Micro Memory MM-6390/512

High Performance VME / VSB Expansion Memory Module

In Stock

Used and in Excellent Condition

Open Web Page

https://www.artisantg.com/61466-2

Table of Contents

SECTION I

GENERAL INFORMATION

SECTION II

INSTALLATION

SECTION III

PROGRAMMING INFORMATION

APPENDIX A

MODULE SELECTION

APPENDIX B

U1 PLD Equations

FIGURES

TABLES

SECTION I

GENERAL INFORMATION

1.1 INTRODUCTION

This manual describes the elements of operation and installation procedures of the **MM6390D** Dual Port Dynamic Random Access Memory Module.

1.2 GENERAL DESCRIPTION

The **MM6390D** Dual Ported Memory is compatible with the **VMEbus, VME Subsystem Bus (VSBus)** specifications. Inherent to the module are **VMEbus** options D64, D32, D16 & D8 (64, 32, 16 & 8 bit data path width), and A32. A24 (32, 24 bit address path width) and is a **VSBus** D32 *Slave*. The module also generates and stores an even parity bit for each byte written on write cycles and checks parity for each byte read on read cycles. Then if a parity error is detected, the module sets a **Control Status Register** (CSR) bit and may be programmed to assert the bus error signal on the requesting port. (see Section III for CSR programming)

For Extremely High Performance operation, the **MM6390D** Dual Ported Memory is equipped with a Dual Port PAGE MODE High Speed *BLOCK Transfer* capability. This allows the use of High Performance Block Mode Controllers to transfer upto 2048 (MLBT)/256 (BLT) bytes on the **VMEbus**, and 256 bytes on the **VSBus**, for increased system performance.

While the **MM6390D** Dual Ported Memory was designed as a 64 bit wide (data path) memory board, which complies with **VMEbus** Specifications (Revision D) and can be addressed as 8 bit bytes, 16 bit words, 32 bit longwords or 64 bit octalwords. It also complies with the **VSBus** Specifications (Revision C) as a D32 *Slave*.

The **MM6390D** Dual Ported Memory can be configured from 32M (128M) to 128M (512M) bytes capacity by populating it with 4096K X 9 (16384K X 9) 80ns CMOS DRAM Single-Inline-Packages (SIP).

1.2.1 Memory capacity options

The **MM6390D** Dual Ported Memory memory modules are available in several options depending on the capacity required. Table 1.1 lists the optional part numbers for ordering purposes, the total memory capacity provided by each version.

Table 1.1 Model Numbering

Model Number	Capacity
MM6390D/32M	32.0M Bytes
MM6390D/64M	64.0M Bytes
MM6390D/128M	128.0M Bytes
MM6390D/256M	256.0M Bytes
MM6390D/512M	512.0M Bytes

1.2.2 Operational features

The **MM6390D** Dual Ported Memory module contains its own address register and data buffers for each port for total compatibility with the **VMEbus** Specification (Rev. D) and **VSBus** Specification (Rev. C)

Module selection for the **MM6390D** resides on any 1024K (100000 hex) bytes boundaries (A24/A32). The module may be mapped anywhere within the 4 G bytes of physical address space, on each port, both the **VMEbus** and the **VSBus**.

The **MM6390D** Dual Ported Memory 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 HIGH Performance BLOCK Transfers (MBLT) & (BLT).

The **MM6390D** Dual Ported Memory responds to various Address Modifier Codes. The codes are decoded by U1 (a socketed PLD), Various combinations are available to the user (see Section II, jumpers).

The user may select to create his/her own Address Modifier code, in this event the Address Modifier evaluation PLD (U1) can be replaced to accommodate the systems' requirements (see Section II, jumpers).

A **Control Status Register** (CSR) is available to allow various dynamic options to to be utilized by the software programmer. Included in the CSR is a bit to make the **MM6390D** Dual Ported Memory a Private Bus only board (ie: **VSBus** only). A bit controls the response of the **VSBus** *Cache* lead. The CSR is an A16 Slave and has separate address selection jumpers, and may be addressed on any WORD boundary in the **VMEbus** I/O address space, using a combination of jumper & VMEbus *Low Limit* switch settings. (See Section II for selection & Section III for programming details).

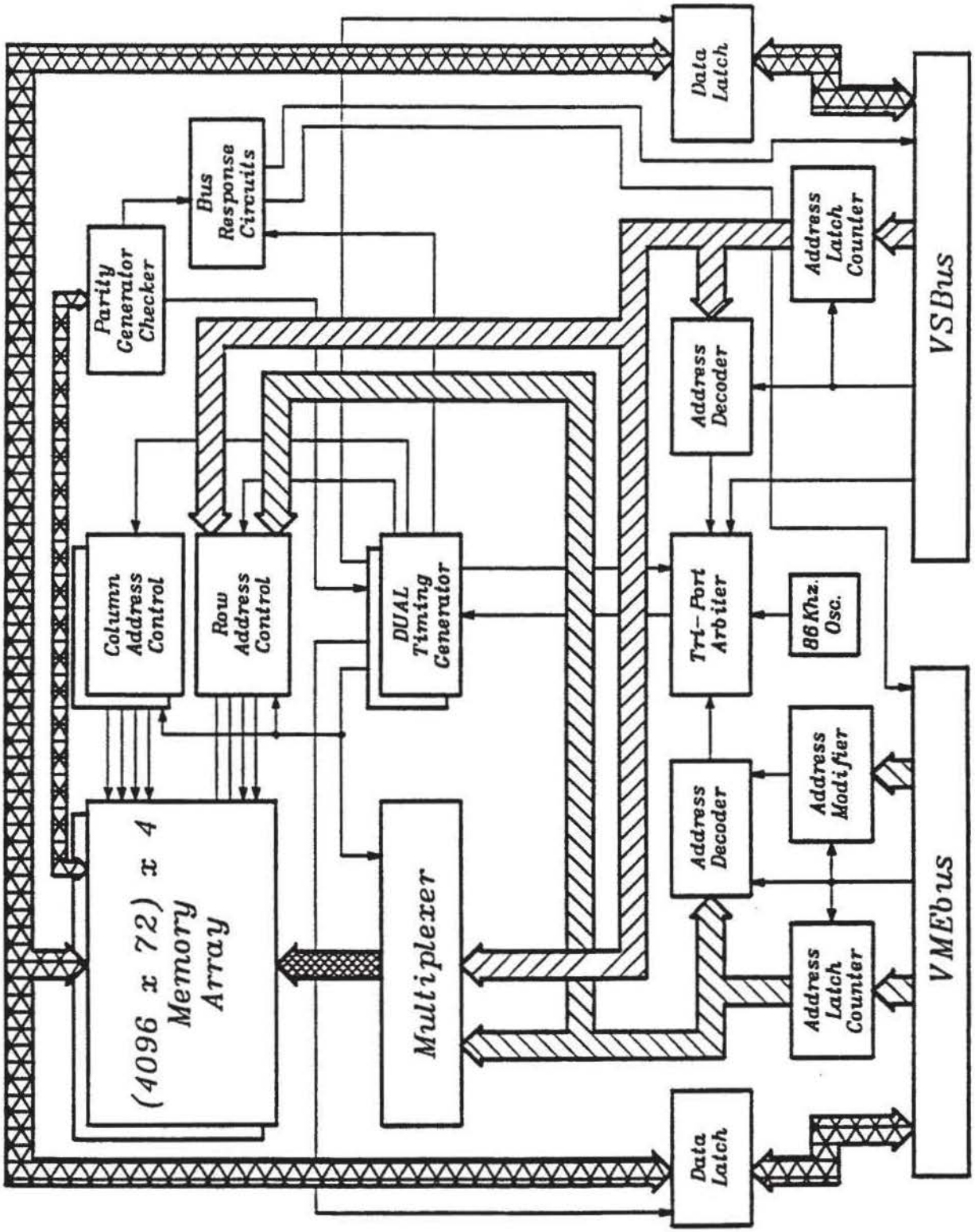


Figure 1.1 Block Diagram

1.3 GENERAL SPECIFICATIONS

Table 1.2 General Specifications

Characteristics	Specifications	
Capacity	32M ,64M, 128M, 256M or 512Mbytes	
Cycle Time	250ns. (non-Block mode) 100ns. (Block mode)	
Access Time (Non-Block mode)	Read 94nsec VSB, 195nsec VME Write 34nsec VSB, 120nsec VME	
Access Time (Block Mode)	Read 13nsec VSB, 34nsec VME Write 13nsec VSB, 34nsec VME	
Address	32 bits A32/BLT/MBLT/UAT VSBus	
Data In/Data Out	8/16/24/32/64 bits bidirectional with three-state output D08/D16/D32	
Modes of Operation	Read, Write, Read/Modify/Write (RMW), Block Transfers (BLT), (MBLT) Un-aligned (UAT)	
Address Modifiers	6 bits, jumper selectable, or user programmable PLD.	
Parity	EVEN, 1 bit for each byte Generated on each WRITE Checked on each read.	
VMEbus and VSBus Module Selection	Memory selected on 1024K byte ($100000) boundary.	
Control Status Register Selection	CSR Selection based on VMEbus Start Address, in VMEbus 'A16'I/O Address space.	
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)	4.5A	5.5A

1.4 INTERFACE TIMING

The MM6390D Dual Ported Memory memory modules are designed to accept 4096K x 9 (16384K X 9) DYNAMIC Random Access Memory modules with a 80nsec typical access delay Time.

1.4.1 Refresh Timing

The refresh cycle timing is illustrated by Figure 1.2. Refresh cycles are repeated approximately once every 15usec. The cycle interval timing is generated by a free running oscillator (U98) providing a refresh transparent to memory access cycles.

The MM6390D contains a refresh request counter and can store upto 31 refresh requests. During memory cycles that take more than the 15us. refresh interval, this counter will record how many refresh requests have been made and upon completion of the cycle (the memory board is released ie: PAS* negated for VSB cycles and AS* negated for VME cycles) the refresh circuit run upto the 31 stored refresh requests.

(See Section 2.5.6 regarding VME SYSFAIL* - Refresh Timeout)

- **NOTE: Memory cycles must be completed within 420us. or REFRESH cycles will be lost. Resulting in possible data corruption.**

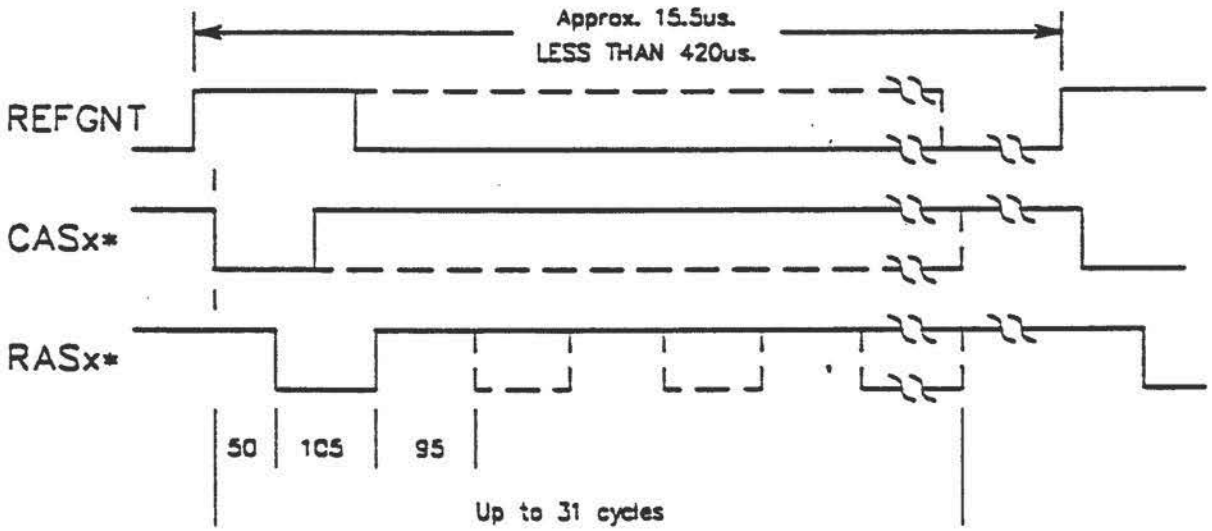


Figure 1.2 Refresh Timing

1.4.2 VMEbus Interface Timing

The VMEbus Interface Timing Diagram (Figure 1.3), illustrates the timing relationships on the Data Transfer bus. for Read and Write Cycles. Access delay time is measured from the leading edge of the Data strobes. (Figure 1.3).

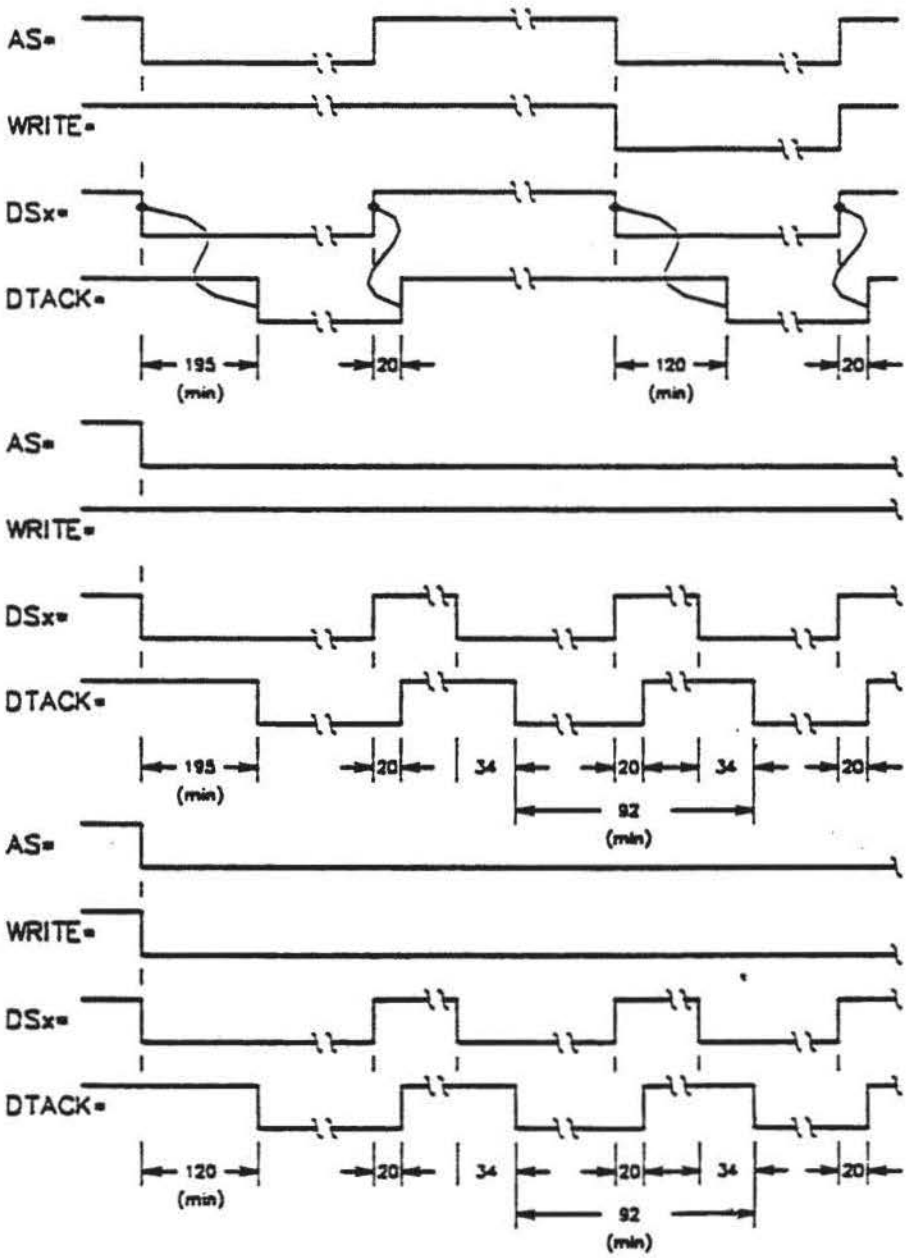


Figure 1.3 VME Interface Timing

1.4.3 VSBus Interface Timing

The VSBus Interface Timing Diagram (Figure 1.4), illustrates the timing relationships on the Data Transfer bus, for Read and Write Cycles. Access delay time is measured from the leading edge of the Data strobes. (Figure 1.4).

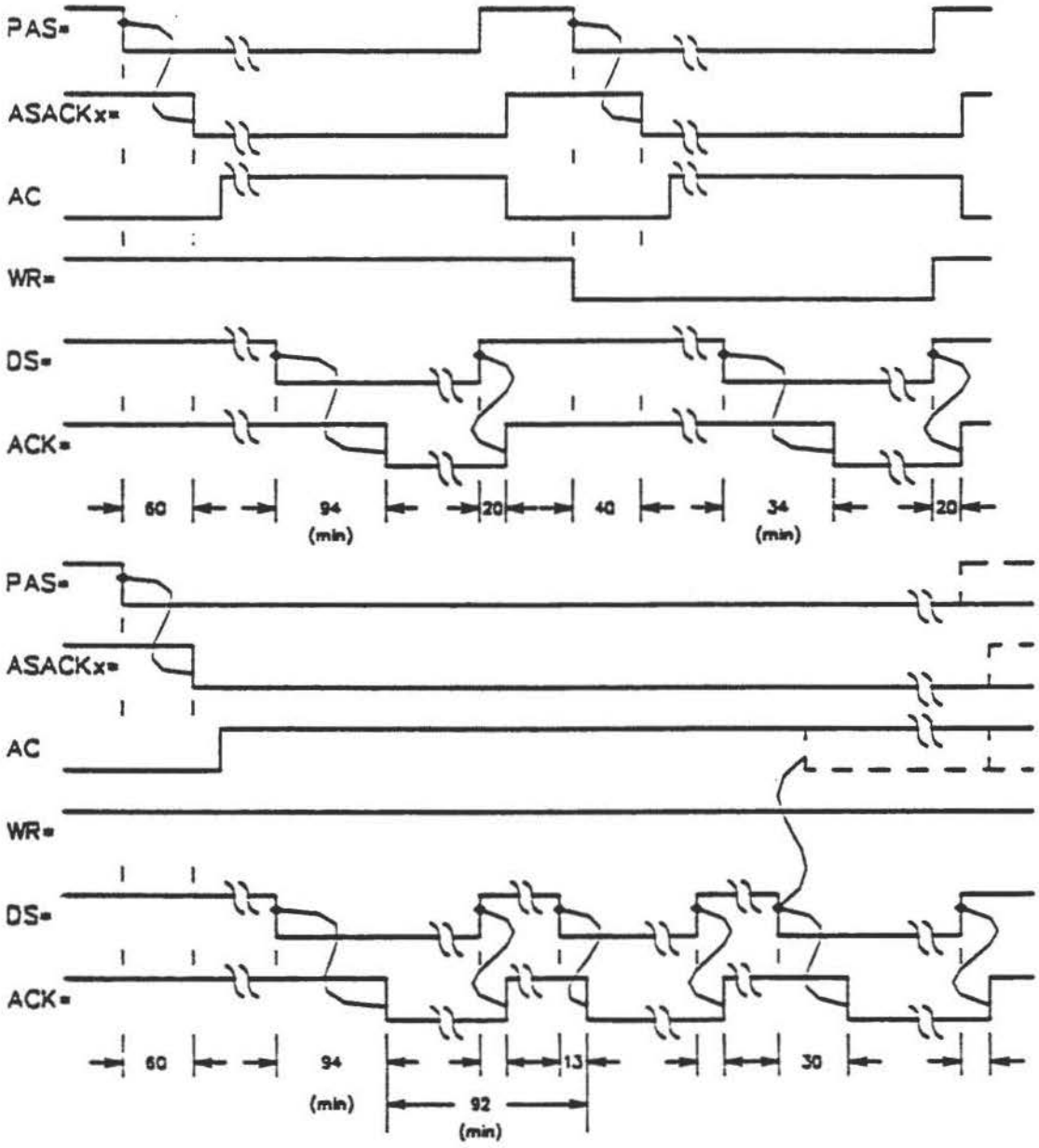


Figure 1.4 VSBus Interface Timing

Table 1.3 P1 - VMEbus Connector Pin Definitions

Pin	A	B	C
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	A17
22	IACKOUT*	SERDAT	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 MM6390D Dual Ported Memory

Table 1.4 P2 - VME/VSB Connector Definitions

Pin	A	B	C
1	AD00	+5v	AD01
2	AD02	GND	AD03
3	AD04	RESERVED	AD05
4	AD06	A24	AD07
5	AD08	A25	AD09
6	AD10	A26	AD11
7	AD12	A27	AD13
8	AD14	A28	AD15
9	AD16	A29	AD17
10	AD18	A30	AD19
11	AD20	A31	AD21
12	AD22	GND	AD23
13	AD24	+5v	AD25
14	AD26	D16	AD27
15	AD28	D17	AD29
16	AD30	D18	AD31
17	GND	D19	GND
18	IRQ*	D20	GND
19	DS*	D21	GND
20	WR*	D22	GND
21	SPACE0	D23	SIZE0
22	SPACE1	GND	PAS*
23	LOCK*	D24	SIZE1
24	ERR*	D25	GND
25	GND	D26	ACK*
26	GND	D27	AC
27	GND	D28	ASACK1*
28	GA0	D29	ASACK0*
29	GA1	D30	CACHE*
30	GA2	D31	WAIT*
31	BGIN*	GND	BUSY*
32	BREQ*	+5v	BGOUT*

** - No Connection on MM6390D Dual Ported Memory

1.5 MEMORY ARRAY ORGANIZATION

The memory array is partitioned into four (4) 4096K x 72 (16384K x 72) bit segments. The minimum memory configuration would be 4096K x 72 (16384K x 72) bits. including the four parity bit DRAMs in segment zero. This would translate to 32Megabytes (128Megabytes) as a minimum configuration for the **MM6390D Dual Ported Memory**. Each partition or block is organized as 72 bits each (64 plus 8 parity), however the system will only see 64 bits of data during memory references, as the parity is generated and checked transparent to the user. The 64 bits of data are divided into 8 bytes (8 bits each). While the memory reads and writes may be 8, 16, 32 or 64 bits wide, the memory is organized as 64 bit wide locations with logic controlling which byte the user is reading or writing.

If the contents of the first 8 locations in memory are S01 S23 S45 S67 S89 SAB SCD SEF, 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, and Longword).

```
BYTE Format          WORD Format          LONGWORD Format
Location    Data     Location    Data     Location    Data
-------------------------------------------------------------
S00000000   S01   |  S00000000   S0123  |  S00000000  S01234567
S00000001   S23   |  S00000002   S4567  |  S00000004  S89ABCDEF
S00000002   S45   |  S00000004   S89AB  |
S00000003   S67   |  S00000006   SCDEF  |
S00000004   S89   |-----------------------------------------
S00000005   SAB   |              QUADWORD Format
S00000006   SCD   |       Location    Data
S00000007   SEF   |       S00000000   S0123456789ABCDEF

                                BIT Organizations
QUADWORD
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
S0  S1  S2  S3   S4  S5  S6  S7   S8  S9  SA  SB   SC  SD  SE  SF
```

Figure 1.5 Memory Array Organization

SECTION II

INSTALLATION

2.1 INTRODUCTION

This section details a step by step procedure to interface the MM6390D Dual Ported Memory to both the VMEbus and VSBus.

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 Section 2.5, for jumper options).

2.3.1 Handling Procedure

The **MM6390D** Dual Ported 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.

- Device leads should contact conductive material to avoid building of any static charge, except during testing or operation.
- Soldering iron tips, metal fixtures, tools, and handling facilities used in preparing the module for operation should be grounded.
- 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.
- **The memory module should never be plugged in or out of the cardcage while power is applied.**
- External signals should not be applied to device inputs while power is removed.
- 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 MM6390D Dual Ported Memory is an extremely High Density memory product. Containing, 96 bits of high current (48ma.) data bus drivers, numerous high speed components. And as a result, the environment into which the MM6390D Dual Ported Memory is placed, must be carefully considered.

2.4.1 Airflow and Cooling

Due to the extremely high component density of the MM6390D Dual Ported 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 MM6390D Dual Ported 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 Tables 1.3 & 1.4) of both the P1 and the P2 connector prior to operation of the MM6390D Dual Ported Memory.

2.5 JUMPER OPTIONS

There are several jumper options attainable on the MM6390D Dual Ported Memory module, depending on the user application. All of the available options are selected by installing or removing minijumpers on wirewrap pins on 0.100 inch centers. (see Figure 2.1)

2.5.1 Address Modifier Evaluation (Factory, A32 and A24 ENABLED)

The MM6390D Dual Ported Memory evaluates the Address Modifiers from the VMEbus to determine if they are to be responded to. As shipped from the factory, the MM6390D Dual Ported Memory responds to the EXTENDED A32 (using all 32 address bits) Address Modifier codes and the STANDARD A24 (ignoring the upper 8 bits). Optional, the USER may remove E8 to disable the EXTENDED A32 selection and/or E7 STANDARD A24 selection. These Address Modifier codes are described in Appendix D. The user may select to replace the Address Modifier selection PLD (U1), to allow for some other combination or the creation of a special (user defined) code or codes.

2.5.2 VSBus Alternate Address Space Selection (Factory, OPEN)

The MM6390D Dual Ported Memory normally is selected in the VSBus in the System Space, however the user may configure the MM6390D Dual Ported Memory to respond to the Alternate Space selection by removing jumper E5 and shorting jumper E6. (See Figure 2.1 for locations)

2.5.3 Rescinded DTACK*/ACK* (Factory, DISABLED)

As shipped from the factory the MM6390D Dual Port Memory is configured to use an 74S38 (an OPEN Collector device) to generate DTACK* (VMEbus) and ACK* (VSBus). This conforms to the VMEbus and VSBus specifications. However, on heavily loaded systems (ie: full 21 slot cardcage) this can cause DTACK* to take over 40ns to be negated. This can prevent most systems from attaining the maximum bus bandwidth of 40 Megabytes per second (100ns bus cycles). On certain system configurations (ie: those without 'PARTICIPATING SLAVES') the designer MAY wish to use a 'RESCINDED DTACK*'. That is to say an 74FCT240 (with 3 state control) is used in place of the 74S38. Moving jumper E4 from pins 1 to 2, to pins 2 to 3 will enable the VMEbus 'RESCINDED DTACK*', while moving jumper E1 from pins 1 to 2, to pins 2 to 3 will enable the same on the VSBus.

- **Caution!** This will *violate* VMEbus specifications, and should ONLY be attempted with a thorough understanding of the system and its configuration.

2.5.4 VMEbus Fast! DTACK* (Factory, DISABLED)

Due to VMEbus specification, the MM6390 Dual Ported Memory is required to wait a minimum of 30ns before reponding to DS0* or DS1* with the DTACK* signal. If jumper E3 is removed, the MM6390 Dual Ported Memory can repond in as little as 13ns from the earliest Data Strobe (DS0* or DS1*). Data Strobe skew (DS0* or DS1*) may cause un wanted results.

- **Caution!** This will *violate* VMEbus specifications, and should ONLY be attempted with a thorough understanding of the system and its configuration.

2.5.5 Control Status Register (CSR) Enable (Factory, ENABLED)

The Control Status Register (CSR) address decoding may be disabled by removing jumper E16.

2.5.6 System FAIL (SYSFAIL*) Refresh Timeout (Factory ENABLED)

If memory cycles exceed the refresh counter (See Section 1.4.1 Refresh Timing) capacity of 31 requests the MM6390D light the FAIL L.E.D. and attempt to report a 'System FAILURE' by asserting SYSFAIL* on the VME bus. Removal of jumper E3 will prevent SYSFAIL* from being asserted.

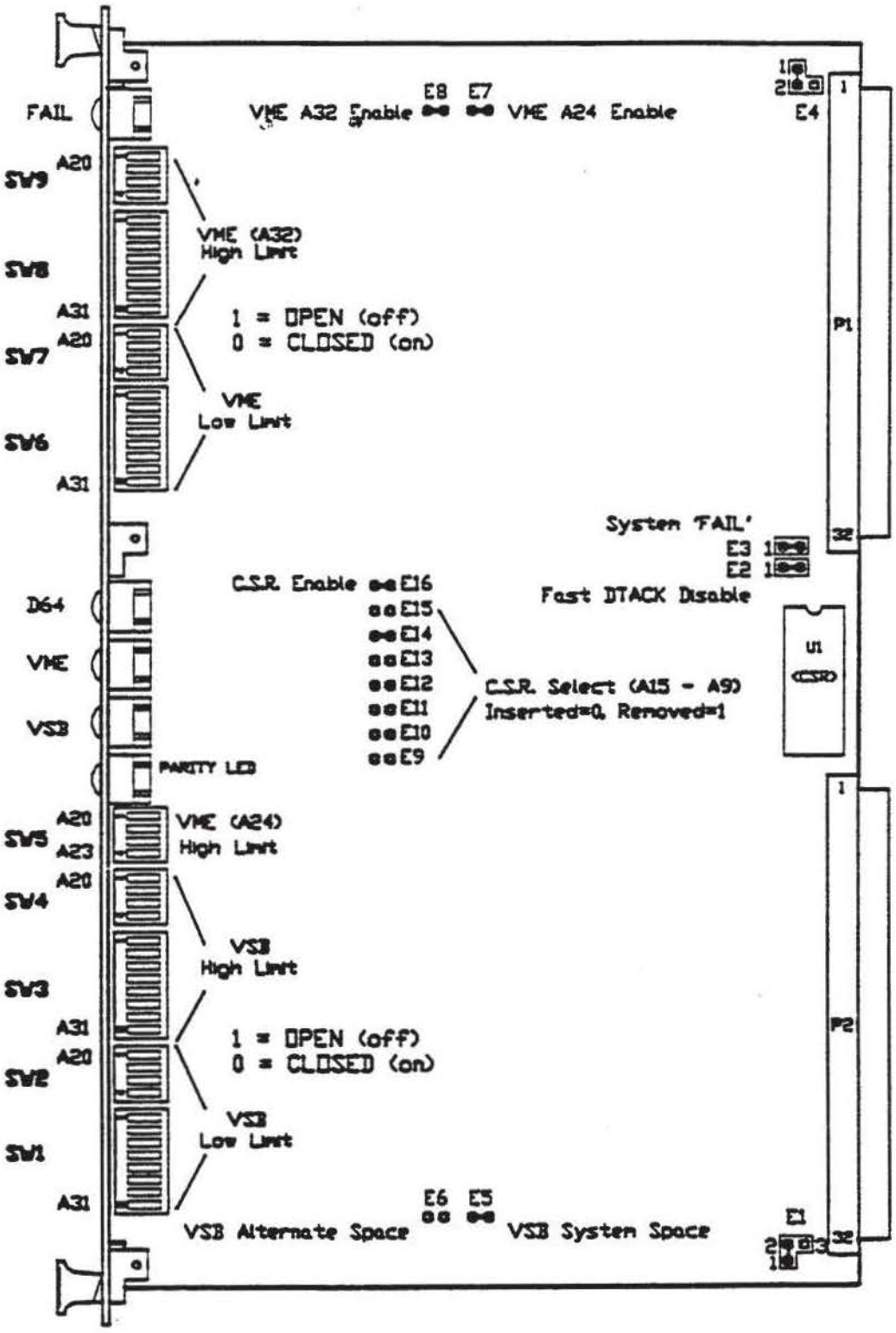


Figure 2.1 Jumpers & Switch Locations

2.6 CONFIGURATION JUMPER & SWITCH SETTINGS

The **MM6390D** Dual Port Memory contains four (4) major jumper blocks and a single jumper pair that must be configured prior to installation before proper operation can be obtained. (See Figure 2.1 for jumper locations)

Each port of the **MM6390D** Dual Port Memory must be addressed within it's four (4) Gigabyte address space. The **VMEbus** port is controlled by switches SW6, SW7, SW8 & SW9, also by jumpers E7 (A24 Address Modifier Enable) and E8 (A32 Address Modifier Enable See Section 2.5.1). While the **VSBus** port is controlled by switches SW1, SW2, SW3 & SW4, also by jumpers E5 & E6 (Select Alternate Space or System Space See Section 2.5.2).

2.6.1 VMEbus Module Selection

Selection of the **VMEbus** lower limit and upper limit on 1024Kilobyte byte (S100000) boundaries and accomplished using switches SW6 & SW7 for the LOW LIMIT and switches SW8 & SW9 for the A32 space UPPER LIMIT. Position 8 of switches SW6 & SW8 is the most significant address (A31) and position 1 of switches SW7 & SW9 is the least significant address (A20) for both UPPER and LOWER limits. While Position 4 or SW5 is the most significant address (A23) and position 1 is the least significant address (A20) for A24 space UPPER LIMIT.

When the **MM6390D** Dual Ported Memory is selected using an A24 address modifier, the **MM6390** will automatically force a 0 on internal addresses A24 thru A31, forcing the address request into the FIRST 16Megabyte block and will use SW5 for the UPPER LIMIT selection (Position 4 is the most significant address (A23) while position 1 is the least significant (A20)).

- NOTE: The UPPER LIMIT is an *inclusive* address. Therefore an 128MByte board addressed to start at S00000000 would have the LOW LIMIT set to S00000000 and the UPPER LIMIT set to S07FFFFFF (S000 to S07F on the switches). **NOT** S000 to S080.

2.6.2 VSBus Module Selection

As with the **VMEbus** module selection, the **VSBus** lower limit and upper limit may be selected on any 1024Kilobyte byte (S100000) boundaries and is accomplished using switches SW1 & SW2 for the LOW LIMIT and switches SW3 & SW4 for the UPPER LIMIT. Position 8 of switches SW1 & SW3 is the most significant address (A31) and position 1 of switches SW2 & SW4 is the least significant address (A20) for both UPPER and LOWER limits.

- NOTE: The UPPER LIMIT is an *inclusive* address. Therefore an 128MByte board addressed to start at S00000000 would have the LOW LIMIT set to S00000000 and the UPPER LIMIT set to S07FFFFFF (S000 to S07F on the switches, **NOT** S000 to S080).

2.6.3 Adjusted Addressing

In normal operation, the **MM6390D** Dual Ported Memory would be selected on a 128/512 Megabyte boundary and simply set the *UPPER* and **LOWER** limits accordingly. However, in some applications the user may wish to more accurately control the starting and ending addresses. An example would be when the user wants to start the memory immediately following some on-board memory (IE: 1 Megabyte CPU on-board memory). To keep both ports aligned (IE: first byte on **VSBus** port is first byte on **VMEbus** port), the addresses applied to the **MM6390D** Dual Ported Memory must be adjusted. This is accomplished by on board circuitry that subtracts the LOW LIMIT address from the address applied from the bus. The **MM6390D** Dual Ported Memory allows the user to select an address on any 1 Megabyte boundary, and still be fully aligned.

2.6.4 Control Status Register (CSR) Selection

If enabled (See section 2.5.5) the CSR address selection is a combination of jumpers E15, E14, E13, E12, E11, E10, E9 and the **VMEbus** LOW LIMIT selection. The CSR is located in **VMEbus** A16 (or I/O) space, and responds to ONLY A16 Address Modifiers (S29 & S2D. See section 2.5.1 for more details). Jumper E15 corresponds to the most significant address A15 while E9 corresponds to the least significant address A9. Addresses A8 down through A1 correspond to the **VMEbus** LOW LIMIT address positions for addresses A27 through A20. Where SW7 position 8 (A27) corresponds to A8, SW7 position 7 (A26) corresponds to A7 and so on.

2.7 INSTALLATION IN THE SYSTEM

After selecting the memory for the proper address locations by opening/closing the appropriate switches (see Section 2.6) and installing/removing the user defined option jumpers (see Section 2.5). And after following the guidelines set forth in Section 2.3, remove power, install the **MM6390D** Dual Ported Memory into the system, reapply power to the system.

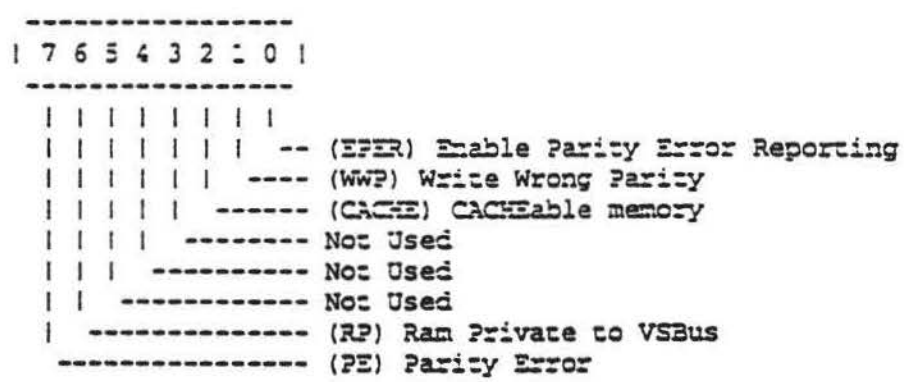
SECTION III

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 MM6390D Dual Ported Memory. In the following sections, the CSR bits are defined and their use is explained.

Table 3.1 C.S.R. Bit Definitions



3.2 Parity Error Detection

The MM6390D Dual Ported Memory generates and checks for errors in the memory by calculating the parity of each byte in memory. During write cycles, the MM6390D Dual Ported Memory writes this bit into the array, and on read cycles it checks this bit against the bit in the array. If, during a read cycle, a difference is detected the CSR PE bit is set and if the CSR EPER bit is also set, the appropriate ERROR lead is asserted. In the case of a VMEbus reference, the BERR* lead is asserted instead of the DTACK* lead. However on the VSBus, the ERR* is asserted instead of the ACK* lead. The PE bit is cleared by reset or by writing a 0 into CSR bit7.

- To insure proper parity operation, the MM6390D Dual Ported Memory must be *Initialized*. That is to say a memory location MUST be written *before* it is read.

- NOTE: Due to certain memory caching systems used with the VSBus, the MM6390D Dual Port Memory reads *ALL* 32 bits during *ANY* read cycle. As a result, it is recommended that *ALL* memory be *Initialized* using a 32 bit write instruction (IE: The 680x0 "CLR.L" instruction or equivalent), to prevent unexpected Parity ERRORS.

3.3 RP - Ram Private

The MM6390D Dual Ported Memory may be reserved in *TOTAL*, by setting the RP bit in the CSR to a 1. Any references made of the MM6390D Dual Ported Memory from the VMEbus port are ignored while this bit is set.

3.4 Cacheable Memory Enable

When set, this bit disables the CACHE* lead on the VSBus, indicating, to a bus master, that the selected memory is *NOT* cacheable.

3.5 Write Wrong Parity

When set, this bit will force the MM6390 to generate the WRONG parity for each byte written. This allows testing of the parity bit memory device for increased system confidence.

- This mode may be used to also test BUS ERROR Reporting circuits on other boards.

APPENDIX A
MODULE SELECTION

VME/VSB Switch Table (A31 - A28)

(A31 - A28) 1 = Open
0 = Close

(VMEbus = SW6 & SW8) Low Limit					(VSBUS = SW1 & SW3) High Limit				
SWITCH POSITION	SW6/SW1 8	7	6	5	SWITCH POSITION	SW8/SW3 8	7	6	5
S0yz00000	0	0	0	0	S0yzFFFFF	0	0	0	0
S1yz00000	0	0	0	1	S1yzFFFFF	0	0	0	1
S2yz00000	0	0	1	0	S2yzFFFFF	0	0	1	0
S3yz00000	0	0	1	1	S3yzFFFFF	0	0	1	1
S4yz00000	0	1	0	0	S4yzFFFFF	0	1	0	0
S5yz00000	0	1	0	1	S5yzFFFFF	0	1	0	1
S6yz00000	0	1	1	0	S6yzFFFFF	0	1	1	0
S7yz00000	0	1	1	1	S7yzFFFFF	0	1	1	1
S8yz00000	1	0	0	0	S8yzFFFFF	1	0	0	0
S9yz00000	1	0	0	1	S9yzFFFFF	1	0	0	1
SAyz00000	1	0	1	0	SAyzFFFFF	1	0	1	0
SByz00000	1	0	1	1	SByzFFFFF	1	0	1	1
SCyz00000	1	1	0	0	SCyzFFFFF	1	1	0	0
SDyz00000	1	1	0	1	SDyzFFFFF	1	1	0	1
SEyz00000	1	1	1	0	SEyzFFFFF	1	1	1	0
SFyz00000	1	1	1	1	SFyzFFFFF	1	1	1	1

VME/VSB Switch Table (A27- A24)

(A27 - A24) 1 = Open
0 = Close

(VMEbus = SW6 & SW8) (VSBUS = SW1 & SW3)

Low Limit					High Limit				
SWITCH POSITION	SW6/SW1 4	3	2	1	SWITCH POSITION	SW8/SW3 4	3	2	1
$x0z00000	0	0	0	0	$x0zFFFFF	0	0	0	0
$x1z00000	0	0	0	1	$x1zFFFFF	0	0	0	1
$x2z00000	0	0	1	0	$x2zFFFFF	0	0	1	0
$x3z00000	0	0	1	1	$x3zFFFFF	0	0	1	1
$x4z00000	0	1	0	0	$x4zFFFFF	0	1	0	0
$x5z00000	0	1	0	1	$x5zFFFFF	0	1	0	1
$x6z00000	0	1	1	0	$x6zFFFFF	0	1	1	0
$x7z00000	0	1	1	1	$x7zFFFFF	0	1	1	1
$x8z00000	1	0	0	0	$x8zFFFFF	1	0	0	0
$x9z00000	1	0	0	1	$x9zFFFFF	1	0	0	1
$xAz00000	1	0	1	0	$xAzFFFFF	1	0	1	0
$xBz00000	1	0	1	1	$xBzFFFFF	1	0	1	1
$xCz00000	1	1	0	0	$xCzFFFFF	1	1	0	0
$xDz00000	1	1	0	1	$xDzFFFFF	1	1	0	1
$xEz00000	1	1	1	0	$xEzFFFFF	1	1	1	0
$xFz00000	1	1	1	1	$xFzFFFFF	1	1	1	1

VME/VSB Switch Table (A23- A20)

(A23 - A20) 1 = Open
0 = Close

(VMEbus = SW7 & SW9) Low Limit					(VSBUS = SW2 & SW4) High Limit				
SWITCH POSITION		SW7/SW2			SWITCH POSITION		SW9/SW4		
	4	3	2	1		4	3	2	1
$xy000000	0	0	0	0	$xy0FFFFF	0	0	0	0
$xy100000	0	0	0	1	$xy1FFFFF	0	0	0	1
$xy200000	0	0	1	0	$xy2FFFFF	0	0	1	0
$xy300000	0	0	1	1	$xy3FFFFF	0	0	1	1
$xy400000	0	1	0	0	$xy4FFFFF	0	1	0	0
$xy500000	0	1	0	1	$xy5FFFFF	0	1	0	1
$xy600000	0	1	1	0	$xy6FFFFF	0	1	1	0
$xy700000	0	1	1	1	$xy7FFFFF	0	1	1	1
$xy800000	1	0	0	0	$xy8FFFFF	1	0	0	0
$xy900000	1	0	0	1	$xy9FFFFF	1	0	0	1
$xyA00000	1	0	1	0	$xyAFFFFF	1	0	1	0
$xyB00000	1	0	1	1	$xyBFFFFF	1	0	1	1
$xyC00000	1	1	0	0	$xyCFFFFF	1	1	0	0
$xyD00000	1	1	0	1	$xyDFFFFF	1	1	0	1
$xyE00000	1	1	1	0	$xyEFFFFF	1	1	1	0
$xyF00000	1	1	1	1	$xyFFFFFF	1	1	1	1

```
AM24    = AM5 * AM4 * AM3 * /AM2 * /AM1 * AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * /AM2 * AM1 * /AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * /AM2 * AM1 * AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * AM2 * /AM1 * AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * AM2 * AM1 * /AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * AM2 * AM1 * AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * /AM2 * /AM1 * /AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * AM2 * /AM1 * /AM0 * /SRST * /IACK * /DIS

AMCSR   = AM5 * /AM4 * AM3 * /AM2 * /AM1 * AM0 * /SRST * /IACK
        + AM5 * /AM4 * AM3 * AM2 * /AM1 * AM0 * /SRST * /IACK

BLK     = /AM5 * /AM4 * AM3 * /AM2 * AM1 * AM0 * /SRST * /IACK * /DIS
        + /AM5 * /AM4 * AM3 * AM2 * AM1 * AM0 * /SRST * /IACK * /DIS
        + /AM5 * /AM4 * AM3 * /AM2 * /AM1 * /AM0 * /SRST * /IACK * /DIS
        + /AM5 * /AM4 * AM3 * AM2 * /AM1 * /AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * /AM2 * AM1 * AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * AM2 * AM1 * AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * /AM2 * /AM1 * /AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * AM2 * /AM1 * /AM0 * /SRST * /IACK * /DIS

D64     = /AM5 * /AM4 * AM3 * /AM2 * /AM1 * /AM0 * /SRST * /IACK * /DIS
        + /AM5 * /AM4 * AM3 * AM2 * /AM1 * /AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * /AM2 * /AM1 * /AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * AM2 * /AM1 * /AM0 * /SRST * /IACK * /DIS

SHTBLK  = AM5 * AM4 * AM3 * /AM2 * AM1 * AM0 * /SRST * /IACK * /DIS
        + AM5 * AM4 * AM3 * AM2 * AM1 * AM0 * /SRST * /IACK * /DIS
        + /AM5 * /AM4 * AM3 * /AM2 * AM1 * AM0 * /SRST * /IACK * /DIS
        + /AM5 * /AM4 * AM3 * AM2 * AM1 * AM0 * /SRST * /IACK * /DIS

RDCSR   = AM5 * /AM4 * AM3 * /AM2 * /AM1 * AM0 * /SRST * /IACK * /WR * CSRHEQ * CSRLEQ
        + AM5 * /AM4 * AM3 * AM2 * /AM1 * AM0 * /SRST * /IACK * /WR * CSRHEQ * CSRLEQ

WRCSR   = AM5 * /AM4 * AM3 * /AM2 * /AM1 * AM0 * /SRST * /IACK * WR * CSRHEQ * CSRLEQ
        + AM5 * /AM4 * AM3 * AM2 * /AM1 * AM0 * /SRST * /IACK * WR * CSRHEQ * CSRLEQ
```

This page left intentionally blank

VME/VSB Switch Table (A23- A20)

(A23 - A20) 1 = Open
0 = Close

(VMEbus = SW7 & SW9) Low Limit					(VSBUS = SW2 & SW4) High Limit				
SWITCH POSITION	SW7/SW2 4	3	2	1	SWITCH POSITION	SW9/SW4 4	3	2	1
$xy000000	0	0	0	0	$xy0FFFFF	0	0	0	0
$xy100000	0	0	0	1	$xy1FFFFF	0	0	0	1
$xy200000	0	0	1	0	$xy2FFFFF	0	0	1	0
$xy300000	0	0	1	1	$xy3FFFFF	0	0	1	1
$xy400000	0	1	0	0	$xy4FFFFF	0	1	0	0
$xy500000	0	1	0	1	$xy5FFFFF	0	1	0	1
$xy600000	0	1	1	0	$xy6FFFFF	0	1	1	0
$xy700000	0	1	1	1	$xy7FFFFF	0	1	1	1
$xy800000	1	0	0	0	$xy8FFFFF	1	0	0	0
$xy900000	1	0	0	1	$xy9FFFFF	1	0	0	1
$xyA00000	1	0	1	0	$xyAFFFFF	1	0	1	0
$xyB00000	1	0	1	1	$xyBFFFFF	1	0	1	1
$xyC00000	1	1	0	0	$xyCFFFFF	1	1	0	0
$xyD00000	1	1	0	1	$xyDFFFFF	1	1	0	1
$xyE00000	1	1	1	0	$xyEFFFFF	1	1	1	0
$xyF00000	1	1	1	1	$xyFFFFFF	1	1	1	1

APPENDIX B
U1 PLD Equations

```
TITLE     MM6390 VMEbus AM Control
PATTERN   U1 - PN46102
REVISION  NC
AUTHOR    Michael Hasenfratz
COMPANY   Micro Memory Inc. Chatsworth, Ca.
DATE      January 4, 1993 [04:59pm]

;       This device controls the VMEbus Address Modifier/Selection

CHIP AM PAL22V10

AM0       ; Address Modifier 0
AM1       ; Address Modifier 1
AM2       ; Address Modifier 2
AM3       ; Address Modifier 3
AM4       ; Address Modifier 4
AM5       ; Address Modifier 5
SRST      ; System RESET
AS        ; Address Strobe
DIS       ; VMEbus DISable
WR        ; VME WRite
/CSRHEQ   ; CSR HIGH .EQ.
GND       ; Ground (Vss)
/CSRLEQ   ; CSR LOW  .EQ.
NC        ; No Connection
/IACK     ; Interrupt ACKnowlage
D64       ; VME D64 (MBLT) AM
SHTBLK    ; SHORT BLOCK AM
BLK       ; BLOCK MODE REQUESTED
/WRCSR    ; WRITE to CSR
/RDCSR    ; READ from CSR
/AMCSR    ; VME I/O (A16) AM
/AM24     ; VME A24 AM
/AM32     ; VME A32 AM
VCC       ; Power (Vdd)

NC        ; No Connection

EQUATIONS

AM32      = /AM5 * /AM4 *  AM3 * /AM2 * /AM1 *  AM0 * /SRST * /IACK * /DIS
          + /AM5 * /AM4 *  AM3 * /AM2 *  AM1 * /AM0 * /SRST * /IACK * /DIS
          + /AM5 * /AM4 *  AM3 * /AM2 *  AM1 *  AM0 * /SRST * /IACK * /DIS
          + /AM5 * /AM4 *  AM3 *  AM2 * /AM1 *  AM0 * /SRST * /IACK * /DIS
          + /AM5 * /AM4 *  AM3 *  AM2 *  AM1 * /AM0 * /SRST * /IACK * /DIS
          + /AM5 * /AM4 *  AM3 *  AM2 *  AM1 *  AM0 * /SRST * /IACK * /DIS
          + /AM5 * /AM4 *  AM3 * /AM2 * /AM1 * /AM0 * /SRST * /IACK * /DIS
          + /AM5 * /AM4 *  AM3 *  AM2 * /AM1 * /AM0 * /SRST * /IACK * /DIS
```