AM5 * /AM4 * AM3 * /AM2 * /AM1 * AM0 * /RESET * /IACK)

/BLK = /( /BAS * AM5 * AM4 * AM3 * /AM1 * /AM0
       + /BAS * /AM5 * /AM4 * AM3 * /AM1 * /AM0
       + /BAS * /AM5 * /AM4 * AM3 * AM1 * AM0
       + /BAS * AM5 * AM4 * AM3 * AM1 * AM0
       +  BLK * AM5 * AM4 * AM3 * /AM1 * /AM0
       +  BLK * /AM5 * /AM4 * AM3 * /AM1 * /AM0
       +  BLK * /AM5 * /AM4 * AM3 * AM1 * AM0
       +  BLK * AM5 * AM4 * AM3 * AM1 * AM0
       +  BLK * BAS)

LAM24 = /BAS * AM5 * AM4 * AM3 * AM0
      + /BAS * AM5 * AM4 * AM3 * AM1
      + LAM24 * AM5 * AM4 * AM3 * AM0
      + LAM24 * AM5 * AM4 * AM3 * AM1
      + LAM24 * BAS
```