GENERAL AUTOMATION

GA-16/220/230/240 addendum

88A00654A-A

NOTICE

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

General Automation makes no warranty or representation with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. General Automation shall not be liable for errors contained herein.

General Automation assumes no responsibility for the use or reliability of its software on equipment that is not furnished by General Automation.

88A00654A-A

GA-16/220/230/240 addendum

GENERAL AUTOMATION
1055 South East Street
Anaheim, California 92805
(714) 778-4800
TWX 910-591-1695

Prepared on GA Econotext system

RECORD OF REVISIONS

Title: GA-16/220/230/240 Addendum

Document Number: 88A00654A

DATE	ISSUE
Sep 80	Original Issue

FOREWORD

The purpose of this GA-16/220/230/240 Addendum is to supplement the current issues of the following GA publications:

Document No.	Title
88A00508A	GA-16/110/220 System Reference Manual
88A00509A	GA-16/110/220 Maintenance Manual
88A00525A	How To Use Your GA-16/220

This addendum describes the differences between the GA-16/220 microcomputer and the GA-16/230/240 microcomputer. The major part of the information supplied in the 110/220 manuals specified above is applicable to the GA-16/230/240 and need not be repeated in this document.

TABLE OF CONTENTS

Section	Title	Page
1	INTRODUCTION	1-1
	1.1 GA-16/220 VS GA-16/230/240	1-1
	1.2 EXTENDED INSTRUCTION SET	1-2
	1.3 INSTALLATION AND SET-UP BOARD CONFIGURATION	1-2
	1.4 MEMORY CONFIGURATION SWITCHES	1-2
2	HYPAK MEMORY WITH ECC	2-1
	2.1 HYPAK MEMORY	2-1
	2.1.1 MEMORY BOARD CONFIGURATIONS	2-1
	2.1.2 MODULE ADDRESS SWITCH S1	2-2
	2.1.3 DEPOPULATED MEMORY SWITCH S2	2-2
	2.1.4 ECC STATUS PRIORITY	2-3
	2.1.5 MEMORY TIMING	2-3
	2.1.5.1 Read	2-3
	2.1.5.2 Write	2-4
	2.1.5.3 Read-Modify-Write	2-4
	2.1.5.4 Refresh Timing	2-4
	2.1.6 INTERFACE PIN ASSIGNMENTS	2-7
	2.1.6.1 Memory Module to Data Bus and MMS	2-7
	2.1.6.2 ECC to Memory Interface	2-7
	2.1.7 MEMORY POWER	2-7
	2.2 ERROR CHECKING AND CORRECTION	2-12
	2.2.1 OPERATING MODES	2-12
	2.2.1.1 Error Correction Mode OFF (Mode 0)	2-13
	2.2.1.2 Normal 1 Mode	2-13
	2.2.1.3 Normal 2 Mode	2-13
	2.2.1.4 Write Data Only Mode (Mode 4)	2-13
	2.2.1.5 Input Status Mode (Mode 3)	2-14
	2.2.2 ECC MODULE STATUS INDICATORS	2-16
	2.2.3 ERROR CORRECTION INTERRUPTS	2-16
	2.2.3.1 Correctable Error Interrupt	2-16
	2.2.3.2 Non-Correctable Error Interrupt	2-16
	2.2.4 STATUS WORDS	2-17
	2.2.4.1 MMS Status Word	2-17
	2.2.4.2 MPP Status Word	2-17
3	MEMORY MANAGEMENT SYSTEM	3-1
	3.1 INTRODUCTION	3-1
	3.2 MMS INTERFACE	3-2
	3.3 MMS FUNDAMENTALS	3-4
	3.4 MEMORY MAPPING	3-11
	3.4.1 MAP CELL FORMAT	3-13
	3.4.2 MAP DATA WRITING	3-14
	3.4.3 MMS MAP PAGES 0 AND 63	3-15

Section	Title	Page
3.5	MMS INSTRUCTIONS	3-16
3.5.1	MAP LOADING.	3-16
3.5.1.1	Map Reading.	3-16
3.5.1.2	Status	3-18
3.5.1.3	Status Reading	3-22
3.5.2	SET COMMAND VALUE.	3-23
3.5.2.1	T State/M1 State (I)	3-26
3.5.2.2	MSTAL (NI)	3-26
3.5.2.3	Enable DMA Mapping and Page Fault (DPF)	3-27
3.5.2.4	DMA Write Protection (DPP)	3-27
3.5.2.5	DMA Memory Error Detection (DME)	3-27
3.5.2.6	Parity Error Simulation (GU, GL)	3-28
3.5.2.7	Change State	3-28
3.5.2.8	Enable Program Mapping and Page Fault Detection (PF).	3-28
3.5.2.9	Enable Program Protect (PP).	3-29
3.5.2.10	Select Program Protect Map (MAP)	3-29
3.5.2.11	Enable Program Memory Error Detection (PME).	3-29
3.5.2.12	Enable Program Timeout	3-29
3.5.3	INDEXED, REAL MEMORY ADDRESSING.	3-30
3.5.4	SINGLE-CYCLE INSTRUCTIONS.	3-31
3.5.5	CLEAR MMS INSTRUCTION.	3-31
3.5.6	ECC MODE SELECTION	3-31
3.6	MMS INTERRUPTS	3-32
3.6.1	T STATE.	3-32
3.6.2	M STATE.	3-32
3.6.3	MSTAL.	3-32
3.6.4	MMS NON-INHIBITABLE (NI) INTERRUPT	3-33
3.6.4.1	General.	3-33
3.6.4.2	Recovery Considerations.	3-34
3.6.5	NI INTERRUPT PROGRAM OPTIONS	3-35
3.6.5.1	Program Timeout (PTO).	3-35
3.6.5.2	Page Fault Interrupts (PFF or PF).	3-36
3.6.5.3	Program Protect Violation (PP)	3-36
3.6.5.4	Memory Error (ME).	3-36
3.7	MEMORY PROTECT	3-37
3.8	DMA CONSIDERATIONS	3-37
3.9	PAGE-TO-MEMORY ADDRESS CONVERSION.	3-38
3.10	SUMMARY OF MMS COMMANDS.	3-38

LIST OF ILLUSTRATIONS

Number	Title	Page
1-1	GA-16/230 in Compact Chassis	1-3
1-2	GA-16/230 in Jumbo Chassis	1-4
1-3	GA-16/240 in Compact Chassis	1-5
1-4	GA-16/240 in Jumbo Chassis	1-6
1-5	HYPAK Memory Board 31D03055A (No ECC)	1-7
1-6	1/2MB ECC Board, 31D03061A	1-8
1-7	MMS Module, 31D03076A.	1-9
2-1	Read Cycle	2-5
2-2	Write Cycle.	2-5
2-3	Read-Modify-Write Cycle.	2-6
2-4	MMS Interrupt Status, Word 0	2-18
2-5	MPP Status Word.	2-19
3-1	Memory Management System Architecture.	3-3
3-2	Transparent Mode	3-4
3-3	Maximum Memory Size.	3-6
3-4	Simplified Memory and Operating System	3-7
3-5	Simplified Map Function.	3-9
3-6	MMS Dynamic Address Translation (Mapping).	3-12
3-7	Map Cell Format.	3-13
3-8	Map Loading.	3-14
3-9	Indexed Real Memory Address Computation.	3-15
3-10	MMS Status Word.	3-19
3-11	MMS Interrupt Status, Word 0	3-20
3-12	Status Word 4.	3-22
3-13	MMS Command Value.	3-24
3-14	Jumper For PTO Timing.	3-36
3-15	MMS Command Summary	3-39

LIST OF TABLES

Number	Title	Page
1-1	GA-16/230/240 Configurations	1-10
1-2	Switches and Indicators for Memory, ECC and MMS Modules	1-13
2-1	Data Bus Pin Assignments (P1)	2-8
2-2	MMS Pin Assignments (J5)	2-10
2-3	GA-16/230/240 256K x 18 x 22 (31D03055A, 31D03061A)	2-11
2-4	ECC Status Word Decode	2-15
3-1	Instruction Summary for MMS	3-17
3-2	Further Interrupt Status Word 0 Interpretation When Page Fault, Program Protect, or Memory Error Occurs	3-21
3-3	Command Value Description	3-24
3-4	MMS NI Interrupt Summary	3-34
3-5	Page-To-Memory Address Conversion	3-38

1 introduction

1.1 GA-16/220 VS GA-16/230/240

This addendum contains the information necessary for installation and operation of the GA-16/230/240 microcomputer. The GA-16/230/240 is based on the same hardware and software features as the GA-16/220. Major differences are related to the semiconductor memories and are identified below.

The GA-16/220 uses conventional 16K semiconductor memory modules with a total memory capacity of 64K words. MPP is available as an option. HYPAK memory is not available for the GA-16/220.

The GA-16/230 uses the 64K (128K bytes) HYPAK memory. The memory parity and write protect (MPP) option is available separately. A full complement of HYPAK modules on the single memory circuit board provides 128K bytes of 18-bit memory with parity.

The GA-16/240 uses a single memory board with at least 128K bytes; error checking and correction (ECC) and the extended instruction set are included. (Memory parity and write protect are incorporated in ECC.) By inserting HYPAK memory modules into DIP sockets, the memory board capacity can be expanded to 512K bytes in 128K byte increments. A memory management system (MMS) is contained on a single circuit board (31D03076A) that may be installed in a GA-16/240 with 128KB memory; MMS is a prerequisite for a GA-16/240 with any memory larger than 128KB. The memory parity and write protect (MPP) option is available separately for memory up to 128KB.

A GA-16/230/240 can be packaged in a compact or jumbo chassis (nucleus), just as the GA-16/220. In essence, the major difference between the compact and jumbo chassis in a GA-16/230 or GA-16/240 is the power supply and chassis size. The jumbo power supply accepts a range of four AC input voltages; and the larger jumbo chassis provides more I/O slots. Regarding the difference between a GA-16/230 and GA-16/240, the GA-16/240 has ECC in all configurations and increased memory with MMS in most configurations. These differences are demonstrated in the configuration listings, Table 1-1.

Table 1-1, GA-16/230/240 Configurations, specifies the GA-16/230/240 configurations for each chassis and the available expansion kits and options, and identifies a part or assembly number (engineering drawing) for each constituent component.

In summary, this addendum describes the General Automation HYPAK with ECC and the memory management system as used with the GA-16/230/240. The memory parity and write protect feature used with the GA-16/230/240 is described in the GA-16/220 publications.

1.2 EXTENDED INSTRUCTION SET

A GA-16/240 contains a CPU-1 board that has a smaller circuit board attached in a piggyback fashion. This attached piggyback contains the extended instruction set for the CPU-1 board; it enhances CPU operation when the GA-16/240 is used with GA-16/440 software. Although the basic GA-16/230/240 or GA-16/220 operates successfully with 440 software, overhead penalties reduce system efficiency. The extended instruction set ensures optimum performance.

1.3 INSTALLATION AND SET-UP BOARD CONFIGURATION

Installation and set-up procedures for a GA-16/230/240 are essentially the same as described in the GA-16/220 publications. However, circuit board configurations in the compact or jumbo chassis differ for the GA-16/220 and the GA-16/230/240. Figures 1-1 and 1-2 show typical GA-16/230 configurations. Memory module addressing is determined by switches mounted on the module and is independent of module position (slot). There is, however, one constraint relative to slot position for the GA-16/240 memory board(s): Due to the piggyback ECC module, a GA-16/240 memory board requires two slot spaces, hence the adjacent lower number slot must be unoccupied. For example, the GA-16/240 memory board in Figure 1-4 must be installed in slot J15 or J16 rather than J14. The MMS board can be installed in any unoccupied memory slot (CPU section of card cage).

1.4 MEMORY CONFIGURATION SWITCHES

The switches and indicators on the memory board (31D03055A), the piggyback ECC board (31D03061A), and the MMS board (31D03076A) are shown in Figures 1-5, 1-6 and 1-7; and are defined in Table 1-2. All switches on these boards relate to the memory configuration board; namely the memory board (module) address, the memory board ECC priority, and the size of the memory. These memory configuration switches must be set as described in Table 1-2, prior to the start of system operation. A more detailed explanation of the memory configuration switches is provided in the Section 2 and Section 3 text.

With one exception, there are no strapping requirements for setting operating variables on the HYPAK, ECC or MMS boards. The sole strapping requirement relates to the program timeout interval on the MMS board and is explained in Section 3.6.5.1.

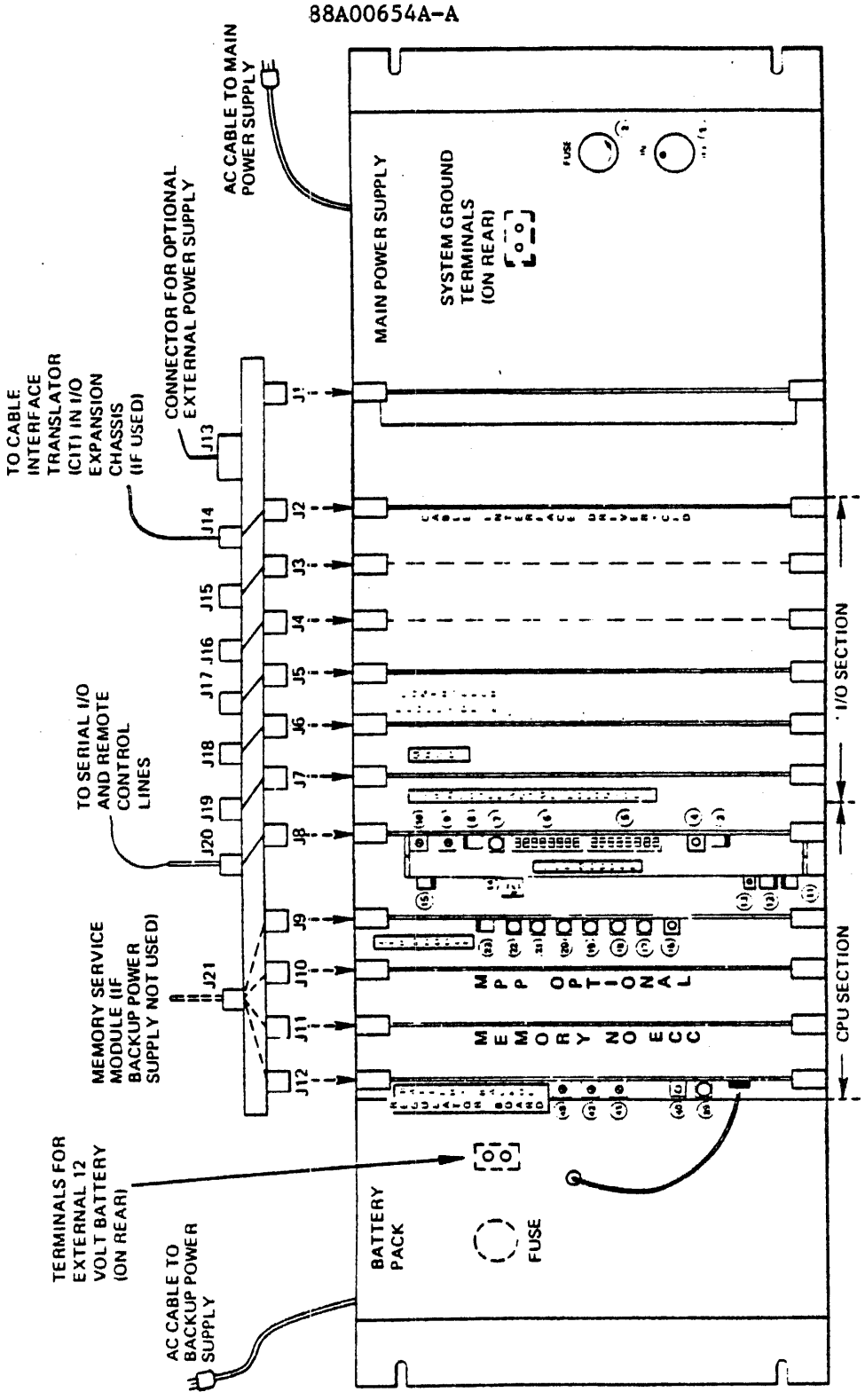


Figure 1-1. GA-16/230 in Compact Chassis

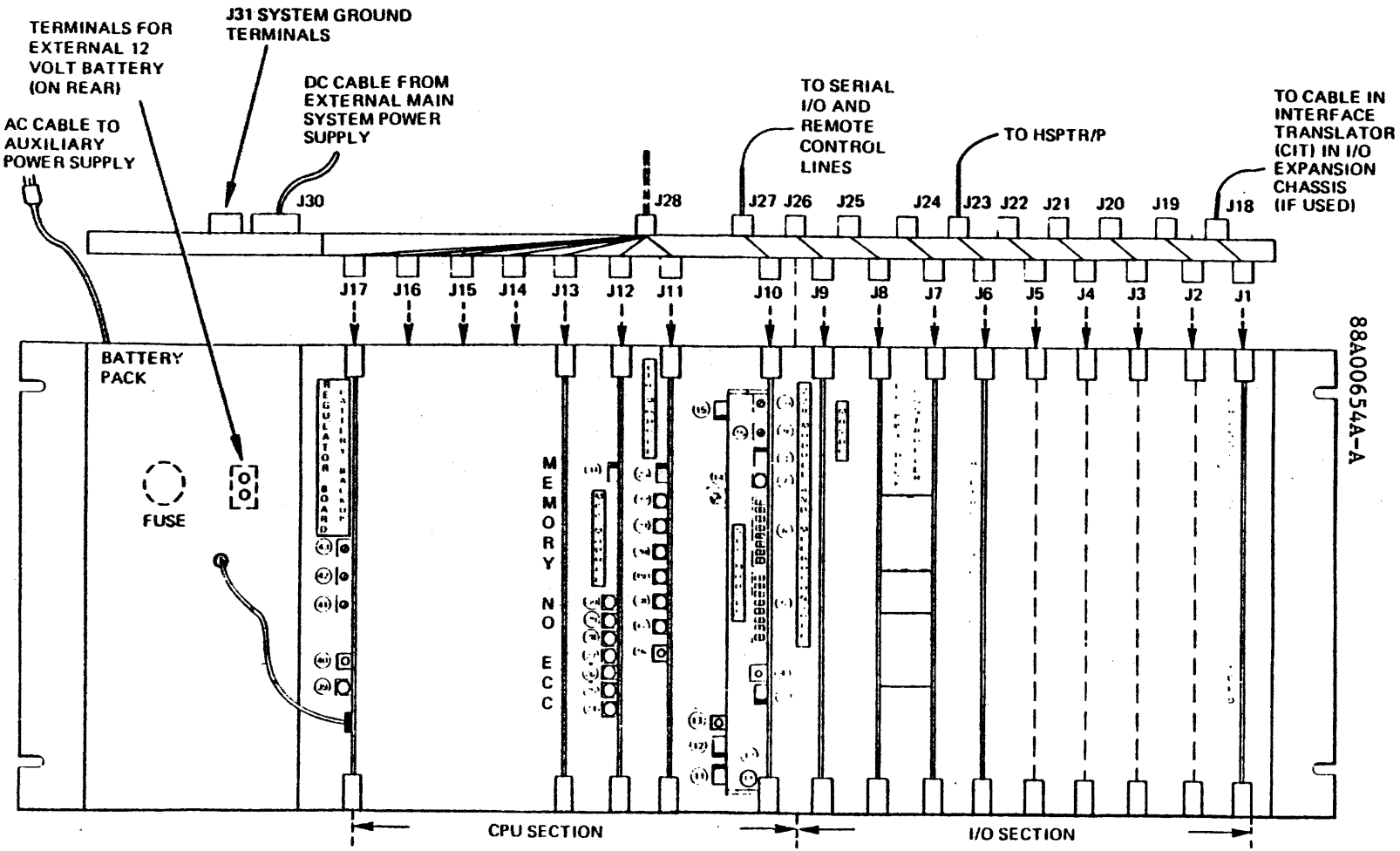
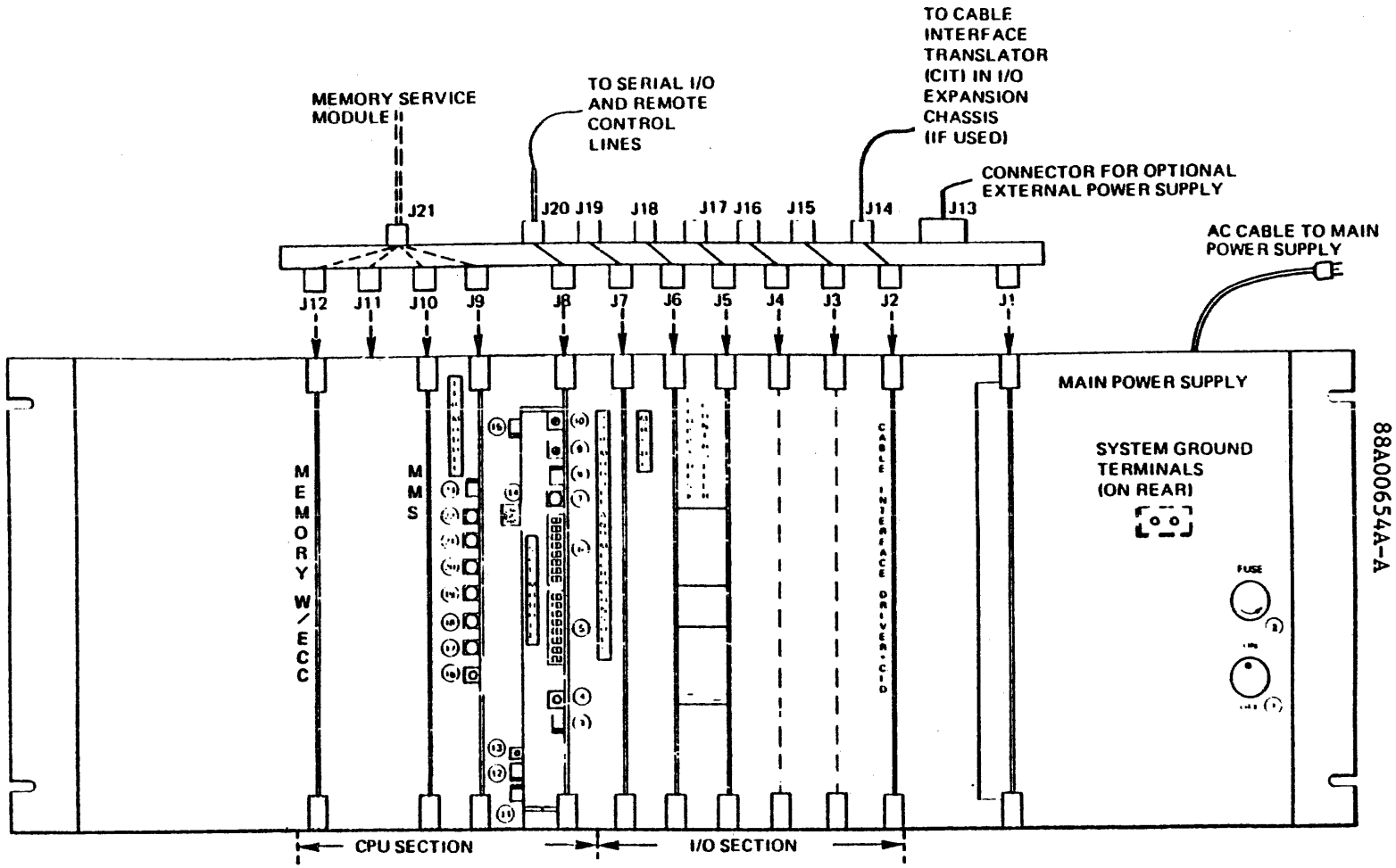


Figure 1-2. GA-16/230 in Jumbo Chassis

*CIRCLED NUMBERS REFER TO TABLES 2-5.



IF MMS IS INSTALLED IN J12, THE MEMORY BOARD WILL BE IN J11.

Figure 1-3. GA-16/240 in Compact Chassis

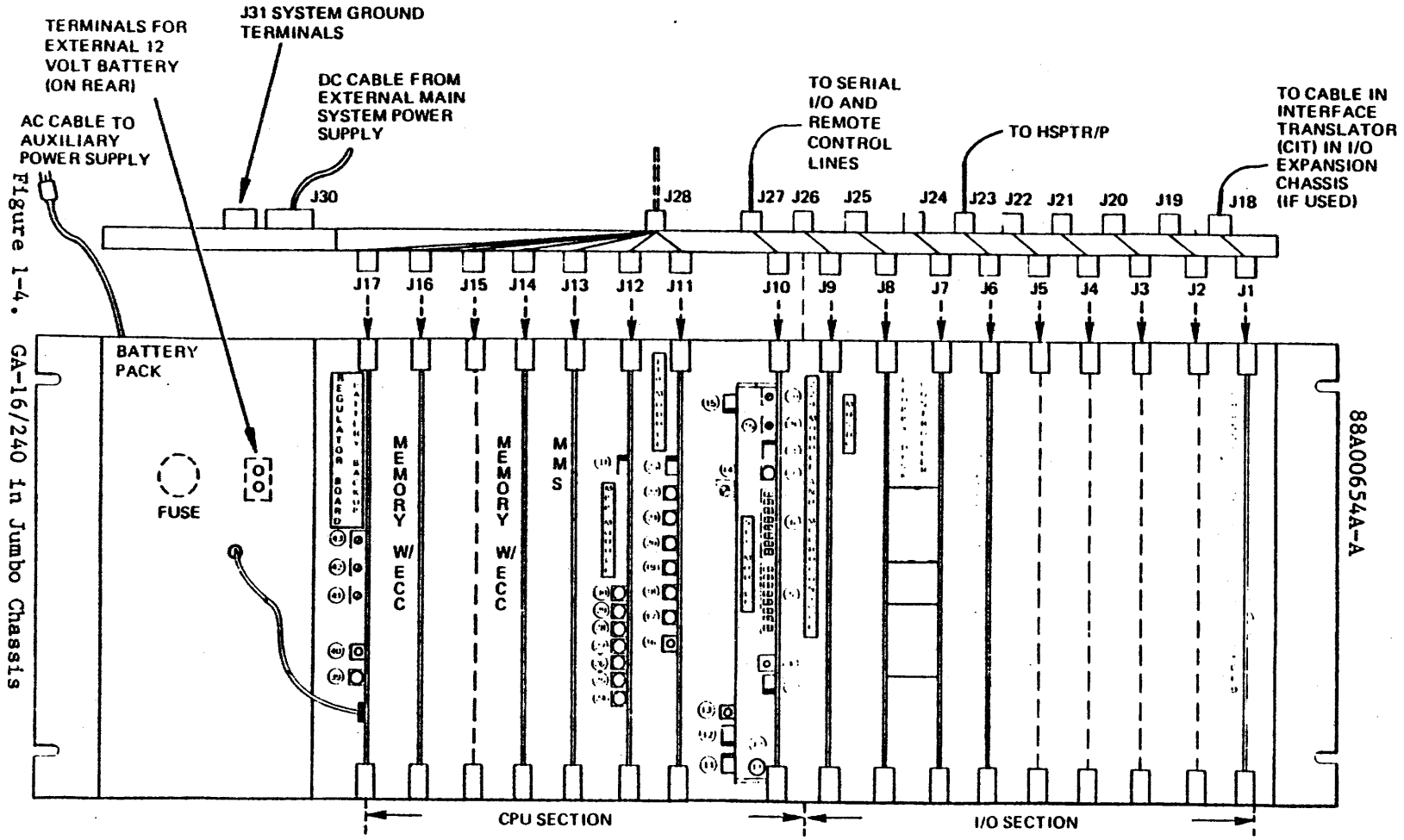


Figure 1-4. GA-16/240 in Jumbo Chassis

*CIRCLED NUMBERS REFER TO TABLES 2-5.

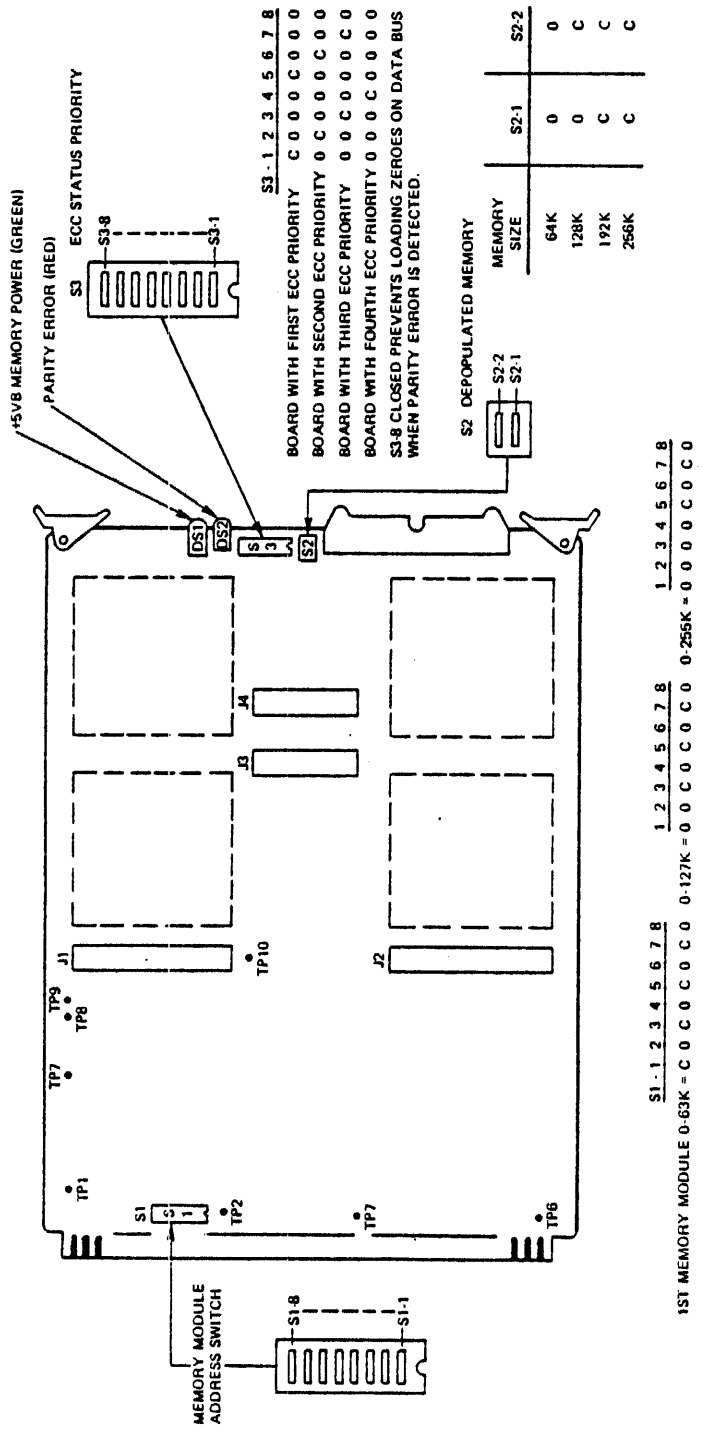
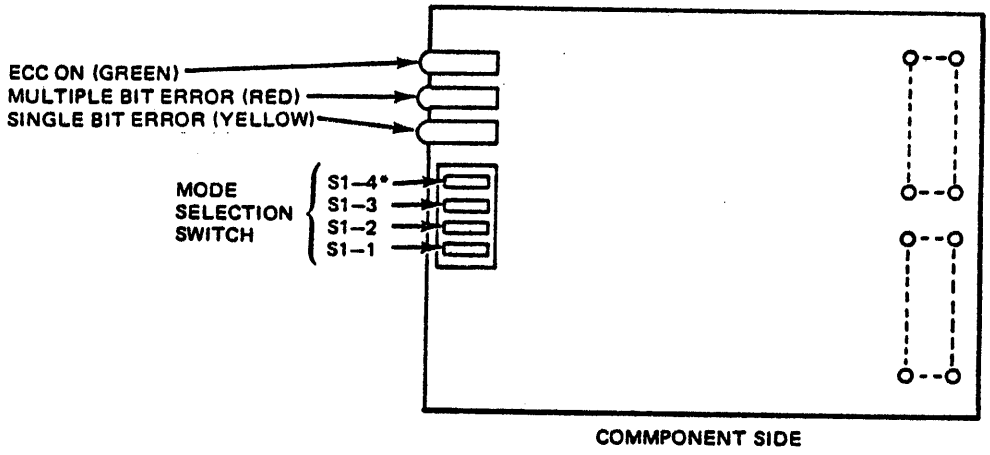


Figure 1-5. HYPAK Memory Board 31D03055A (No ECC)

1/2MB Piggyback ECC Memory

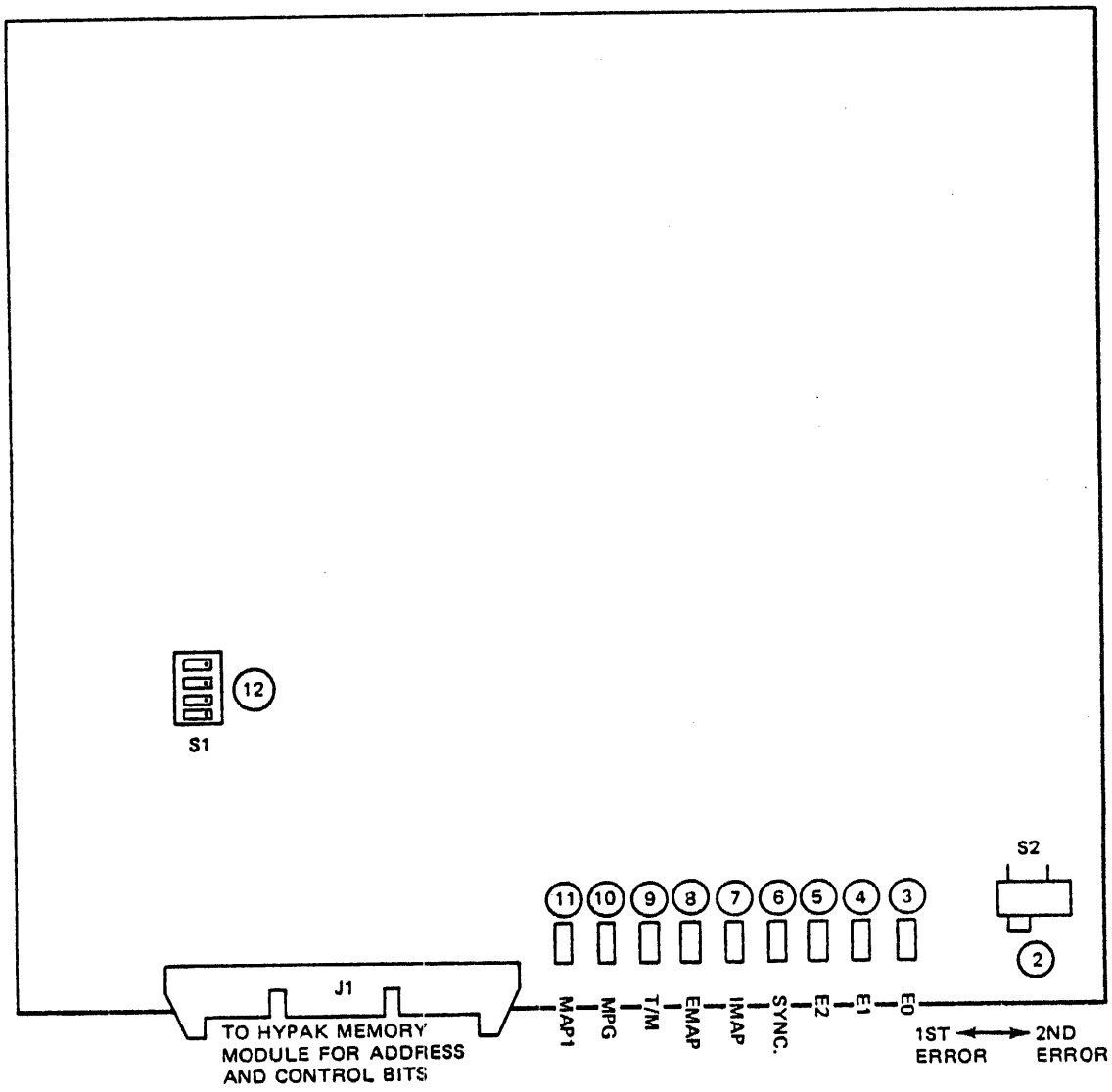


*S1-4 is not used.

Manual Mode Selection Switch Settings

ECC Mode	Function	Switch S1-3	S1-2	S1-1
0	OFF	O	O	O
1	MODE 1	O	O	C
2	MODE 2	O	C	O
3	INPUT STATUS MODE	O	C	C
4	WRITE DATA ONLY	C	O	O

Figure 1-6. 1/2MB ECC Board, 31D03061A



See Table 1-2 for explanation of MMS controls and indicators.

Figure 1-7. MMS Module, 31D03076A

Table 1-1. GA-16/230/240 Configurations (Sheet 1 of 3)

Model Number	Nomenclature	Drawing
GA-16/230 Compact Chassis		
1623-1030	GA-16/230 CPU (115V)	01P0275A
	Compact chassis and power supply	51D00085A 51P00085A 90C02470A
	System console interface with IPL	31D02405A 31P02405A 90C02405A or 31D03116A 31P03116A 90C03116A
	RS232/current loop (TTY) adapter	31C02486A
	Memory service module	31D02301A 31P02301A 90C02301A
	128KB HYPAK memory; 18-bit, non ECC	31D03055A 31P03055A 90C03055A
1623-0050	Memory Parity Protect option	31D02531A 31P02531A 90C02531A
GA-16/230 Jumbo Chassis[1]		
1623-X040	GA-16/230 CPU	01P02075A0Z
	Jumbo chassis and power supply	51D00093A 51P00093A 90C02088A

[1]Except for the changes specified below, the model-assembly-drawing listings for a GA-16/230 with jumbo chassis are the same as shown for a GA-16/230 with compact chassis.

Table 1-1. GA-16/230/240 Configurations (Sheet 2 of 3)

Model Number	Nomenclature	Drawing
GA-16/240 Compact Chassis		
1624-1031	GA-16/240 CPU (115V)	01P02074
	Compact chassis and power supply	51D00085A 51P00085A 90C02470A
	System console interface with IPL	31D02405A 31P02405A 90C02405A or 31D03116A 31P03116A 90C03116A
	RS232/current loop (TTY) adapter	31C02486
	Memory service module	31D02301A 31P02301A 90C02301A
	128KB HYPAK Memory; 22 bit with ECC	31D03055A 31P03055A 90C03055A or 31D03061A 31P03061A 90C03061A
GA-16/240 Jumbo Chassis[2]		
1624-X041	GA-16/240 CPU	01P02074A1Z
	Jumbo chassis and power supply	51D00093A 51P00093A 90C02088A

[2]Except for changes specified below, the model-assembly-drawing listings for a GA-16/240 with jumbo chassis are the same as shown for a GA-16/240 with compact chassis.

Table 1-1. GA-16/230/240 Configurations (Sheet 3 of 3)

Model Number	Nomenclature	Drawing
Conversion Kit: GA-16/230 to GA-16/240[3]		
1624-0724	Conversion kit 16/230 - 16/240	01P02074A
	128KB, ECC piggyback memory module (64Kx4)	31D03061A 31P03061A 90C03061A
	Extended instruction set	31D03005A 31P03005A 90C03005A
1624-0893 Memory Expander for GA-16/240[4]		
1624-0893	Memory expander for GA-16/240	01P02074A
	Memory management system	31D03076A 31P03076A 90C03076A
	128KB, 22-bit HYPAK chips with ECC	01D02081A
1624-0823 Memory Expander For GA-16/240[5]		
1624-0823	128KB, 22-bit HYPAK chips (11) with ECC	01P02074A 01D02081A

[3]The 1624-0724 conversion kit provides a memory piggyback board with two HYPAK chips to convert 128kB of non ECC memory to ECC HYPAK memory. It also provides an extended instruction set that is installed on the CPU-1 board. The kit converts a compact GA-16/230 (1623-1030) to a compact GA-16/240 (1624-1031) or converts a jumbo GA-16/230 (1023-X040) to a jumbo GA-16/240 (1624-X041). ECC is added to memory, but memory size remains the same.

[4]This memory expander consists of a set of 11 HYPAK chips (128KB memory) with ECC and an MMS board.

[5]This expansion kit is applicable to a GA-16/240 if the current memory is 256KB or higher. The kit provides 11 chips comprising 128K bytes with ECC.

Table 1-2. Switches and Indicators for Memory, ECC and MMS Modules (Sheet 1 of 3)

Switch/ Indicator	Function
Memory Board 31D03055A	
DS1-LED IND (11D)	Power-on. Green power-on indicator turns on if +5V is received from the power supply.
DS2-LED IND (11E)	Parity error. Red memory parity error indicator turns on if ECC board detects parity error.
S1-DIP SW (1D)	Module address. S1 assigns the module address as shown in Figure 1-5.
S2-DIP SW (11J)	Depopulated memory. S2 should be set as shown in Figure 1-5.
S3-DIP SW (11G)	ECC status priority. If more than one memory board is installed, S3 selects the response priority when the memory (ECC) boards are interrogated for ECC status. If only one HYPAK memory board is installed, that board should be considered the "first" memory board and should be assigned the highest ECC priority. S3 positions are defined in Figure 1-5.
ECC Board 31D03061A	
DS1-LED IND (1A)	ECC mode select. Green indicator turns on if ECC mode is selected by S1.
DS2-LED IND (1B)	Multiple parity error. Red indicator turns on if multiple parity error is detected by ECC.
DS3-LED IND (1C)	Single parity error. Yellow indicator turns on if single parity error is detected by ECC.
S1-DIP SW (1E)	Manual/ECC mode select. S1 positions are defined in Figure 1-6. Mode definitions are provided in Section 2.

Table 1-2. Switches and Indicators for Memory, ECC and MMS Modules (Sheet 2 of 3)

Switch/ Indicator	Function
MMS Board 31D03076A	
Switch S2 (2)	2-position switch in LEFT position, the indicators (3), (4), (5), (6), (7), and (8) are illuminated by an error which caused an NI interrupt (first error). In the RIGHT position the indicators are illuminated by a second error which can cause a MSTAL condition (refer to Section 3.5.2.2).
E0, E1, E2 LED IND (3) (4) (5)	Indicators are decoded to determine type of interrupt condition which occurred as follows (illuminated = 1): E2 E1 E0 — Condition 0 0 0 — Program Timeout (PTO) 0 1 0 — Page Fault (check sync lamp also) (PF) 0 1 1 — Program Protect (PP) (check sync lamp 6 also) 1 0 1 — Memory Error (ME) (check sync lamp 6 also) These lamps are equivalent to bits 0, 1 and 2 in status word 0 (Section 3.5.1.2).
SYNC LED IND (6)	Synchronous fault indicator. If illuminated concurrently decode of E2, E1, E0 as PF or PP indicates a fault. If not illuminated with PF or PP indicates a DMA/DMT fault. This indicator corresponds with the S/A bit in status word 0. (If illuminated with ME, it indicates an unconnectable program error.)
I MAP LED IND (7)	Error instruction map indicator to determine the instruction map used at the time the interrupt condition was created. only valid if indicator (9) is off. 1 = MAP 1 0 = MAP 0 This indicator corresponds with the IM bit in status word 0 (Section 3.5.1.2).

Table 1-2. Switches and Indicators for Memory, ECC and MMS Modules (Sheet 3 of 3)

Switch/ Indicator	Function
MMS Board 31D03076A	
E MAP LED IND (8)	Error execute map indicator to determine the MAP being used at the time the interrupt condition was created. Only valid if indicator (9) is off. 1 = MAP 1 0 = MAP 0 This indicator corresponds with the EM bit in status word 0 (Section 3.5.1.2).
T/M LED IND (9)	If illuminated indicates MMS in transparent mode when interrupt occurred. This indicator corresponds with the T/M bit of status word 0.
MPG LED IND (10)	Mapping mode indicator. If illuminated indicates that either the CPU or DMA/DMT mapping is enabled.
MAP 1 LED IND (11)	MAP 1 indicator, when lit with (10) indicates MMS is in MAP 1. When off with (10) indicates MAP 0.
Memory Size S1 (12)	Set to indicate maximum memory size so that an overflow condition during indexed/real memory addressing is detected.

Switch Closed	Value	Address Range
1	7	1024K
2,3,4	8	896K
2,3	9	768K
2,4	A	640K
2	B	512K
3,4	C	384K
3	D	256K
4	E	128K
All Open	F	0K

HYPAK memory with ECC 2

The GA-16/230/240 are the enhanced version of the GA-16/220, with the exception of the memory related features noted in Section 1. This section describes two of the three GA-16/230/240 memory related features that are not incorporated in the GA-16/220; namely, the HYPAK memory and error checking and correction (ECC). (The third feature, the memory management system, is described in Section 3.)

2.1 HYPAK MEMORY

The standard GA-16/230/240 HYPAK memory module is a single printed circuit board (31D03055A) containing 128, 256, 384, or 512 kilobytes of random access memory. Four connectors are provided on the circuit board for connection to the piggyback ECC module. A memory board can be used with or without an ECC module up to 128KB. Memory is added to the basic 128KB configuration in increments of 128KB by inserting HYPAK memory chips in DIP sockets.

2.1.1 MEMORY BOARD CONFIGURATIONS

The 31D03055A memory board and the 31D03061A ECC board come in four versions, as shown:

31D03055A21	64K x 18	Memory Board
31D03061A24	64K x 4	ECC Board
31D03055A41	128K x 18	Memory Board
31D03061A44	128K x 4	ECC Board
31D03055A51	192K x 18	Memory Board
31D03061A54	192K x 4	ECC Board
31D03055A61	256K x 18	Memory Board
31D03061A64	256K x 4	ECC Board

The memory board and the ECC board both use the General Automation 70A00780A HYPAK memory module in a 28-pin DIP which contains 65,536 x 2 bits of memory. The HYPAK module incorporates eight 16K x 1 memory chips which are interconnected and used as a 64K x 2 bit configuration. The memory chips used in the HYPAK memory modules are 16K x 1 bit type TMS 4116 or MB 8216 chips. (Refer to GA specification 61A00016A for electrical parameters and test procedure.)

Refresh circuitry for the semiconductor memory is incorporated on the memory board and provides an internal refresh cycle issued every 16 usec. A refresh cycle has priority over a memory cycle. This guarantees that each of the 128 row addresses will be refreshed every 2 msec.

When the ECC board is not plugged in, the memory operates as a 16-bit data word plus 2 parity bits, one for each byte. During a write cycle, two parity bits are generated from the data present on the data bus and then stored in the memory. During a read cycle, two parity bits are generated from the data read from memory and compared with the two parity bits stored in the memory during the write cycle. If any of the parity bits do not match, a parity error is then detected. When a parity error is detected, (no MPP), the data bus is disabled (all zeros) until the error is cleared with system reset.

2.1.2 MODULE ADDRESS SWITCH S1

Address assignments for the memory module are shown in Figure 1-5 (Section 1). Logic drawing 90C03055A, sheet 4, shows how memory module address switch S1 generates an active MSEL memory select signal when an S1 setting corresponds to the address on MA16- through MA19- of the memory address bus. MSEL enables the read or write select gating and addressing logic for the memory chips, enabling the addressed memory module to respond to an MREQ memory request from the CPU.

2.1.3 DEPOPULATED MEMORY SWITCH S2

Depopulated memory board switch S2 is set to conform to the memory board capacity (chip population) as shown in Figure 1-5. If S2 conforms to the memory size, address lines MA16- and MA17- can develop row address strobes RAS0- through RAS15- to initiate memory access (refer to 90C03055A, sheet 7).

2.1.4 ECC STATUS PRIORITY

If the ECC board is operating in the Input Status Mode, the CPU can execute a read at memory address 0 and thereby read the ECC status word from each ECC board (memory module with piggyback ECC). The ECC status words, one per board, are consecutively read in a priority sequence established by ECC status priority switch S3. Figure 1-5 shows how S3 is set to establish 1st, 2nd, 3rd or 4th priority for the associated board.

NOTE

Only applicable if all memories have an error. Refer to Section 2.2.1.5.

Drawing 90C03055A, sheet 14, shows S3 and the priority selection lines. Priority lines PRL1-, PRL2- and PRL3- are on the MMS address line that connects all memory boards. With contacts 1 and 5 closed (first priority board) an active MPSI- is sent to the piggyback ECC board on J1-50. MPSI- enables the ECC board to place its status word and address on the In-Bus to the CPU. When the In-Bus data has been read, the ECC board sends an active MPSO- term to the memory module on J1-48. This signal is routed through S3-5; the resulting PRL1- is sent to the other memory boards. The second priority module has S3-2 and S3-6 closed to generate MPSI- and MPSO- with the results described for the first board.

2.1.5 MEMORY TIMING

Figures 2-1, 2-2 and 2-3 contain timing diagrams for the memory read, write, and read-modify-write cycles.

2.1.5.1 Read

Memory request signal (MREQ-) from the CPU must remain low (active) until memory ready signal (MRDY+) is returned high (inactive) or within 1.6usec, and must remain off for 100ns minimum.

All address lines (ADXX-) must be valid for a minimum of 20ns on the address bus before the leading edge of memory request (MREQ-) and remain valid for 175ns minimum on the bus.

Data on the data bus (TDXX-) must be valid at the leading edge of memory data ready signal (MRDY+) and remain valid until memory request signal (MREQ-) goes high.

2.1.5.2 Write

Memory request signal (MREQ-) and address lines requirements are the same as for the read timing.

Memory write signal (MWRE-) not using Stop Read (STRD-), must go low within 40nsec maximum after memory request signal (MREQ-) goes low and go high when MREQ- goes high.

Data from the memory is inhibited from the data bus during the write cycle by means of a tri-state bus drivers disabled with memory write signal (MWRE-) activated (low).

2.1.5.3 Read-Modify-Write

Memory request signal (MREQ-) must remain low until memory data ready signal (MDRY+) is returned high after the write portion of the cycle.

Stop read signal (STRD-) may go low after memory data ready (MDRY+) is returned high and remain low until memory write signal (MWRE-) goes low. Refer to Figure 2-3.

Memory write signal (MWRE-) may go low 100nsec minimum after stop read signal (STRD-) and should go high when memory request signal (MREQ-) goes high.

The data on the data bus (TDXX) must be valid between the trailing edge of memory data ready signal (MDRY+) (low to high transition) and the leading edge of stop read signal (STRD-).

The complete Read-Modify cycle must be terminated within 2 usec maximum from the time of memory request signal (MREQ-) is activated.

2.1.5.4 Refresh Timing

The refresh circuit provides a 16usec refresh timer. If the timer becomes active during a memory cycle, the refresh is enabled at the end of the memory cycle.

If a memory request signal (MREQ-) is activated during a refresh cycle, the cycle will be delayed until the refresh cycle is completed. Each refresh cycle refreshes each of the 128 row addresses at least once every 2msec.

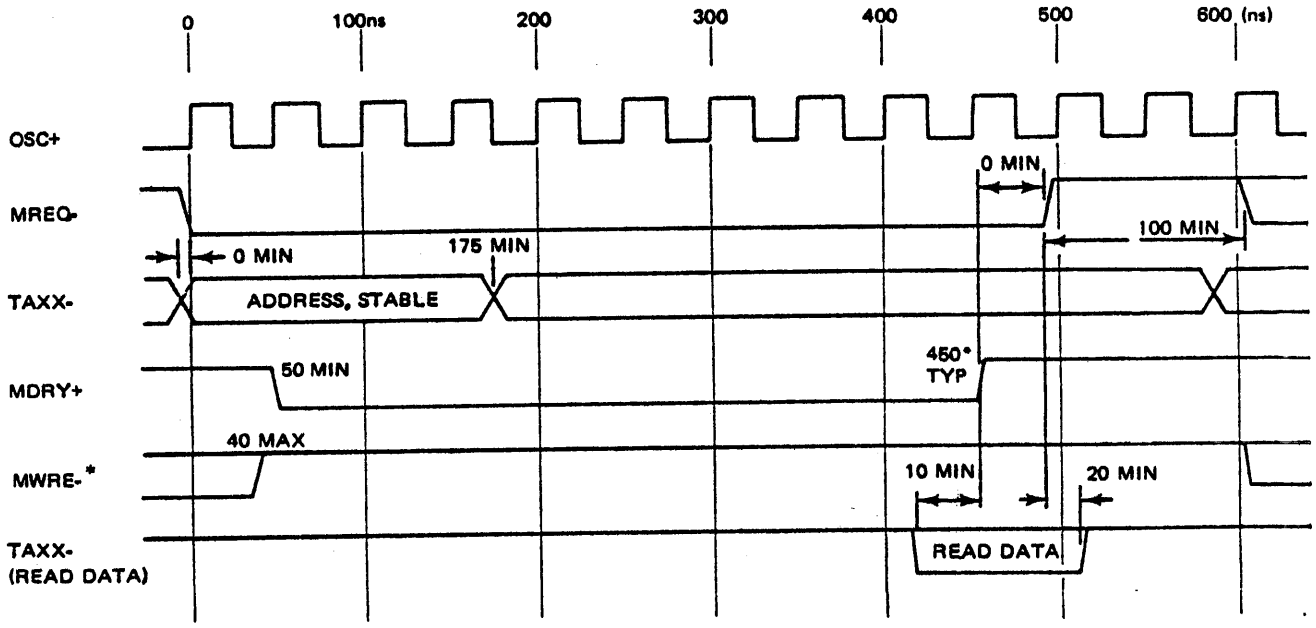
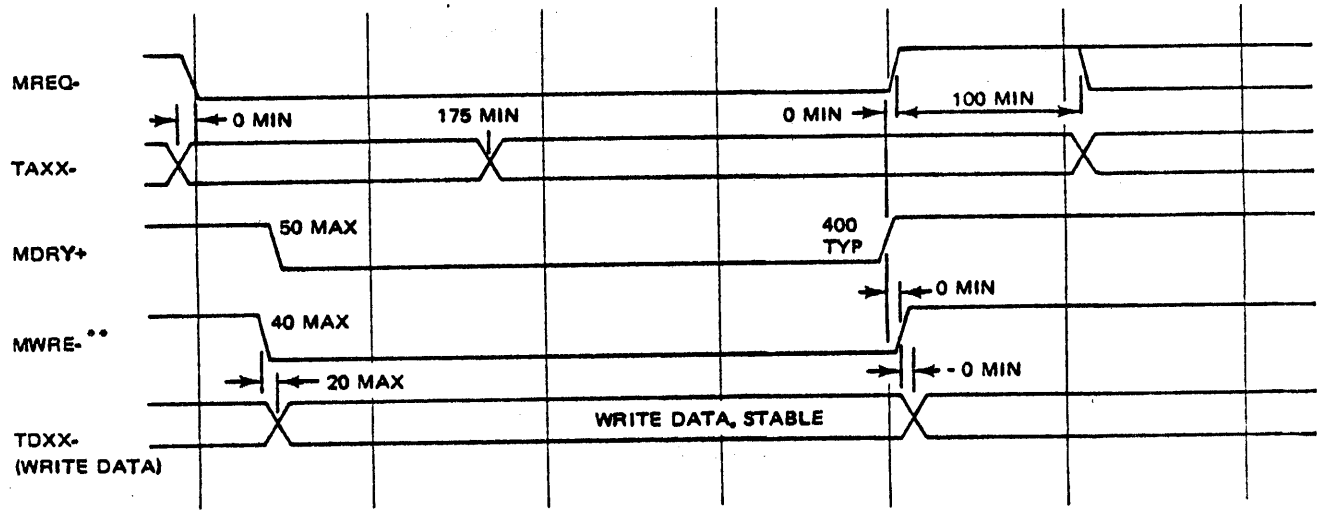


Figure 2-1. Read Cycle



* 530 TYP. FOR CYCLE HAVING CORRECTED DATA.
** STOP READ, STRD – IS HIGH DURING READ OR WRITE CYCLE.

Figure 2-2. Write Cycle

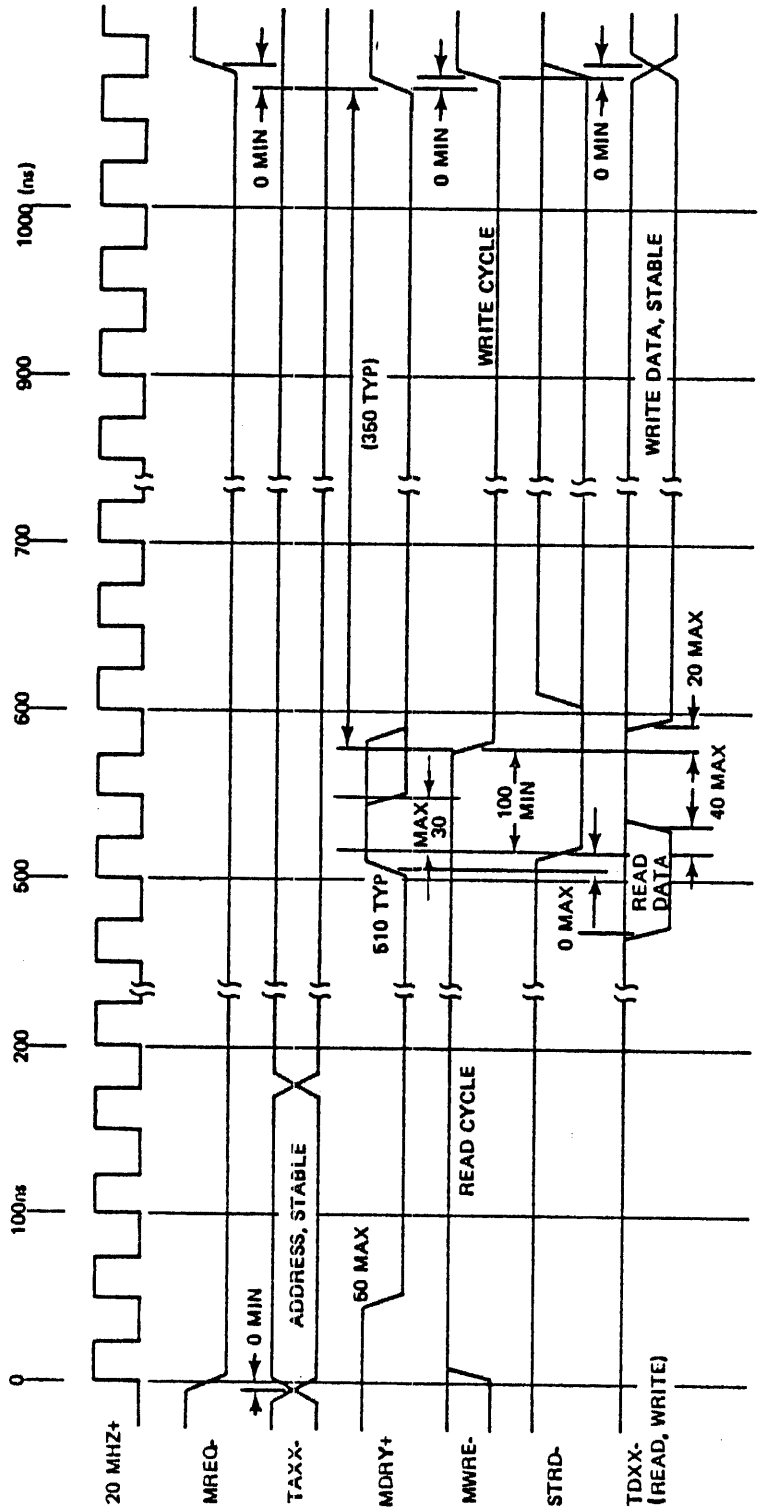


Figure 2-3. Read-Modify-Write Cycle

2.1.6 INTERFACE PIN ASSIGNMENTS

2.1.6.1 Memory Module to Data Bus and MMS

Table 2-1 gives pin assignments and term definitions for the P1 edge connector that connects the memory module to the CPU data (I/O) bus. Table 2-2 gives pin assignments for the J5 connector that connects the memory module to the MMS board (on the GA-16/240).

2.1.6.2 ECC to Memory Interface

The error correction board assembly 31D02061A is interfaced with the memory board assembly 31D03055A via four connectors, J1 through J4. J1 and J2 are 50 pins and J3 and J4 are 12 pin connectors.

The ECC board assembly plugs in the back of the memory board non-components side. The unique lines which were not listed in Tables 2-1 and 2-2 are identified below:

ERROR CLOCK	J1-34	ECM-	ECM- (Error Correction Mode) generates an extra clock to strobe corrected data into data register.
CORRECTABLE or NON-CORRECTABLE ERROR	J4-9	ECCER+	Error flag used only when no MPP or MMS is used. Forces the tri-state data bus to all zeros. Cleared with system reset (SYRT-) only.
DATA CORRECTION BITS	ERB00 -	ERB15	Single bit error correction bits.
WRITE ECC BITS	WECC-	J1-33	Read/write line for the ECC bits. During a write data only mode, WECC- line disable write into ECC bits.

2.1.7 MEMORY POWER

Table 2-3 defines power requirements for all configurations of a GA-16/230/240 memory board.

Table 2-1. Data Bus Pin Assignments (P1) (Sheet 1 of 2)

POWER PINS:	1,140		+5V Supply
	71		+5VB Battery Back-Up
	81		+12V Supply
	82		-5V Supply
	3,4,137,138		Ground
DATA BUS:	94	TD00-	
	95	TD01-	
	96	TD02-	
	97	TD03-	
	98	TD04-	
	100	TD06-	
	101	TD07-	
	102	TD08-	
	103	TD09-	
	104	TD10-	
	105	TD11-	
	106	TD12-	
	107	TD13-	
	108	TD14-	
	109	TD15-	
	74	TD16-	Parity Bit Byte 0
	76	TD17-	Parity Bit Byte 1
ADDRESS BUS:	110	TA00-	Least Significant Bit
	111	TA01-	
	112	TA02-	
	113	TA03-	
	114	TA04-	
	115	TA05-	
	116	TA06-	
	117	TA07-	
	118	TA08-	
	119	TA09-	
	120	TA10-	
	121	TA11-	
	122	TA12-	4K Select
	123	TA13-	8K Select
	124	TA14-	16K Select
	125	TA15-	32K Select
MEMORY REQUEST:	126	MREQ-	Initiates the memory cycle.

Table 2-1. Data Bus Pin Assignments (P1) (Sheet 2 of 2)

MEMORY PROTECT	45	MEMPR-	Inhibits Memory Cycle
MEMORY WRITE	127	MWRE-	Low for a write cycle.
STOP READ	128	STRD-	Aborts read cycle.
MEMORY DATA READY	129	MDRY+	Indicates data valid for read and memory response for write.
OSCILLATOR	69	OSC+	20 MHz processor clock
MEMORY PARITY AND PROTECT	63	MPP-	Low (ground) when memory protect option is used.
SYSTEM RESET	46	SYRT-	System resets when low. Used for power up/down sequence.
NON-INHIBITABLE INTERRUPT 1	77	NIIR1-	When low, resets parity error latches.
DATA PARITY ERROR BUS	50	MEMPY-	Low indicates data parity error. It also disables data from the data bus. Used only when no MMS or MPP is in the system.
FORCE PARITY ERROR LOW BYTE	68	FPBL-	Low causes wrong parity to be written byte 0.
FORCE PARITY ERROR HIGH BYTE	70	FPBU-	Low causes wrong parity to be written byte 1.
NON-CORRECTABLE ERROR	83	NCERR-	Used with ECC only. Causes error interrupt with non-correctable error.
CORRECTABLE ERROR	84	COERR-	Used with ECC only. Causes error interrupt with correctable errors and ECC mode selection.
ECC MODE CONTROL LINES	25 26 27	ECMD0- ECMD1- ECMD2-	Select one of five ECC modes.
ECC MODE CLOCK	52	WCE-	Used to clock in ECMD0 - ECMD2 into mode register.

Table 2-2. MMS Pin Assignments (J5)

MEMORY MANAGEMENT UNIT	J5-3	MMS-	Low (ground) when memory management option is used.
PRIORITY LINES	J5-5 J5-7 J5-9	PRL1- PRL2- PRL3-	Priority lines, used with memory management option, to sequentially propagate error status through memory modules.
ADDRESS ZERO	J5-1	ADZS-	Low (active) when addresses A00-A15 are zero and ECC error status mode is activated.
MEMORY WRITE	J5-37	MMWRE-	Low for a write command from the MMS.
EXTENDED MEMORY ADDRESS LINES	J5-11 J5-13 J5-19 J5-21	MA16- MA17- MA18- MA19-	64K Select 128K Select 256K Select 512K Select
ERROR CORRECTION ENABLED	J5-2	ECCEN-	Low (ground) when ECC unit is plugged in.
MEMORY REQUEST	J5-35	MMREQ-	Low for initiating a memory cycle from the MMS.
ADDRESS LINES FROM MMS (MEMORY MANAGEMENT OPTION)	J5-29 J5-23 J5-25 J5-27 J5-31 J5-37	MA10- MA11- MA12- MA13- MA14- MA15-	Used for the ECC status word.

Table 2-3. GA-16/230/240 256K x 18 x 22 (31D03055A, 31D03061A)
Memory System at Room Temperature (25°C)

VERSIONS: 18 BIT = PARITY 22 BIT = ECC	+5VC		+5VB			+12VB			-5VB		
	OP	STDBY	OP	STDBY	BATT OP	OP	STDBY	BATT OP	OP	STDBY	BATT OP
64K x 18 BITS	1.3A	1.2A	.76A	.72A	.75A	.27A	.08A	.08A	1 ma	.6 ma	.6 ma
64K x 22 BITS	2.4A	2.3A	.89A	.81A	.85A	.34A	.1A	.1A	1 ma	.6 ma	.6 ma
128K x 18 BITS	1.3A	1.2A	.76A	.72A	.75A	.36A	.17A	.17A	1 ma	.6 ma	.6 ma
128K x 22 BITS	2.4A	2.3A	.89A	.81A	.85A	.43A	.2A	.2A	1 ma	.6 ma	.6 ma
192K x 18 BITS	1.3A	1.2A	.76A	.72A	.75A	.44A	.25A	.25A	1 ma	.6 ma	.6 ma
192K x 22 BITS	2.4A	2.3A	.89A	.81A	.85A	.53A	.3A	.3A	1 ma	.6 ma	.6 ma
256K x 18 BITS	1.3A	1.2A	.76A	.72A	.75A	.52A	.33A	.33A	1 ma	.6 ma	.6 ma
256K x 22 BITS	2.4A	2.3A	.89A	.81A	.85A	.63A	.4A	.4A	1 ma	.6 ma	.6 ma
REFERENCE: 16K x 18 BITS (31D02578A)	.83A	.83A	.42A	.42A	.42A	.93A	.06A	.06A	4 ma	.4 ma	.4 ma

2.2 ERROR CHECKING AND CORRECTION

The GA-16/230/240 ECC feature provides single-bit error correction and multiple-bit error detection. The ECC module is a single printed circuit board which plugs into four connectors on a 64Kx18 bit memory module. To accomplish error detection and/or correction, the ECC module generates and stores six "check bits" per data word during a write operation. During a read operation, these check bits are exclusive-ORed with the new check bits (generated during the read). If the result is zero, all data was read correctly. A single bit error results in a non-zero result, with the bit position in error specified by the decode of the result. A multiple bit error also generates a non-zero result, however the decode does not represent a valid bit position.

NOTE

Upon initial power-up, all ECC bits are incorrect. All memory locations must be initialized by a store operation before the ECC is enabled. Failure to do so will cause false error indications.

2.2.1 OPERATING MODES

The error correction board operates in five different modes. The modes are selected by executing a DTOR/M to the MMS (or MPP) with the appropriate mode coded in bits 4, 5 and 6 of the output data words. In addition, manual switches on the ECC boards provide manual selection of the five operating modes. (Manual selection should aid in testing or troubleshooting the system.) Mode selection is received by the ECC board via the ECMD0 - ECMD2 control lines. Coding is as follows:

ECMD0	ECMD1	ECMD2	MODES
L	L	L	ECC OFF
L	L	H	NORMAL 1 MODE
L	H	L	NORMAL 2 MODE
L	H	H	INPUT STATUS MODE (MODE 3)
H	L	L	WRITE DATA ONLY MODE (MODE 4)

Figure 1-6 (Section 1) explains the manual mode selection.

2.2.1.1 Error Correction Mode OFF (Mode 0)

This mode, when selected, disables the error correction capability. System reset will select the OFF mode.

2.2.1.2 Normal 1 Mode

This mode, when selected, enables the error correction circuitry to generate an interrupt on both correctable (single-bit) and non-correctable (multiple-bit) errors. The applicable indicator on the ECC board turns on if an error is detected (Figure 1-6).

2.2.1.3 Normal 2 Mode

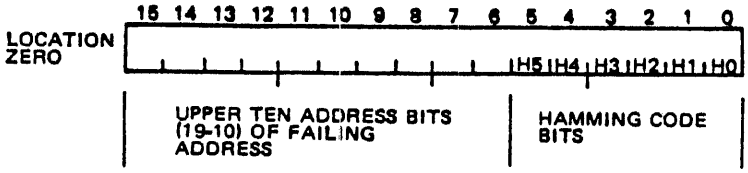
This mode, when selected, allows the error correction circuitry to generate an interrupt on non-correctable errors only; single-bit errors are corrected. The applicable indicator on the ECC board turns on if an error is detected (Figure 1-6).

2.2.1.4 Write Data Only Mode (Mode 4)

This mode is restricted to diagnostic and test purposes. When selected, new check bits are generated, but not stored in memory. The original check bits remain in the memory.

2.2.1.5 Input Status Mode (Mode 3)

When the input status mode is selected (by the program), each ECC board will generate a status word if it has detected a single or multiple bit error on the associated memory board. The input status mode causes a status priority signal from the CPU to be propagated through each error correction board. The first board with an error will trap the status priority signal. If the CPU executes a read for memory location zero while input status mode is selected, the ECC board that trapped the status priority signal will place the following status word on the data bus:



Following the read execution, the status priority signal is propagated to the next lower priority ECC board. Each ECC board with a detected error blocks propagation of the priority status signal and responds to a CPU read of location zero, just as described for the first board to trap the priority signal. After transmitting the status priority signal, the CPU continues to read location zero until a status word with all zeros is received, indicating there are no more ECC boards with error status.

Table 2-4 defines the ECC status word bits, and shows the decode of hamming bits 0-5 to identify the error type (single or multiple bit), and the identity of the erroneous bit for a single bit error.

NOTE

A single bit failure of hamming bit 5 occurring in the first 1K of memory will result in a status word of X'0000', rather than a status word with an address and a hamming code. Therefore, status should be checked twice if X'0000' is received on the first status read. (The MSS/MPP status can also be read in the conventional manner to identify an error condition in a memory board with ECC.)

Table 2-4. ECC Status Word Decode

ECC STATUS DECODE FOR 240 CPU

BITS 15-----6 IDENTIFY A 1K BLOCK OF MEMORY
WHERE AN EC ERROR HAS OCCURED.

BITS 5-----0 IDENTIFY THE TYPE OF ERROR, MULTI OR SINGLE,,
AND SPECIFY THE ERRONEOUS BIT FOR SINGLE-BIT ERRORS.

In the following tabulation for hamming bits 5-0:

"B" IDENTIFIES A DATA WORD BIT

"H" IDENTIFIES A HAMMING CHECK BIT

"MULTI" IDENTIFIES A MULTIPLE-BIT ERROR CODE

BITS 5---0							BITS 5---0						
5	4	3	2	1	0		5	4	3	2	1	0	
0	0	0	0	0	0	H5	0	1	0	0	0	0	H4
0	0	0	0	0	1	H0	0	1	0	0	0	1	B8
0	0	0	0	1	0	H1	0	1	0	0	1	0	B9
0	0	0	0	1	1	B0	0	1	0	0	1	1	B10
0	0	0	1	0	0	H2	0	1	0	1	0	0	B11
0	0	0	1	0	1	B1	1/0	1	0	1	0	1	MULTI
0	0	0	1	1	0	B2	0	1	0	1	1	0	B12
0	0	0	1	1	1	B3	1/0	1	0	1	1	1	MULTI
0	0	1	0	0	0	H3	0	1	1	0	0	0	B13
0	0	1	0	0	1	B4	0	1	1	0	0	1	B14
0	0	1	0	1	0	B5	1/0	1	1	0	1	0	MULTI
1/0	0	1	0	1	1	MULTI	1/0	1	1	0	1	1	MULTI
0	0	1	1	0	0	B6	1/0	1	1	1	0	0	MULTI
0	0	1	1	0	1	B7	1/0	1	1	1	0	1	MULTI
1/0	0	1	1	1	0	MULTI	0	1	1	1	1	0	B15
1/0	0	1	1	1	1	MULTI	1/0	1	1	1	1	1	MULTI

For example, the fourth tabulation in the left-hand column contains a 1 in hamming bits 0 and 1, and 0s in bits 2, 3, 4 and 5. This hamming code specifies an error in data bit 0 (B0).

2.2.2 ECC MODULE STATUS INDICATORS

The ECC module contains three indicators which indicate the presence of the ECC module and error status. The yellow indicator is illuminated when a correctable error has occurred. The red indicator is illuminated when a non-correctable error has occurred.

NOTE

When a non-correctable error occurs, the yellow indicator may or may not be illuminated.

The green indicator is illuminated when the ECC is on. Figure 1-6 shows the location of these indicators.

2.2.3 ERROR CORRECTION INTERRUPTS

The following interrupts are generated by the error correction board:

- Correctable Error Interrupt
- Non-Correctable Error Interrupt

2.2.3.1 Correctable Error Interrupt

An error correction from an ECC board will cause the MMS/MPP to generate a noninhibitable interrupt.

2.2.3.2 Non-Correctable Error Interrupt

The ECC board will generate incorrect byte parity on the erroneous word to signify a non-correctable error. Response to this depends on whether the GA-16/440 is equipped with MMS or MPP. The MMS will respond by generating a non-inhibitable interrupt with a CPU reset. The MPP will generate a non-inhibitable interrupt without a CPU reset.

2.2.4 STATUS WORDS

The occurrence of both non-correctable and correctable interrupts may be distinguished by examining the MPP or MMS status word.

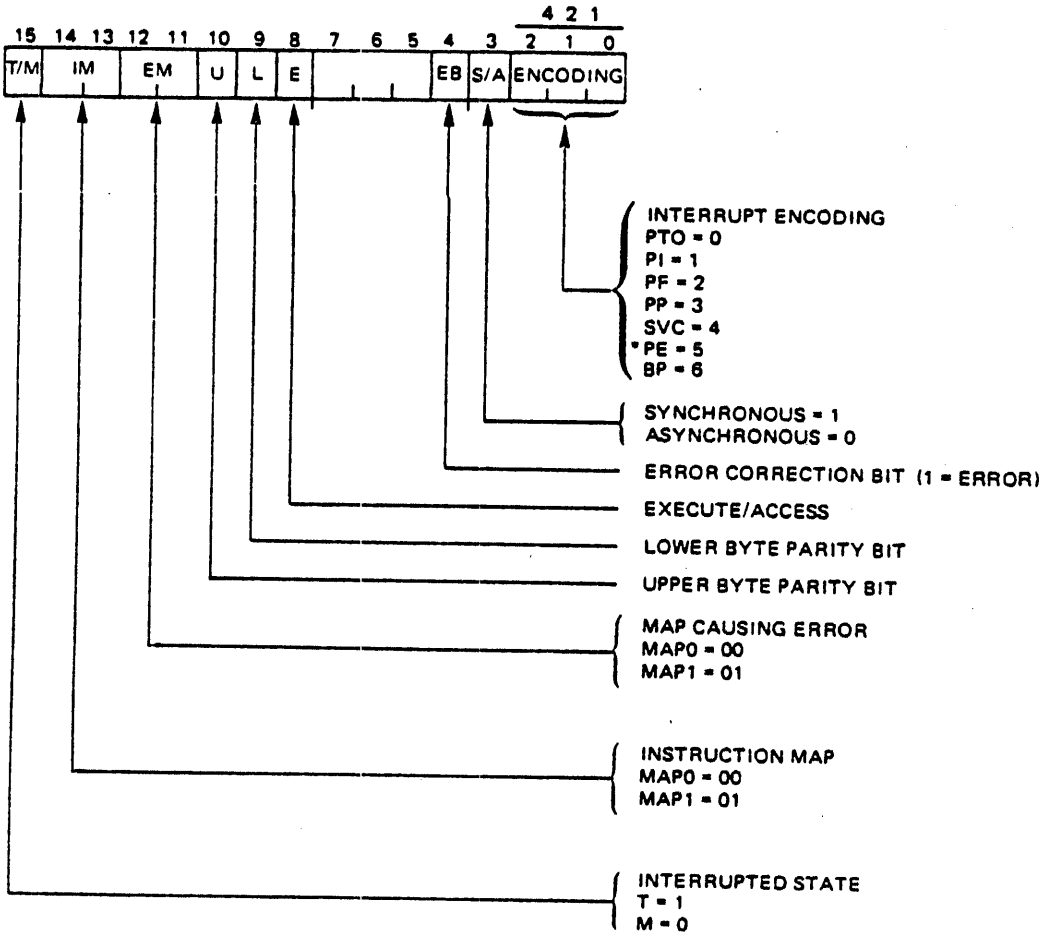
2.2.4.1 MMS Status Word

Error correction status may be determined by examining bit 4 and bits 0, 1 and 2 of MMS status word zero. Figure 2-4 illustrates the MMS status word zero format used with the GA-16/230/240.

Bits 0, 1 and 2 of MMS status word zero are binary weighted for error indication. Correctable errors are indicated by a decode of these bits being equal to zero, and bit 4 equal to a one. A non-correctable error is indicated when bits 0, 1 and 2 decode to five and bit 4 is equal to a one.

2.2.4.2 MPP Status Word

If MMS is not installed, error correction may be determined by examing the MPP status word. A correctable error is indicated by bit 12 of the MPP status word being on. Non-correctable errors are indicated by the error correction bit (bit 12) and the parity error bit (bit 0) being on. Figure 2-5 shows the MPP status word coding.



*NON CORRECTABLE ERRORS CAUSE ECC BOARD TO GENERATE ODD PARITY, HENCE PE IS INDICATED

Figure 2-4. MMS Interrupt Status, Word 0

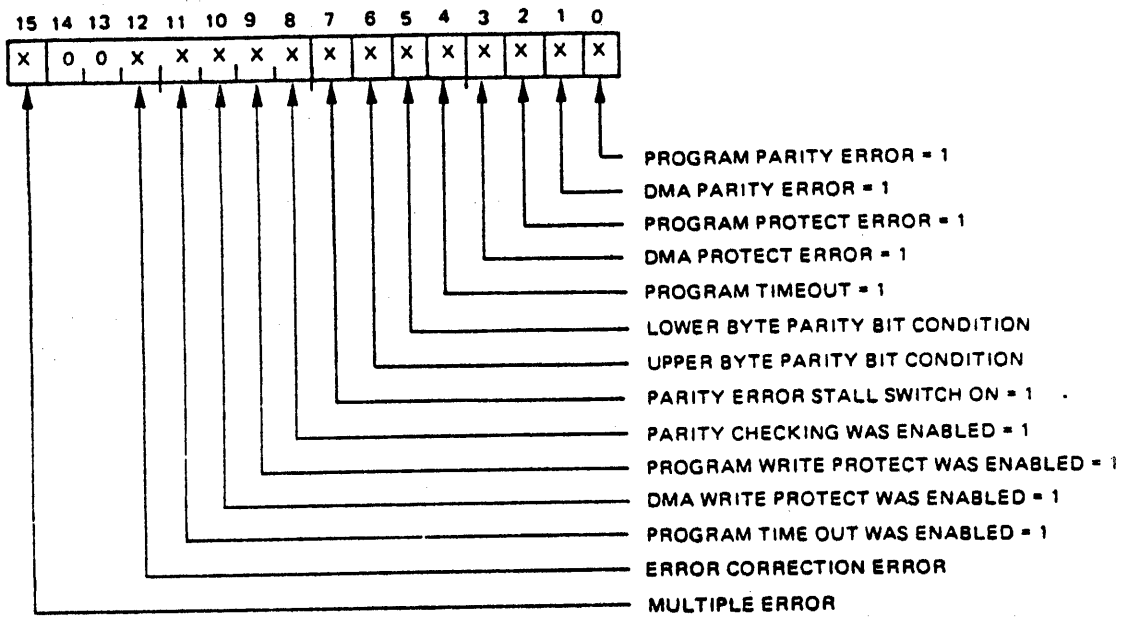


Figure 2-5. MPP Status Word

memory management system 3

3.1 INTRODUCTION

The Memory Management System (MMS) provides an operating environment external to the CPU which permits the implementation of a protected, multi-programmed, memory management operating system. The application of this option is directed toward multi-user and multi-process real-time systems. In providing an environment for multi-programming, MMS protects user programs from one another and protects the operating system from user programs. Many users can have access to the CPU, but the interaction between the various user programs can be rigidly controlled by the operating system. Additionally, the operating system or executive program can be protected from user programs.

The memory mapping scheme that is implemented in MMS is very flexible and efficient in reconfiguring logical to physical memory. Very little overhead time is required by the operating system to consolidate programs. The features of MMS that provide these capabilities are:

- Memory Expansion to 2048 Kilobytes (KB) of physical memory
- Mapping of physical memory (2048KB) into 128KB blocks of logical memory
- Two MAP system capable of separate DMA/DMT I/O mapping
- Memory write protection
- DMA/DMT write protection
- Mapping page fault protection
- Program timeout protection
- Instruction execution protection
- Indexed real memory addressing capability

The MMS is mechanized in a single circuit board, 31D03076A. The MMS board controls and indicators are identified in Figure 1-7 and defined in Table 1-2.

3.2 MMS INTERFACE

MMS interfaces the GA-16/240 with its memory subsystem (Figure 3-1). It monitors CPU activity on the I/O and memory buses to provide an environment for protected multi-programming of non-cooperating, contentious programs. By monitoring and controlling the memory bus, MMS provides a mapping mechanism to expand usable real memory from the GA-16/240's inherent architectural limit of 128KB to 2048KB.

Although only 128KB of real memory may be employed at any instant of time, MMS mapping commands may reconfigure whatever part of the real 2048KB of memory will be accessed by the CPU. MMS does this by transforming or translating the GA-16/240's logical addresses to the appropriate real addresses through its mapping functions. By routing memory control through MMS, program or DMA/DMT write protection is possible.

Finally, MMS monitors and controls some CPU activity. For example, if ISE is turned off too long (program timeout), the operating system may lose control of the machine. In this case, and for other error conditions, an NI interrupt will be generated to signal abnormal user program activity. This NI interrupt causes the CPU map to operate in transparent mode; that is, it enables the operating system to regain control.

The next few parts of this section will cover a general description of memory mapping which will lead to specific descriptions of MMS capabilities and commands. When references are made to DMA maps or options, it is to be understood that DMT is also implied.

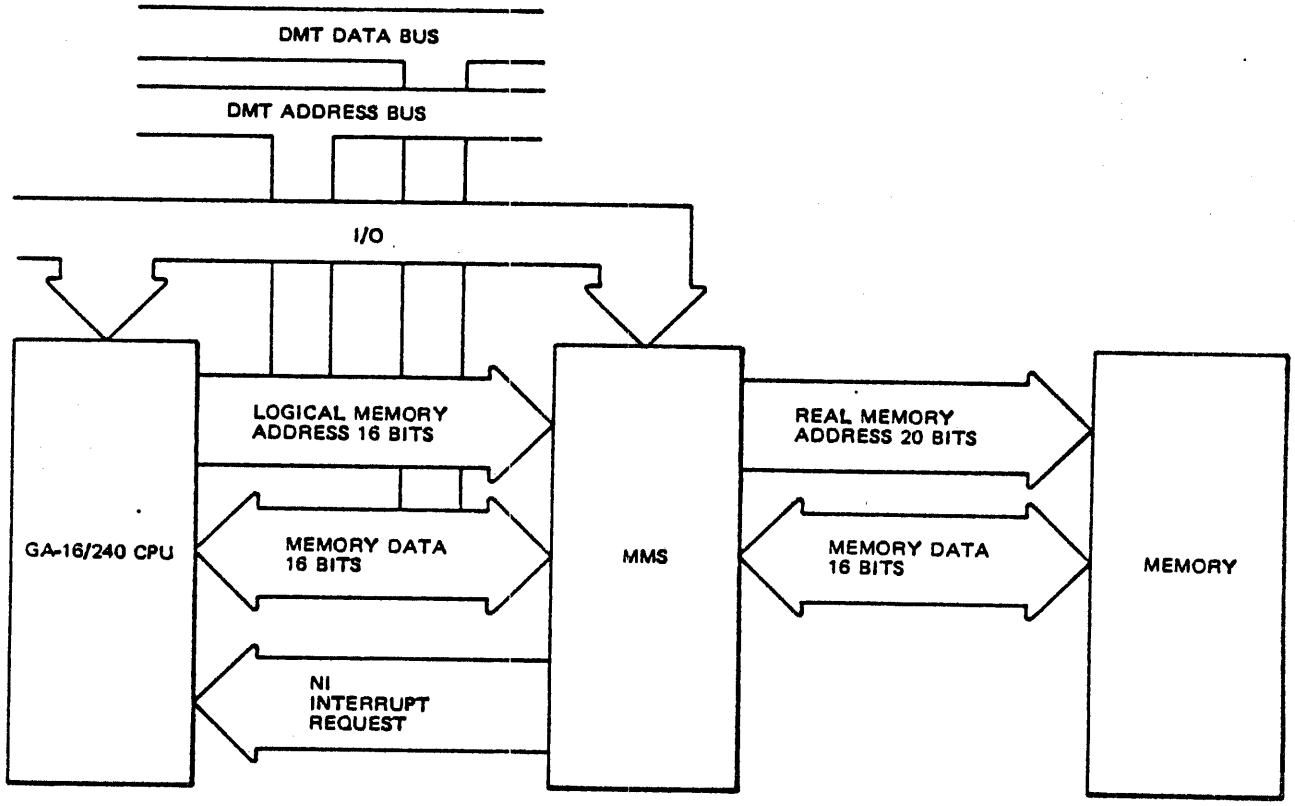


Figure 3-1. Memory Management System Architecture

3.3 MMS FUNDAMENTALS

The basic function performed by MMS is the translation, through its mapping feature, of logical addresses to real addresses. Memory mapping is accomplished by making a "logical-to-real" address translation. Address translation is necessary to address locations greater than 64K because the program counter (register P) is only 16 bits long (X'FFFF' = 64K). This P counter limit of memory addressing or address space is called the "logical" address of the machine. With MMS installed, the physical memory size may be increased to 1024KW or 2048KB. This physical memory will be referred to as "real" memory with "real" address. The logical and real address will correspond if MMS is installed but mapping is not enabled. This is called the "transparent" mode (Figure 3-2).

Real memory can contain 64K logical memory spaces. MMS contains two maps, each map can translate the logical memory space address to a real memory space address. The logical memory spaces will be shown as contiguous blocks of memory for ease of grasping the basic concepts. Actually, each mapped space is made up of sixty-four 1K pages, of which sixty-two may be "scattered" throughout memory as required.

It should be noted that page zero (0) will always address real memory, and page sixty-three (63) will always address the System Console Interface (SCI) if present. That is to say, page 0 will always address locations X'0000-X'03FF of real memory and page 63, address locations X'FC00-X'FFFF will always reference the SCI. If the SCI is not present real memory will be addressed.

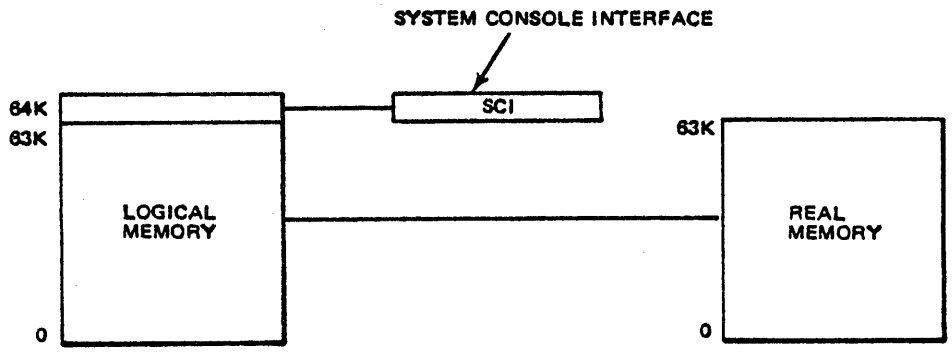


Figure 3-2. Transparent Mode

In Figure 3-3, real memories 5 through 16 could be thought of as containing different logical memories. However, only one logical-to-real translation can be mapped.

Real memories 2 to 16 could be accessed by changing map 1 data. In other words, if programs that resided in real memory 2 had been executed to completion, map 1 data could be changed to effect a different address translation to access a different part of real memory.

The above representation implies that only one user can be mapped at any one time, that each user could access 64K, and that each user would operate as if he were the only user. In reality, it would be highly unlikely for each user program to require exactly 64K of memory space; most likely, a software operating system would be incorporated to effect executive control over the various programs. Also, there might be some routines that different user programs need to share. Lastly, MMS must incorporate some protective features to stop user programs from destroying each other.

A supervisory operating system (OS) can reside in memory space. However, the part of the OS which controls MMS functions would typically be placed in lower memory, since any error condition sensed by MMS invokes an NI interrupt and the transparent mode (logical = real address). This allows easy access to the operating system's control. Some of the controls performed by the operating system are:

- Allocating blocks of memory for users
- Loading user programs in memory
- Monitoring users' action
- Setting priorities for execution of programs
- Error recovery

A simplified organization of memory running under control of an operating system might appear as represented in Figure 3-4.

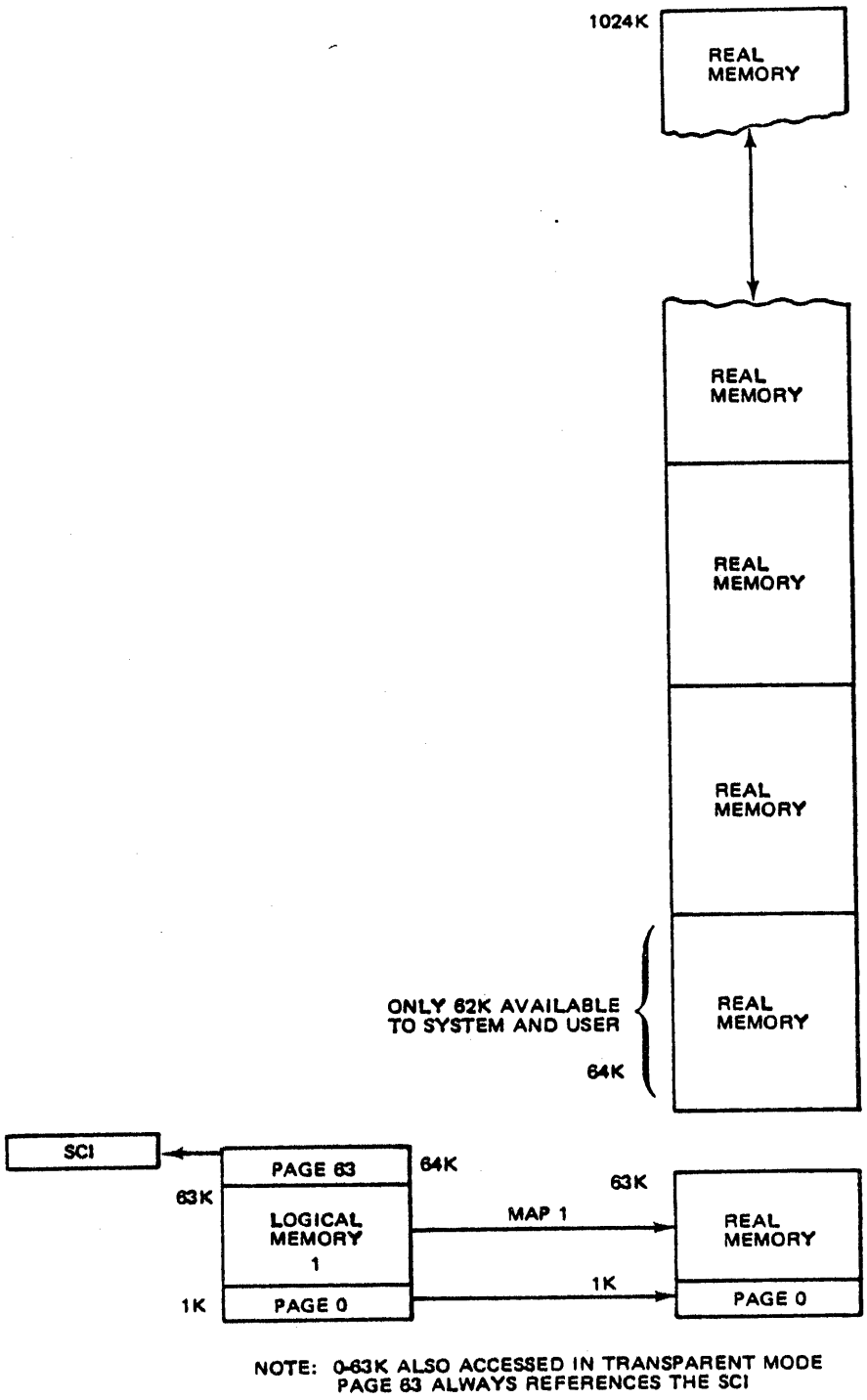


Figure 3-3. Maximum Memory Size

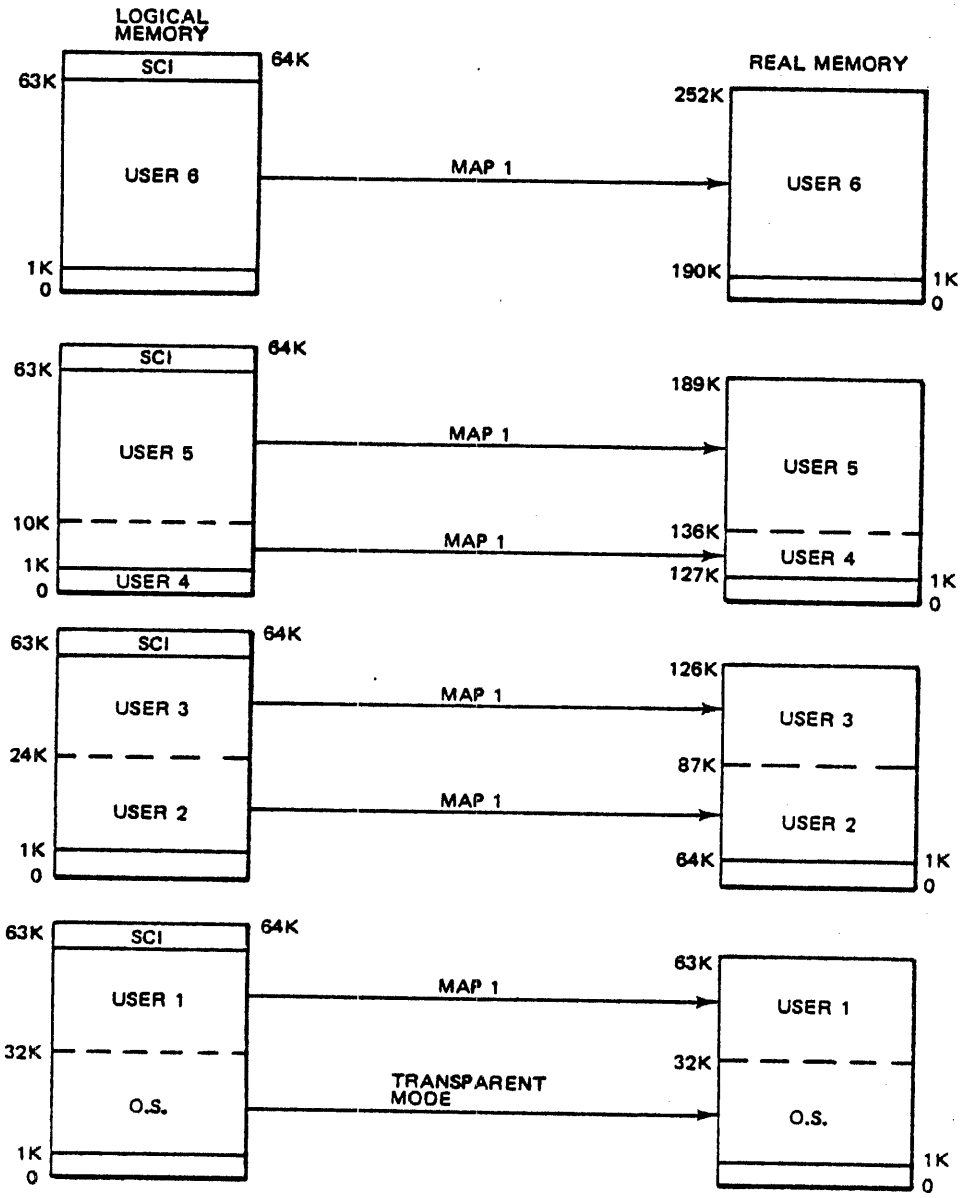


Figure 3-4. Simplified Memory and Operating System

In Figure 3-4, not every program occupies 64K. In fact, MMS can allocate memory in blocks as small as 1K. Suppose that the operating system ordered user programs to execute in numerical sequence. As soon as user 1 had been serviced for its allocated time period, the operating system could redefine MAP 1 for user 5, then transfer control to user 5. When user 5 was finished, MAP 1 could be redefined to allocate memory for user 6, and so on down the line.

A typical way for control to be transferred from a user program to the operating system would incorporate the real-time clock (RTC). The operating system could determine the time intervals allocated for each user. The RTC interrupt routine can be designed to force the transparent state and, thereby, transfer control back to the operating system. Thus, the user would execute its program until the RTC interrupted and transferred control back to the operating system. The operating system would count the intervals and either return to the original user or select the next user for its time period.

If an error occurs during user execution, an NI interrupt will be generated. It forces the transparent state and returns to the operating system.

The example shown in Figure 3-4 is not completely representative of MMS capability. First, it only shows a maximum of 252K of memory, and more important, it does not show the interleaving possible between logical and real memory. The use of MAP 0 is not shown and it falsely implies that mapping is done only in contiguous blocks. If that were the case, a lot of operating system overhead time would be taken up by moving various user routines into contiguous spaces. Also some users may share some routines. For a more representative picture of mapping, see Figure 3-5.

In Figure 3-5 both user 3 and 4 share, through their respective maps, a part of real memory from 120K to 128K for an I/O call routine. Note that neither user 3 nor user 4 has completely contiguous blocks in real memory. However, as far as the CPU is concerned, it sees only logical memory; therefore, program execution is maintained in an orderly fashion so long as the operating system does its map bookkeeping orderly.

MAP 1 could be considered as the 'system MAP' in that portions of the operating system and all DMA/DMT operations reside in real memory space controlled by MAP 1. Map 0 should be considered as the 'user's MAP' on a multiple map operating system. The user programs will reside in the portion of real memory controlled by MAP 0.

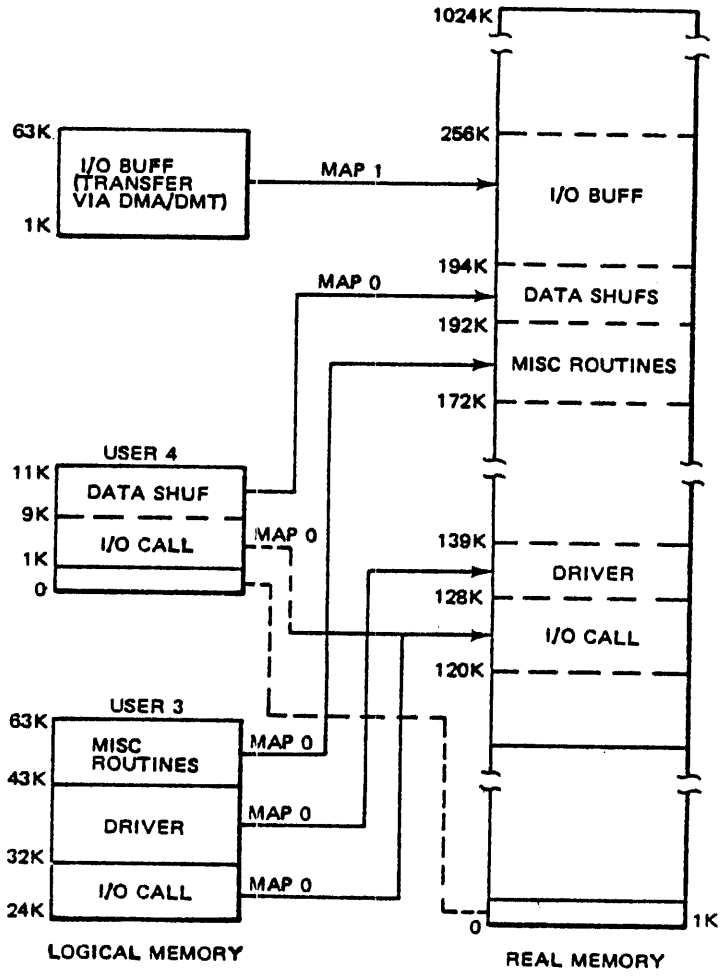


Figure 3-5. Simplified Map Function

In Figure 3-5, both user 3 and 4 share a routine in real memory labeled I/O call. If the first user on the line started some DMA activity with I/O CALL, the other user would have to wait until the first user was finished before he could use I/O CALL for his DMA requirements. If the associated DMA operation entailed a large block transfer, the wait time would be prohibitive. To circumvent this, MMS allows DMA to use MAP 1 and the user to user MAP 0. DMA operations are always mapped through MAP1 when DMA mapping is enabled. That is, DMA activity from one user may be overlapped with program execution of another.

If the map interleaving of Figure 3-5 is combined with the transparent mode of Figure 3-4, and again combined to show 1024K of memory as in Figure 3-3 for a dozen or more users, we would finally show a representative total picture of MMS capability. Unfortunately, the various mapping lines would be so numerous as to obliterate all meaning. However, the mapping concept involved should be clear.

MMS employs two maps and a transparent (T) mode. If DMA mapping is enabled, it always employs MAP 1. If DMA mapping is not enabled, it will use transparent mode even if one of the other MAPs is selected for use.

In MMS, the term 'page' refers to a 1KW block of memory, the smallest memory space that may be mapped. If for some reason a program or DMA references an unmapped page of real memory, a page fault (PF) error occurs. A page fault will also occur if the user tries to address a memory location in excess of 1024KW using the indexed real memory addressing feature.

The memory write protect feature of MMS provides the capability of protecting pages of memory from undesired access by write operations. As its name implies, write protection prohibits any changes in selected memory pages by either DMA or program activity. Instruction execute protection inhibits the CPU from executing instructions in selected pages. Write protection or instruction execution protection violation can give a program protect error (PP).

Program timeout signals the fact that some user has kept ISE off too long, or that a long consecutive string of non-interruptable instructions has been executed.

3.4 MEMORY MAPPING

MMS may operate in either the transparent mode (in which no address translation occurs) or in a mapped mode in which MAP 1 or MAP 0 is used for address translations. Commands to MMS (see Section 3.5) specify transparent or mapped mode. DMA mapping occurs through MAP 1 when enabled. MMS divides the 64K locations of logical address space into sixty-four 1024 (1K) word pages, 62 of which may be located anywhere in memory. The logical address space represented by the MMS map may intersect real memory in a random fashion. In addition, the logical space may include pages of the entire real space including those in the first 64K which are also accessible by the transparent or unmapped mode.

The mapping mechanism provided by MMS causes translation of the logical address referenced by the executing program to a real address in the real 1024K, 16-bit word memory. Logical addresses may range from 0 to 64K. Figure 3-6 describes the mapping mechanism used to convert a 16-bit (0 to 64K) logical address into a 20-bit (0 to 1024K) real address. Address translation adds 100ns to each memory reference. The MMS maps are scratch pad memories contained on the MMS option board. For loading and unloading of the MMS maps, see Sections 3.5.1.1 and 3.5.1.2.

MMS in mapped mode causes the CPU generated, 16-bit logical address to be divided into two parts. The most significant 6 bits are a logical page number (LPN) which addresses one of 64 map cells of one of the MMS maps. The least significant part of the map cell is the real page number (RPN) which addresses any one of the 1024K pages of real memory. The least significant part of the logical address (PDISP) is a page displacement which provides a specific address on the page. That is, its 10 bits point to a specific location on the 1K page selected by the RPN. This address translation is called "mapping". The selected real page is said to be "mapped".

In transparent mode, the memory acquires its address directly from the CPU.

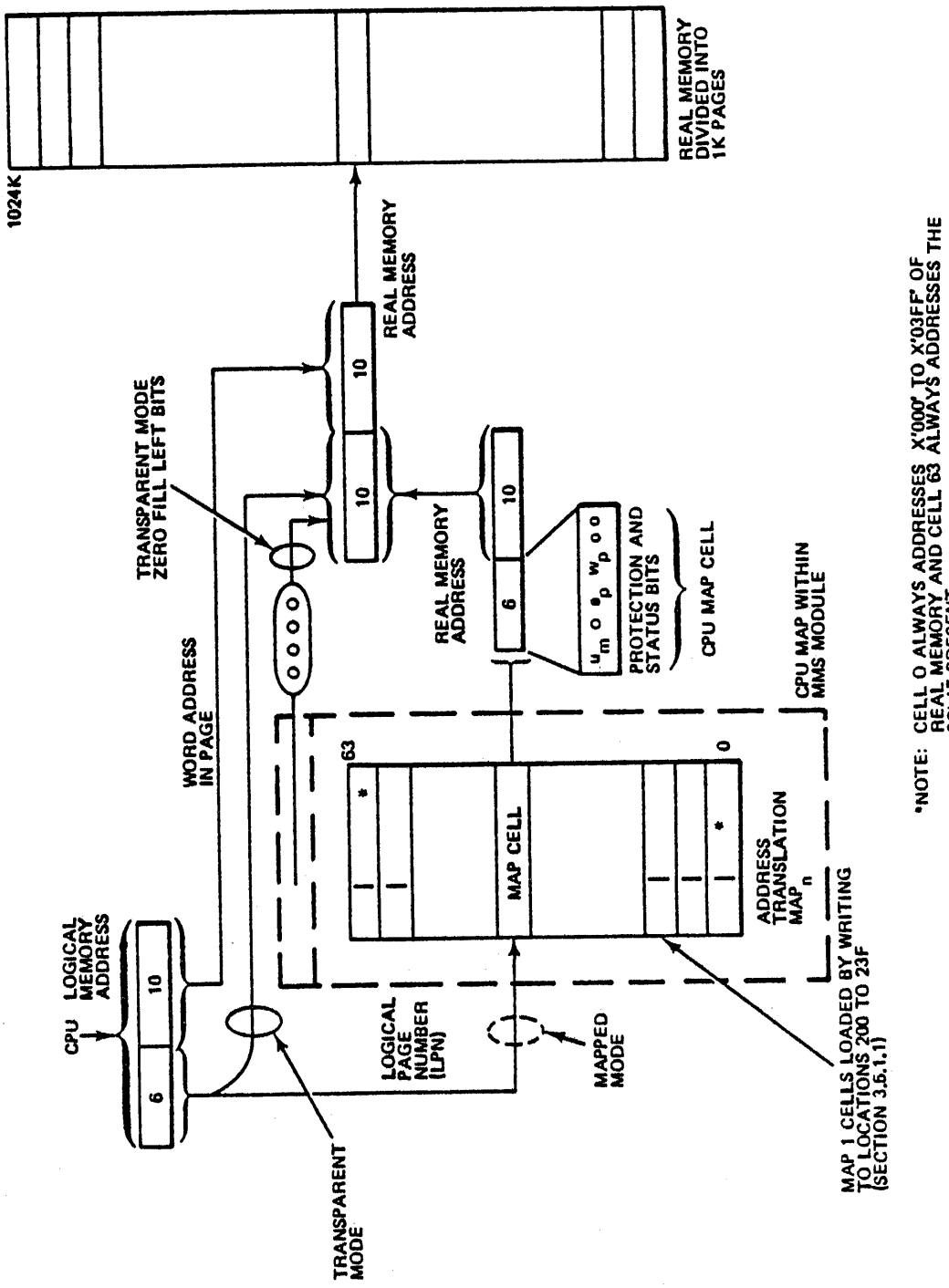


Figure 3-6. MMS Dynamic Address Translation (Mapping)

3.4.1 MAP CELL FORMAT

The most significant parts of the map cell are status and control bits relating to the mapped page (Figure 3-7) and the real page number.

Bit 15 of a map cell word is the unmapped bit. An address translation through a page with this bit on will result in a page fault.

Bit 14 contains no status. This bit position will always be zero.

Bit 13 is the execution protection bit. If this bit is on, and memory protect is enabled (by previous output of command value as described in Section 3.5.2), no locations in the referenced page may be executed as instructions. This bit is not used by the MAP 1 cells used for DMA mapping.

Bit 12 is the write protection bit. When it is on, and memory protect is enabled, no locations in the referenced page may be written into; that is, the page is read-only.

Bit 11 contains no status. This bit position should always be zero.

Bit 10 contains no status. This bit position should always be zero.

Upon power turn-on, the map cell data is indeterminate. One of the first things an operating system using MMS would do is to load maps for subsequent usage. All unused map cells must explicitly be set to 8000 (the unmapped condition) when loading the map.

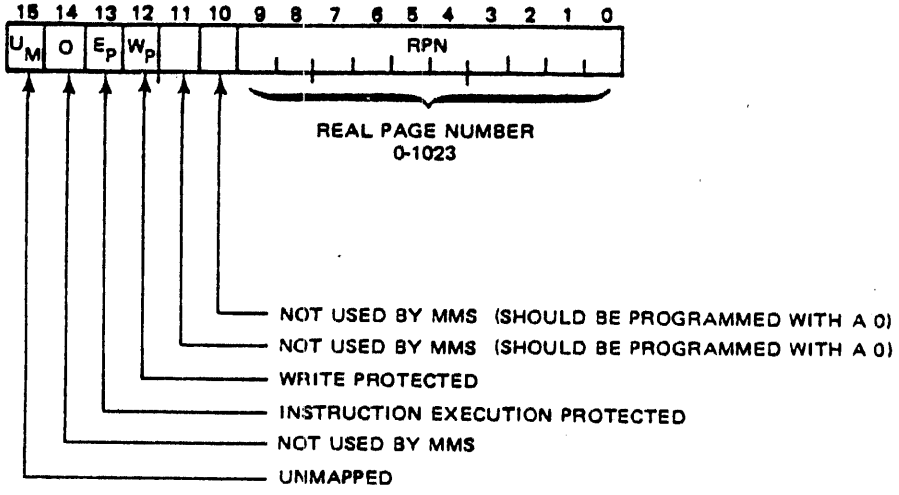


Figure 3-7. Map Cell Format

3.4.2 MAP DATA WRITING

MAP 1 in MMS consists of 64 16-bit entries containing an RPN and miscellaneous status and control bits for the LPN corresponding to that entry. MAP 0 data is loaded by writing the appropriate 16-bit word into logical memory locations X'200' to X'23F'. This data will be written into the MMS scratch pad and also into main memory (Figure 3-9).

Similarly the MAP 1 is loaded by writing the appropriate 16-bit word into logical memory locations X'240' to X'27F' (Figure 3-8).

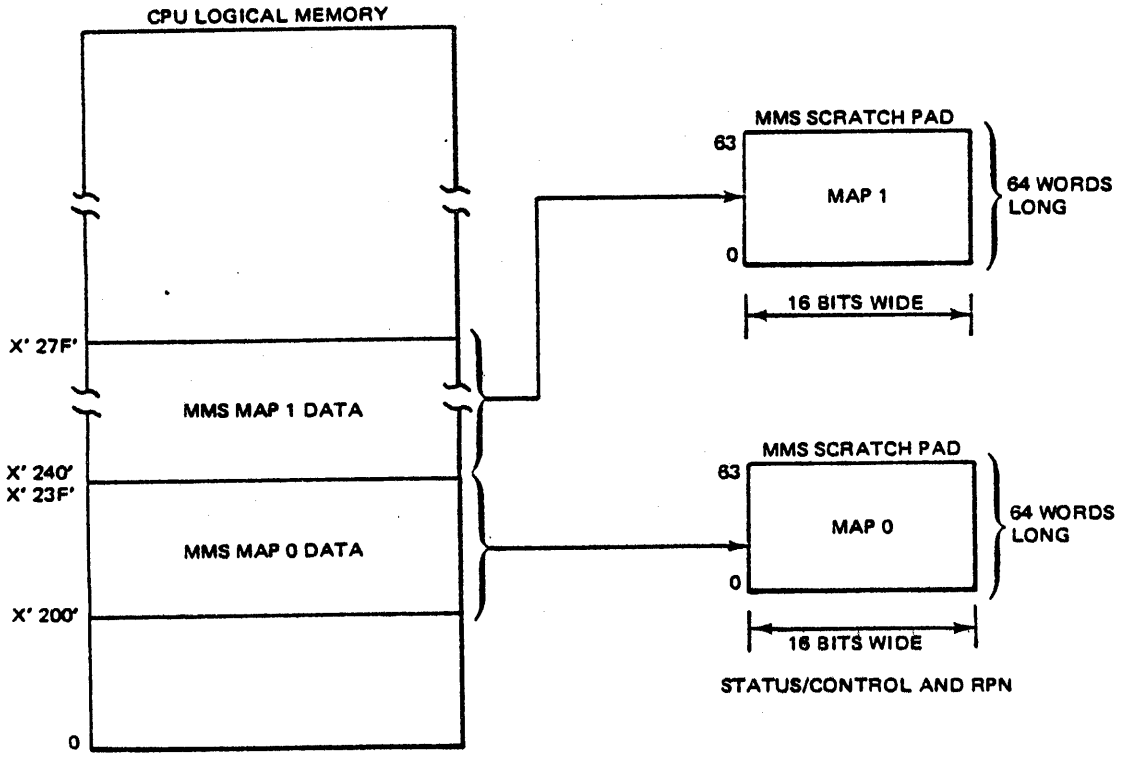
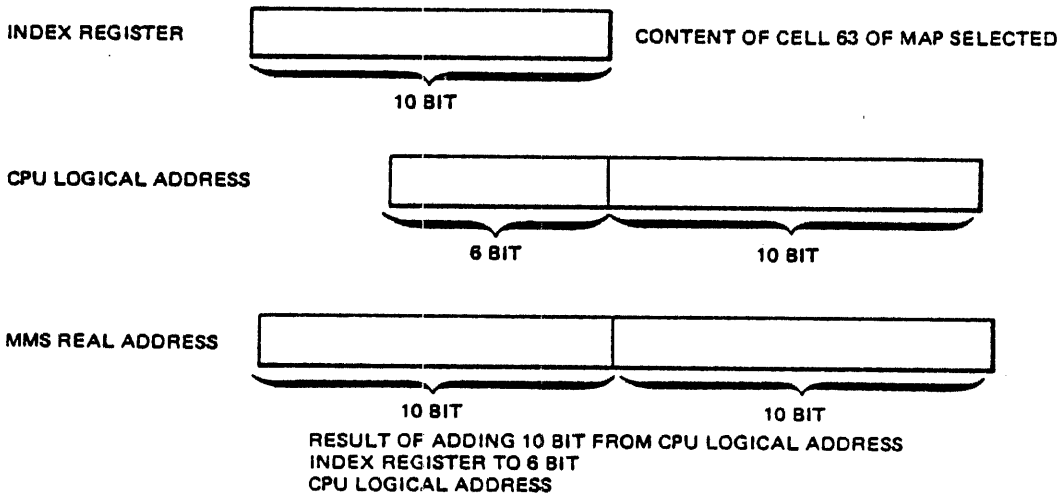


Figure 3-8. Map Loading

3.4.3 MMS MAP PAGES 0 AND 63

Page 0 of logical memory will always reference page 0 of real memory. The MMS board has been designed so that when a page 0 memory address reference is detected, it forces the 10 most significant memory address bits produced by the MMS to zero. The contents of the locations 200 and 240, which represent the address translation of page 0 of the user and system maps, should be programmed with the correct information, although this information will not be used by the MMS.

Page 63 of logical memory also will always be page 63 of real memory. The MMS board has been designed to force this condition. Normally cell 63 of MMS should be programmed with the correct information to perform the required translation, although the information is not used by the MMS. For indexed, real memory addressing cell 63 of the currently selected MAP can be loaded with a value that will be used as the index register value. The 16-bit address provided by the CPU will be added to this index, thus enabling a memory fetch from any real memory address (Figure 3-9).



NOTE: ALL INSTRUCTION SEQUENCES TO MMS SHOULD EXECUTE WITH I/O INTERRUPT DISABLED. (ISE FLIP FLOP SET TO ZERO).

Figure 3-9. Indexed Real Memory Address Computation

3.5 MMS INSTRUCTIONS

MMS is activated by sequences of programmed I/O commands to device code X'39' as shown in Table 3-1.

NOTE

All instruction sequences to MMS should execute with I/O interrupts disabled. (ISE flip-flop set to zero.)

The MMS will not prevent DMA devices from doing data transfers while map data is being loaded.

Control 2 and 5 are ignored. Control 0 is detected, but not used.

3.5.1 MAP LOADING

Performing a store into memory locations X'200' to X'27F' will load data into the MAP 0 and MAP 1. Logical pages 0 and 63 will always be transparent for correct operation of the software.

3.5.1.1 Map Reading

There are two modes for reading the map. The two modes differ in that the source of the map data may come from the MMS board or from main memory.

Map data is obtained by performing a read from memory locations X'200' to X'27F'. Data will actually be obtained from memory unless a load command value type has been received, the MMS data is obtained from the MMS board and a memory cycle abort is performed. To read memory locations X'200' to X'284', a system reset or a CTRL 7,X'39' must be performed.

Table 3-1. Instruction Summary for MMS

Device Select Code: X'39'
Interrupt Vector: X'42' (non-inhibitable)
Data Channel Locations: NA
P-Storage: X'7A'; ISE X'7B'

Instruction		Description
1139	CTRL 1,X'39'	Set command value. Prepare for input of command word to specify status of MMS operations and interrupts (see DTOR).
1339	CTRL 3,X'39'	Single cycle. Use MAP 0 to translate logical to real address for all data references between the next two instruction fetches.
1439	CTRL 4,X'39'	Single cycle. Use MAP 1 to translate logical to real address for all data references between the next two instruction fetches.
1639	CTRL 6,X'39'	Indexed, real memory addressing. Use contents of CPU map cell 63 as an index register, the value of which is added to the CPU address reference to obtain a 20-bit address. This addressing mode will be used for all data references between the next two instructions.
1739	CTRL 7,X'39'	Clear. Reset all command value bits to zero and return to transparent mode.
1X79	DTOR R,X'39'	(Following CTRL 1.) Output command value which defines CPU, DMA and MMS options. An RTNIV instruction must follow to cause options to take effect.
1X79	DTOM R,X'39'	(Following CTRL 1.) Same usages as described under DTOR except output command is in memory location referenced by register R.
1X79	DTOR R,X'39'	At any time but not preceded by CTRL 0, or 1, outputs command which defines the ECC mode of operation of the memory.

3.5.1.2 Status

There are five words to MMS NI interrupt status as shown in Figure 3-10. Word 0 specifies the interrupt status. Word 1 specifies the instruction fetch address (IFETCH) of the instruction causing a synchronous NI interrupt (Section 3.6.5). This word may be ignored for asynchronous interrupts. Word 2 specifies the data address which issued the interrupt. Word 3 specifies the preceding IFETCH address to provide diagnostic information for troubleshooting. Word 4 shows the operating state of the MMS, as determined by the command value (Section 3.5.2) prior to the interrupt, to facilitate restoration of MMS after interrupt has been processed.

A status read is recommended when MMS generates an NI interrupt, and is carried out by the interrupt routine referenced by vector X'42'. An NI interrupt should be serviced before a second one is allowed to occur. If, due to a double error condition, a second NI interrupt occurs, the CPU is forced to the MSTAL condition. In MSTAL, the memory is not always ready and the CPU cannot execute instructions. An MSTAL requires a system reset (with resulting abort of all programs). Resetting the NI bit with the "Set Command Value" sequence (described in Section 3.5.2) cancels the potential MSTAL. Refer to Section 3.6 for more detailed information on interrupts.

A detailed description of the four MMS status words follows:

1. Status Word 0: The format of the MMS NI interrupt status word (word 0) is shown in Figure 3-11. Bits 15 through 11 specify the interrupted status of MMS. If bit 15 indicates transparent (T) mode, the IFETCH address is real. Bit 13 specifies the instruction map in use at the time of the interrupt and bit 11 specifies the map which caused the error. The instruction map and the map causing the error may differ if a single cycle instruction causes an NI interrupt. Note that bit 15 may indicate T state if MMS is executing unmapped but protected.

   Bit 3 specifies whether the NI was synchronous (=1) or asynchronous (=0). A synchronous interrupt is program generated; that is, page fault or memory parity error. An asynchronous error is asynchronous to the program; that is, program timeout or DMA induced error.

   Bit 2 to 0 specify the interrupt encoding. Refer to Section 3.6 for further definition of each interrupt. PTO is an invalid synchronous interrupt. All bits of status word 0 are reset by a load command value sequence. For further definition of interrupt status word error interpretation when PF or PP is indicated by bits 2 to 0, refer to Tables 1-2 and 3-2.

2. Status Word 1: Specifies the address of the instruction fetched when NI interrupt occurred. For synchronous interrupt processing, this will be the recovery address for the interrupted program; asynchronous interrupts are recovered via location X'7A' and X'7B'.

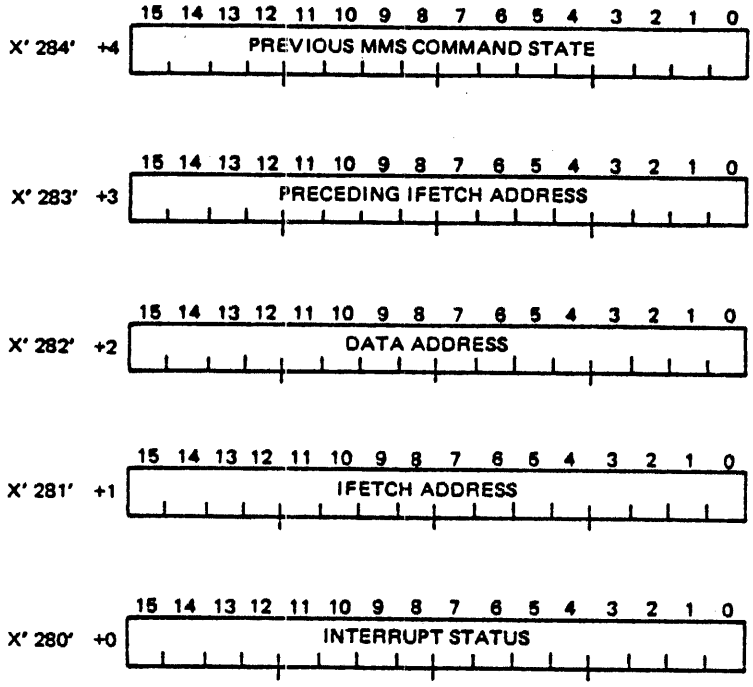


Figure 3-10. MMS Status Word

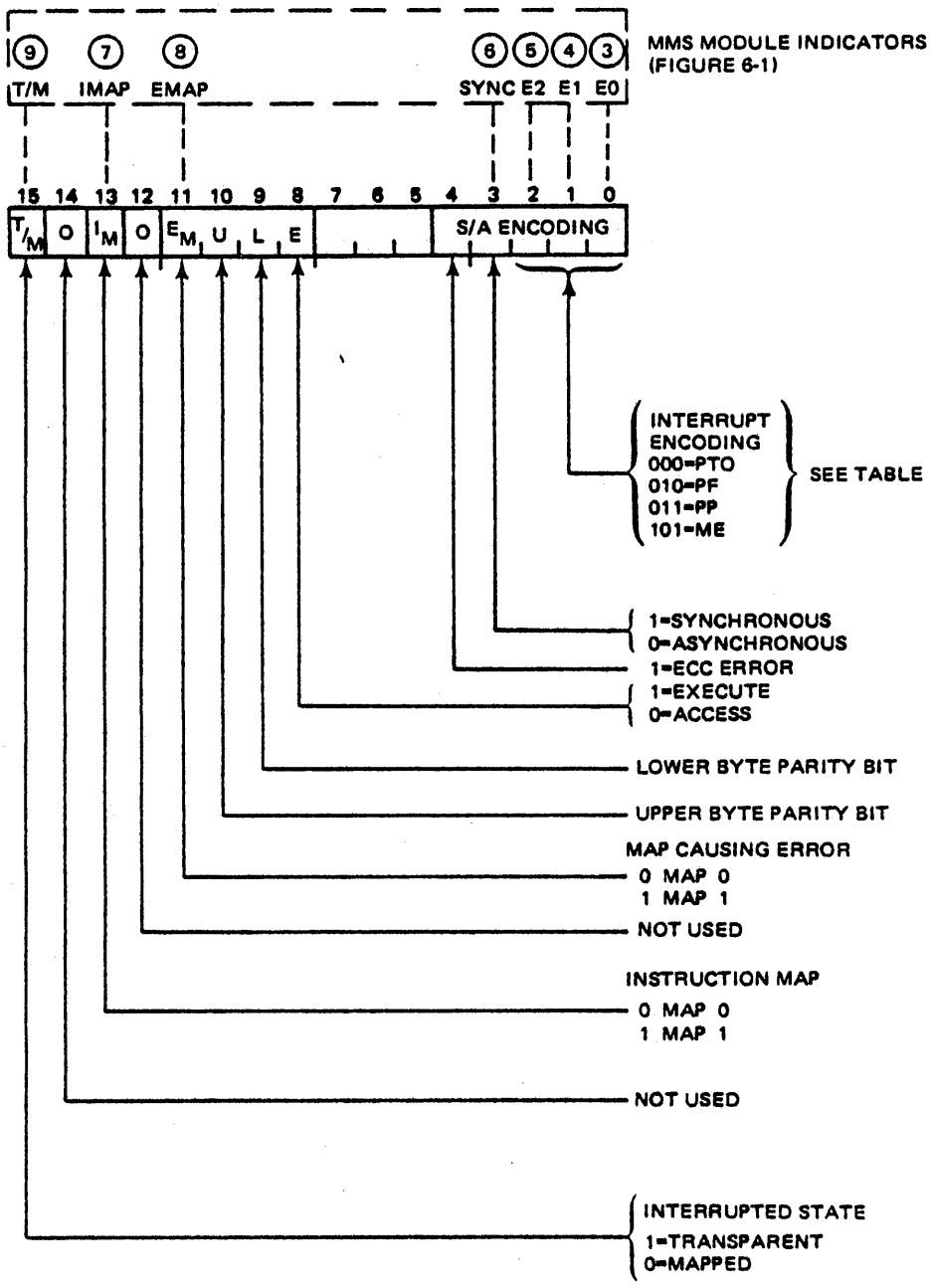


Figure 3-11. MMS Interrupt Status, Word 0

Table 3-2. Further Interrupt Status Word 0 Interpretation When Page Fault, Program Protect, or Memory Error Occurs.

INTERRUPT ENCODING Bits 2,1,0	S/A Bit 3	E Bit 8	CAUSE OF INTERRUPT
010 = PF (Page Fault)	0	NA	DMA PF
	1	0	Data Access
	1	1	Instruction Access
011 = PP (Program Protect)	0	NA	DMA PP
	1	0	Write Protect
	1	1	Execute Protect
101 - ME (Memory Error)	0	NA	DMA Memory Error
	1	0	Data Access
	1	1	Instruction Access

NOTE

Write protection interrupt occurs when the write is attempted into a protected address. Referring to an indirect address or stack pointer that is write protected will not cause an error; the error will occur when an attempt is made to write into a protected address referenced by the indirect address (or pointer).

Data access page fault interrupts will occur any time an address is referenced on a page that is unmapped; therefore, reference to a pointer that is on an unmapped page will cause an interrupt. Similarly, a jump to an unmapped page will cause an instruction address fault.

3. Status Word 2: Specifies the address of the data being operated on when the NI interrupt occurred.

4. Status Word 3: Specifies the instruction address which preceded the instruction when NI interrupt occurred.

5. Status Word 4: Provides the capability to read the MMS command state prior to the NI interrupt. The command state is set by outputting a command word following a CTRL 1 instruction as described in Section 3.5.2. The bits in status word 4 are the same as those in the command word, except bits 10 through 8 which are zero. Status word 4 may be used by the interrupt routine as the source of a new command word to restore conditions prior to the interrupt. Figure 3-12 shows the format of status word 4.

3.5.1.3 Status Reading

Status words 0, 1, 2, 3 and 4 are obtained by performing an instruction that reads memory locations X'280', X'281' X'282' X'283 and X'284'. These memory locations will not actually be read as the read cycle will be aborted, data from the MMS board will be returned to the CPU as if the memory read had been completed. A control 7 or system reset will enable reading from CPU memory.

In this way, it will appear to the program as if the MMS status words were located in memory locations X'280' to X'284'.

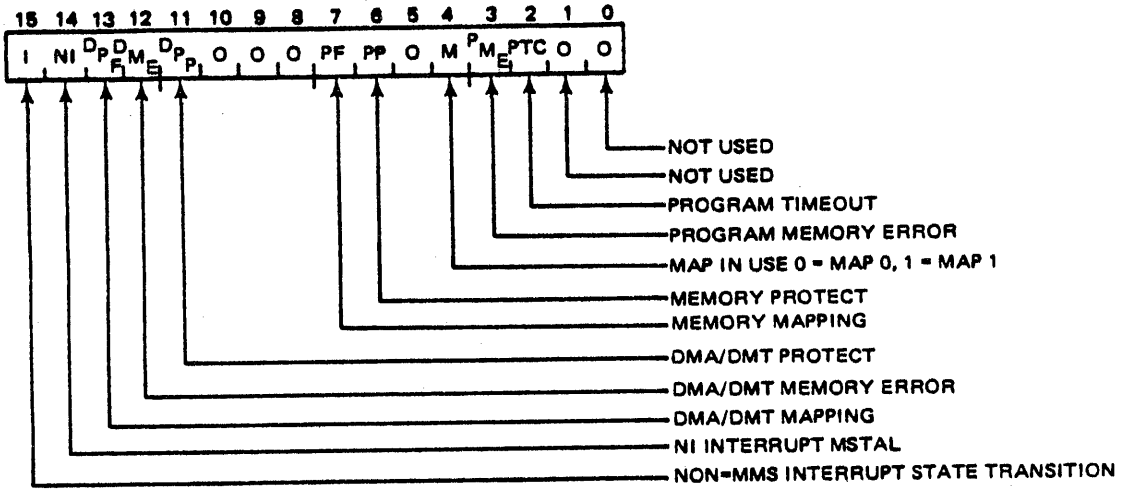


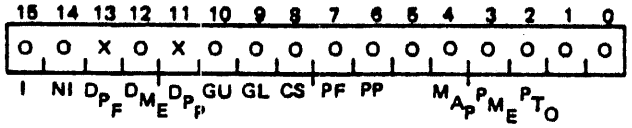
Figure 3-12. Status Word 4

3.5.2 SET COMMAND VALUE

The command value transferred to MMS (after a CTRL 1,X'39' instruction) specifies the status of all MMS options and interrupts. The effect of the command does not occur until the RTNIV instruction has been executed to permit map transition to take place with correct entrance address specification.

The format of the command value is shown in Figure 3-13. Table 3-3 describes the command value fields.

The initial state of the MMS upon power turn-on is the transparent state (T) state. This state is equivalent to the following command value:



This state is also entered upon any NI interrupt from MMS. To set up (or restore) a CPU mapped or protected state of MMS, a command value must be output with bit 8 (CS) = 1 plus the other bits required to set the desired options.

When entering a new mapped state (CS, bit 8 = 1) P-counter transition must occur in a manner which permits clean exit from the old mapped condition and proper entrance to the new. Both the IFETCH value stored in MMS and the P-counter in the CPU must agree. This is accomplished via the instruction "return via indirect vector" (RTNIV). This is a two-word instruction in which the second word points to two locations whose contents define both the new P-counter value and ISE. A typical sequence would appear as:

```
        INH                         Disable interrupts
        CTRL    1,X'39'             Specify 'set command value' group
        DTOM    SAM,X'39'           Transfer command value
        LARS    *STACK              Restore registers for entrance to
                                      new state
        RTNIV   RAY                 Specify new P-counter value and ISE
RAY     DS      1                   State entry address
RAY+1   DS      1                   New ISE
SAM     DC      command-value       Specify options required
```

The address stored in RAY will be used as the entry point in the new mapped state specified in location SAM. Entrance will be made with all general registers valid (per LARS), the state and mode per SAM, and ISE per RAY+1. Note that the effect of the DTOM is delayed for two instructions (IFETCH sequences) to allow the LARS and RTNIV to take place.

After the mapped state has been selected, further control of options will be governed by instructions to MMS and by interrupts (as described in the following sections) which describe each function enabled by the bits in the command value.

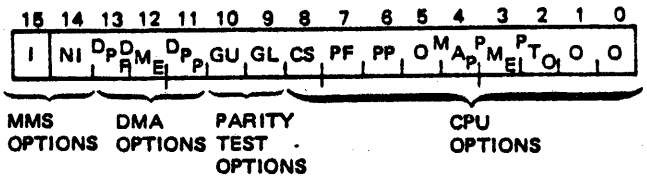


Figure 3-13. MMS Command Value

Table 3-3. Command Value Description (Sheet 1 of 2)

Bit	Mnemonic	Description	State	Significance
MMS Options				
15	I	Non-MMS interrupt state transition	1 0	Non-MMS int.→ M (MAP 1) Non-MMS int.→ T
14	NI	NI interrupt	1 0	Enabled NI will cause interrupt via X'42' Disabled NI will cause interrupt halt (MSTAL)
DMA Options				
13	DPF	Enable DMA I/O mapping and Page Fault Detect	1 0	Enabled, (MAP 1 is always used) Disabled, use real memory
12	DME	Enable DMA memory error detection	1 0	Enabled Disabled
11	DPP	Enable DMA memory protect	1 0	Enabled, (MAP 1 is always used) Disabled

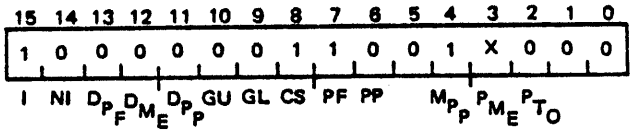
Table 3-3. Command Value Description (Sheet 2 of 2)

Bit	Mnemonic	Description	State	Significance
MMS Options				
Parity Test Options				
10	GU	Generate bad parity upper byte	1	Generate bad parity (for diagnostics)
			0	Do not generate bad parity
9	GL	Generate bad parity lower byte	1	Generate bad parity
			0	Do not generate bad parity
CPU Options				
8	CS	Change state or map function	1	Bits 7, 6 and 4 are valid
			0	Bits 7, 6 and 4 are ignored
7	PF	Enable CPU mapping and Page Fault Detect (unmapped and out of range)	1	M, mapped
			0	T, transparent
6	PP	Enable memory protect (write protect and execute protect)	1	Enabled
			0	Disabled
5				Not used
4	MAP	Map selection used for mapping and for protection	0	MAP 0
			1	MAP 1
3	PME	Enable program memory error detection	1	Enabled
			0	Disabled
2	PTO	Enable program timeout	1	Enabled
			0	Disabled
1				Not used
0				Not used

3.5.2.1 T State/M1 State (I)

The transparent (T) state is entered upon power up, as a result of a MMS NI interrupt or provided bit 15 ('I' bit) of command value = 0, and IN interrupt.

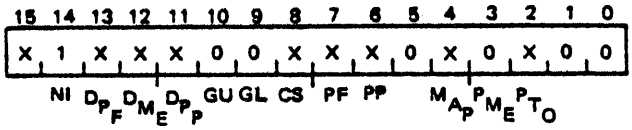
If bit 15 = 1, and the CPU mapping and page fault is enabled (PF = 1), non-MMS interrupts will cause transition to the M state. This transition is equivalent to the following command value:



Refer to Sections 3.6.1 and 3.6.2 for further description.

3.5.2.2 MSTAL (NI)

If the NI bit is set (bit 14 = 1) an NI interrupt will cause a jump via the management interrupt vector (X'42'). Any time an NI interrupt occurs, the NI bit is reset (bit 14 = 0). When NI bit = 0, the system is now prepared to do a memory management stall (MSTAL) when another NI interrupt occurs. The interrupt service routine will normally output a command value which resets the NI bit (bit 14 = 1) so that the stall cannot occur. To reactivate interrupt capability (thereby preventing MSTAL on the next NI interrupt) the following command value is required.

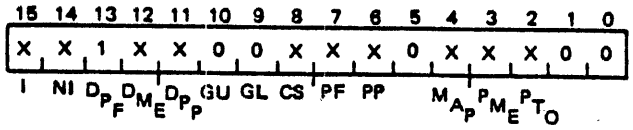


(Bits marked X are as determined by pre-NI interrupt conditions.)

Refer to Section 3.6.3 for further description of MSTAL.

3.5.2.3 Enable DMA Mapping and Page Fault (DPF)

The DMA mapping option is enabled by setting the DPF bit (13). DMA I/O mapping and page fault detection is enabled if the following command value is output.



(The bits marked X are as required for concurrently setting other options.)

DMA mapping is always via the MAP 1. The MAP 1 will therefore, be loaded with map cell data (enabling WP bit and resetting UM) for DMA address transition prior to issuing this command.

3.5.2.4 DMA Write Protection (DPP)

Pages of memory can be DMA write protected by setting the write protect bit of the associated cell in MAP 1 and by setting the DPP bit (1). In mapped mode DMA protection will be enabled for all 1K pages that are mapped (UM bit in map cell = 0) and have write protect (WP) bit = 1. (The EP bit in the map cell has no effect on DMA protection.)

In transparent mode, DMA protection is enabled for each 1K page (of real memory from 0 to 63K) whose corresponding MAP 1 cell has WP bit set.

DMA protection is removed by an MMS clear command (CTRL 7,X'39'); by outputting a new command value with DPP bit (11) = 0, or by reloading MAP 1 cells with the WP bit = 0.

3.5.2.5 DMA Memory Error Detection (DME)

Memory error detection upon DMA read operations is enabled by setting the DME bit (11). Memory error detection is in effect in either transparent or mapped mode. It is disabled by an MMS clear instruction (Section 3.5.5), system reset, or by output of a command value which resets the DPE bit.

3.5.2.6 Parity Error Simulation (GU, GL)

A parity error condition can be enabled for both CPU and DMA memory read operations by setting the GU bit (10) for upper byte and/or the GL bit (9) for lower byte. After issuing a command value, to set these bits a write operation must be performed into any memory location and bad parity (even) will be written to the location. A subsequent attempt to read this location (either in transparent or mapped mode) will result in an NI interrupt caused by MMS detecting the induced parity error. The interrupt routine may then verify the error by first inputting status words (Section 3.5.1), then fetching data via DATA ADDRESS (status word 2), and checking the parity bits (U and L in status word 0), then proceeding with further recovery sequences. GU and GL are reset when the status words are read. These bits are only valid on memories without an ECC board.

3.5.2.7 Change State

As described in Section 3.5.2, the CS bit must be set concurrently with the PF, PP or MAP bits in order for the MMS to change state when an RTNIV instruction is executed. If the CS bit is not set, it will not enter (or change) mapped mode, but other functions may be changed as required. CS bit is not required to enter DMA mapping or protect mode.

3.5.2.8 Enable Program Mapping and Page Fault Detection (PF)

Program mapping and page fault detection is enabled by setting the PF bit (7). Concurrently, the MAP bit (Section 3.5.2.10) is set to define the MAP. A typical command value to enable mapping via MAP 0 is as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	X	0	0	1	1	X	0	0	X	X	0	0
I	NI	DPF	DME	DPP	GU	GL	CS	PF	PP		MAP	PME	PTO		

(Bits marked X are set or reset as required for other functions.)

Prior to issuing the command, the map cells of the MAP 0 will have been loaded (Section 3.5.1) with the necessary data. After the command value is output to MMS, and RTNIV actuates the mapping. Logical addresses are then translated to physical addresses, via the MAP 0, provided that the UM bit in the map cell = 0 for the page being translated. If UM bit = 1, a page fault error results because translation via an unmapped page was attempted.

3.5.2.9 Enable Program Protect (PP)

Program protect for a map is enabled by setting the PP bit (6). The protection at the 1K page level is further controlled by the WP and EP bits in the map cell. If the WP bit is set, the page is write protected. Any instructions that attempt to modify memory on that page will cause a program protect error. If the EP bit is set, any attempt to fetch an instruction from a memory location on the page will cause a program protect error. The GA-16/240, unlike the GA-16/440, has no means of aborting an instruction. The instruction fetch causing the program protect error will proceed to completion, the instruction will be executed and the NI interrupt will occur at the end of the instruction.

3.5.2.10 Select Program Protect Map (MAP)

One of the two maps are enabled for program protect by setting the MAP bit 4 as follows:

MAP 0 = 0
MAP 1 = 1

Mapping remains in effect until an interrupt or clear MMS instruction occurs. A single cycle instruction may enable mapping all data references between two instructions through any map without changing the mapping enable bit.

3.5.2.11 Enable Program Memory Error Detection (PME)

ECC or parity error detection is enabled by setting the PME bit (3). Memory error detection is in operation in either the T state or in the mapped state as defined by the other bits of the command value. Memory error detection remains in effect until the bit is reset by another command, by a clear MMS instruction, or by an interrupt.

3.5.2.12 Enable Program Timeout

Program timeout is enabled when PTO bit (2) is set. When PTO is enabled, the ISE status is monitored. If ISE is inhibited for more than 960 microseconds, a program timeout causes an NI interrupt. PTO remains enabled until the bit is reset by another command, an interrupt occurs, or a clear MMS instruction is executed.

3.5.3 INDEXED, REAL MEMORY ADDRESSING

This mode of operation allows a program to access data words from any location in real memory. All data references between the next two IFETCH values following a CTRL 6 command, will use information in cell 63 of the currently selected map as an index register value. This value is added to the CPU logical address to obtain a 20-bit address which is used by the MMS to address real memory. Any one word memory reference instruction may follow the command. For example, assuming the MMS is in transparent mode and MMS cell 63 contains X'0116', the sequence:

```
CTRL    6,X'39'

LARS    LABEL
                    LABEL = X'29F4'
SARS    LABEL
```

will cause all registers to be loaded from real memory location X'481F4'. The current MMS mode is restored when the reset IFETCH is encountered by MMS; therefore, the SARS instruction will store the registers in location X'29F4'. It is possible to generate NI interrupts via the map cell accessed.

If the real memory address exceeds the amount of memory on the system an 'out of range' condition will exist and an NI interrupt will be caused by the MMS. The status will reflect an unmapped fault existed. The system memory size is set by switches on the MMS board (S1).

The real memory address was derived as shown below:

```
Index reg value (page 63)          0100010110

CPU logical memory address              0010100111110100

Real memory address                0100100000111110100
                                     4   8   1   F   4
```

NOTE

Use of indexed, real memory addressing or single cycle mode of operation with interruptable instructions can cause unpredictable results.

3.5.4 SINGLE-CYCLE INSTRUCTIONS

This facility allows a program to access data words through any MAP. All data references between the next two IFETCH values utilize the map specified by the CTRL 3 or CTRL 4 command. Any one-word instruction may follow the command. For example, assuming the MMS is in transparent mode, the sequence:

```
CTRL    4,X'39'

LARS    REGSAV

SARS    REGSAV      REGSAV = X'1000'
```

causes all registers to be loaded using cell 4 of MAP 1 to translated logical address X'1000'. The current MMS mode is restored when the next IFETCH is encountered by MMS; therefore, the SARS instruction will store registers indirectly via logical address X'1000'. It is possible to generate NI interrupts in the map accessed.

3.5.5 CLEAR MMS INSTRUCTION

The clear MMS instruction resets all bits in the command value mask and returns MMS to the T state.

3.5.6 ECC MODE SELECTION

Systems with greater than 64K of semiconductor memory will utilize ECC instead of parity bits to verify memory data integrity. (With 64K memory, ECC is an option.) ECC operation is controlled by DTOM/DTOR commands being sent to the MMS. The command value sent to the MMS (but not after a CTRL 1 or 2 type instruction) contains the ECC MODE control bits as shown below:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	X	X	E	C	C	X	X	X	X

	0	0	0	Error Correction Off
(report correctable errors)	0	0	1	Normal 1 Mode
(report only non-correctable errors)	0	1	0	Normal 2 Mode
(status is in location zero)	0	1	1	Input Status
	1	0	0	Write Data
	1	0	1	Not Used
	1	1	0	Not Used
	1	1	1	Not Used

Document 82S00587A has a full description of ECC operation and mode selection for the GA-16/440, but it is also applicable to the GA-16/240.

3.6 MMS INTERRUPTS

Associated with MMS is one NI hardware interrupt called Memory Management. It may occur in MMS for a variety of reasons. If it occurs as a result of program execution, it is called synchronous. If it occurs from a non-program event such as DMA or PTO operations, it is called asynchronous. Non-MMS interrupts may cause the MMS to enter transparent mode or a mapped state (depending on the setting of the 'I' bit in the command word) in which a non-MMS interrupt routine is translated into the MAP 1. A single NI interrupt from MMS always causes the MMS to return to the T state. If a second NI interrupt occurs before the interrupt routine can service the first interrupt (and set the NI bit = 1 by issuing a new command value), a memory management stall will occur. The response of MMS to interrupts is predetermined by output of a command value (described in Section 3.5.2, as are the conditions which may cause interrupts).

3.6.1 T STATE

The transparent (T) state is entered when MMS generates an NI interrupt and, depending on the setting of the 'I' bit in the command word, after all inhibitable (IN) interrupts. The T state is unprotected when entered via an interrupt. DMA mapping is not affected. Power-up initializes MMS to the T state. (Refer to Section 3.5.2.1 for entry option.)

3.6.2 M STATE

The "mapped" state is entered after any inhibitable IN interrupt unless the T state option is in effect. The M state is unprotected and mapped through MAP 1, when entered via an interrupt. DMA mapping is not affected. (Refer to Section 3.5.2.1 for entry option.)

3.6.3 MSTAL

The occurrence of a second NI interrupt, before the first NI interrupt is serviced, (assuming that NI had been enabled for the first NI interrupt by setting bit 14 in the command value; if the NI bit is not set any NI interrupt will cause an MSTAL), will cause a special CPU state called Memory Management Stall (MSTALL). In this state, the CPU is busy and cannot execute instructions.

The occurrence of a single MMS NI interrupt clears the MMS NI command value NI (bit 14 = 0). The program may re-enable the interrupt capability by a set command value sequence which sets the NI (bit 14 = 1). Any new NI interrupt occuring, before the first NI's service routine has reset the NI bit, will cause the CPU to enter the MSTAL state. During this state, the run light will be on and all system console interface functions will be disabled. MSTAL is differentiated from PMA (Pulse Monitor Alarm) in that MSTAL does not affect the system safe line, SFEC; PMA does.

3.6.4 MMS NON-INHIBITABLE (NI) INTERRUPT

3.6.4.1 General

The MMS generated NI interrupt is under control of I/O instruction to MMS. This interrupt may be caused by any of the following occurrences provided they have been enabled by a command value (refer to Section 3.5.2):

- Program Timeout (PTO)
- Page Fault (PF)
- Write Protect (WP), DMA or Program
- Memory Error (ME), DMA or Program
- Execute Protect (EP)
- Program Protect (PP)

When the MMS NI interrupt is generated, the T state is entered and control is transferred through vector location X'42' in real memory. See Section 3.6.5 and Table 3-4 for types and options of NI interrupts.

MMS maintains status words that contain the last instruction fetch (IFETCH) address in addition to the last data address and miscellaneous interrupt status bits. The NI interrupt service routine may read the MMS status to obtain total interrupt context, cause, operand address, instruction address, etc. If an instruction execution caused a NI interrupt, the program restart address is the IFETCH address in status word 1 of the MMS (location 281). The address of the instruction preceding the one causing the interrupt is stored in status word 3 (location 283). If the error was not program related (i.e., asynchronous), the restart address may be obtained from real memory locations 7A and 7B (P+1 and ISE storage).

Table 3-4. MMS NI Interrupt Summary
(Refer also to Table 3-2)

Interrupt	Mnemonic	Interrupt Weight	Description	Interrupt Type	
				Asynchronous	Synchronous
Program Timeout	PTO	0	Indicates extended non-interruptable condition (ISE off)	X	
Page Fault	PF	1	Indicates reference to unmapped page (in mapped mode) or out of bounds memory reference		X
Program Protect	PP	2	DMA Write Protect	X	
			or		
			Program Write Protect		X
			or		
			Instruction Execute Protect		X
Memory Error	ME	3	Indicates memory parity or uncorrectable ECC error in a read memory reference		X
			or		
			DMA parity or any correctable ECC error during a read	X	

3.6.4.2 Recovery Considerations

The MMS board has a feature added that enables full recovery at the point of interrupt. This feature is that MMS status word 4 always contains the previous MMS command state prior to any interrupt. That is, the current MMS command state is always transferred to the status word 4 register just prior to any interrupt, and hence just prior to any change in the command state due to the interrupt.

In the case of a NI interrupt occurring immediately after an IN interrupt, the NI service routine can use a status word 4 to return to the point of interrupt with the MMS command status restored intact. This procedure could be used in general to recover any interupt; however, there are easier ways in handling IN interrupts (for example, maintaining a small stack in main memory to tell what level of IN interrupt you are in). It is recommended that recovery by using MMS status word 4 be constrained to NI interrupt routines.

3.6.5 NI INTERRUPT PROGRAM OPTIONS

NI interrupts are of two types: synchronous and asynchronous. Both synchronous NIs and asynchronous NIs cause the interrupt to occur after completion of the instruction. After an asynchronous interrupt a new IFETCH request is not generated by the CPU, hence, the IFETCH value in MMS is invalid for a restart address value. However, X'7A' in real memory contains the interrupted program's restart address and X'7B' in real memory contains the state of ISE. After a synchronous interrupt the memory address of the instruction causing the error is contained in MMS status word 1 and thus this is the restart address. The machine context is as follows:

NI Interrupt Type	Program Restart	Memory Reference Causing NI
Synchronous	IFETCH in MMS	MMS Status Word 1 or 2
Asynchronous	X'7A' in Real Memory	MMS Status Word 2

The interrupts included in the NI group are summarized in Table 3-4. All NI interrupts and the services they represent can be masked individually by a program outputting a command value to MMS as described in Section 3.5.2. Additionally, program protect interrupts require further enabling in 1K memory blocks by the map cell bits WP or EP as described in Section 3.4.1

3.6.5.1 Program Timeout (PTO)

MMS provides a program timeout function which monitors two conditions: the state of ISE and the execution of non-interruptable instructions. ISE must be set for the duration of at least two conservative interruptable instructions in order to reset the PTO timing function. If ISE is on, the timing function is reset each time an interruptable instruction is executed, thus preventing endless JMP and JSR loops.

The PTO timer is inhibited for counting when a MMS NI interrupt occurs, or an IN interrupt occurs, or for a program trap NI interrupt. The timer can be restarted by issuing the appropriate MMS command re-enabling the PTO timer.

The timing period is preset to 1125 microseconds by the PTO jumpers on the MMS board at location U8C. Figure 3-14 includes a tabulation that shows how the PTO jumpers can be set to establish periods other than 1125 microseconds.

		JUMPER NUMBER			
		1	2	3	4
PTO TIME IN μSEC	1200	AB	BC	BC	BC
	1125	BC	AB	BC	BC
	960	BC	BC	AB	BC
	640	BC	BC	BC	AB

200 SERIES MMS BOARD DETAIL AT LOCATION U8C:

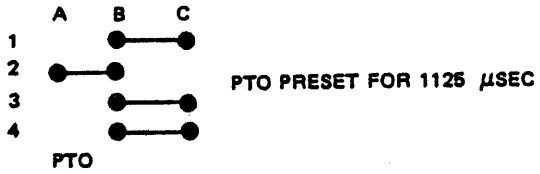


Figure 3-14. Jumper For PTO Timing

3.6.5.2 Page Fault Interrupts (PPF or PF)

This interrupt is generated if MMS detects a logical memory reference via an unmapped MMS cell (bit 15=1 in a referenced map cell). This may also be caused by DMA I/O attempting a logical memory reference via an unmapped MMS cell. Page faults can also be caused when using the indexed real memory addressing feature and a real memory reference is attempted to a location above the available memory address range. Page faults can only occur when operating in mapped mode. Memory will be protected against write operations via an unmapped MMS cell.

3.6.5.3 Program Protect Violation (PP)

This interrupt is generated if MMS detects a write memory reference to a protected page from either the CPU or DMA. It also is generated if the CPU attempts to execute an instruction in an execute protected page. When operating in mapped mode, the translated address via the map cell is protected. When in transparent mode, the MAP protects pages real memory from 0 to 63K in accordance with map cell (0 through 63) WP and EP bits.

3.6.5.4 Memory Error (ME)

This interrupt is generated if MMS detects a parity or ECC error in a memory reference. (Refer to Tables 3-2 and 3-4.)

3.7 MEMORY PROTECT

Protection of memory above 64K is always accomplished when operating in the transparent mode because only the lower 64K is accessible. Protection of any memory in 1K pages is accomplished by first setting the desired protect bits in the map cells by using a load map sequence (Section 3.5.1). The mapped mode is then enabled with a command value sequence (Section 3.5.2) in which the appropriate protect options are specified. Locations X'0' - X'F' are never write protected even though included in a write protect block. Protection may be enabled with transparent or mapped addresses.

3.8 DMA CONSIDERATIONS

DMA mapping is accomplished via the MAP 1. All I/O will be under the control of the operating system, which will ensure memory allocation prior to device initiation. Hence, page faults or protect errors can be attributed to system malfunction in the system software, DMA channel, or the device controller. These malfunctions are, in general, not recoverable but should be serviced to aid troubleshooting. These errors generate an NI interrupt so that the operating system can take corrective action. Note that NI interrupts caused by DMA I/O are not under control of the user's ISE, but, since P is saved in MMS, the user may resume in the event that the NI was not catastrophic and did not cause a system crash. The I/O NI is inserted in the instruction stream at a time when P may be retried without loss of validity. DMA I/O may occur in the transparent mode with no protection, since these facilities are programmable by commands to MMS.

DMA may occur either with or without mapping and protection depending on the current MMS command value. If no DMA mapping is specified, no address translation is performed. If mapping is specified, MAP 1 is employed. The protect feature (PP) may be independently selected. If specified, MAP 1 will be used with or without address translation, depending on the mapped/unmapped selection. If no address translation occurs through the MAP 1, its page protect bits will still protect the corresponding LPN in the transparent mode. Unmapped operators will save 100nsec per memory reference.

DMA operation is independent of the CPUs state or mode. MMS senses DMA memory request signals and maps the next memory reference (if DMA mapping is enabled) using MAP 1. If a DMA page fault or protect error occurs as a result of the reference, MMS generates an NI interrupt, which is executed by the CPU before the next, IFETCH value is generated. The program restart address is stored in memory location X'7A' (as is true of all interrupts).

DMA activity is suppressed by MMS by blocking further DMA request to CPU. The NI interrupt service routine may input status, then may output a new command value re-enabling interrupt. The DMA transfer resumes when new command value is output. By stopping DMA pending interrupt service routine action, unnecessary MSTAL halts can be prevented. (Refer to Section 3.6.3.)

3.9 PAGE-TO-MEMORY ADDRESS CONVERSION

Table 3-5 shows page-to-address translation. Any 1024 block specified in the map cell bits for the page will be accessed by the range of logical addresses shown.

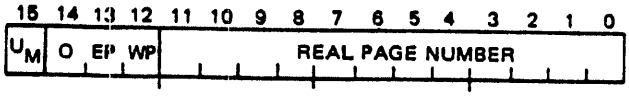
3.10 SUMMARY OF MMS COMMANDS

Figure 3-15 summarizes MMS commands.

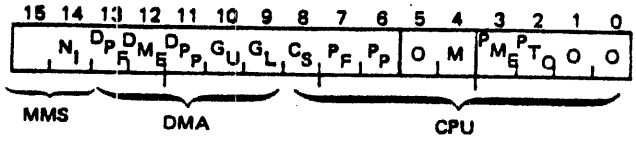
Table 3-5. Page-To-Memory Address Conversion

Page No.		Logical Address Range		Page No.		Logical Address Range	
Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex
0	0	0 - 1023	0000 - 03FF	32	20	32768 - 33791	8000 - 83FF
1	1	1024 - 2047	0400 - 07FF	33	21	33792 - 34815	8400 - 87FF
2	2	2048 - 3071	0800 - 0BFF	34	22	34816 - 35839	8800 - 8BFF
3	3	3072 - 4095	0C00 - 0FFF	35	23	35840 - 36863	8C00 - 8FFF
4	4	4096 - 5119	1000 - 13FF	36	24	36864 - 37887	9000 - 93FF
5	5	5120 - 6143	1400 - 17FF	37	25	37888 - 38911	9400 - 97FF
6	6	6144 - 7167	1800 - 1BFF	38	26	38912 - 39935	9800 - 9BFF
7	7	7168 - 8191	1C00 - 1FFF	39	27	39936 - 40959	9C00 - 9FFF
8	8	8192 - 9215	2000 - 23FF	40	28	40960 - 41983	A000 - A3FF
9	9	9216 - 10239	2400 - 27FF	41	29	41984 - 43007	A400 - A7FF
10	A	10240 - 10263	2800 - 2BFF	42	2A	43008 - 44031	A800 - ABFF
11	B	11264 - 12287	2C00 - 2FFF	43	2B	44032 - 45055	AC00 - AFFF
12	C	12288 - 13311	3000 - 33FF	44	2C	45056 - 46079	B000 - B3FF
13	D	13312 - 14335	3400 - 37FF	45	2D	46080 - 47103	B400 - B7FF
14	E	14336 - 15359	3800 - 3BFF	46	2E	47104 - 48127	B800 - BBFF
15	F	15360 - 16383	3C00 - 3FFF	47	2F	48128 - 49151	BC00 - BFFF
16	10	16384 - 17407	4000 - 43FF	48	30	49152 - 50175	C000 - C3FF
17	11	17408 - 18431	4400 - 47FF	49	31	50176 - 51199	C400 - C7FF
18	12	18432 - 19455	4800 - 4BFF	50	32	51200 - 52223	C800 - CBFF
19	13	18456 - 20479	4C00 - 4FFF	51	33	52224 - 53247	CC00 - CFFF
20	14	20480 - 21503	5000 - 53FF	52	34	53248 - 54271	D000 - D3FF
21	15	21504 - 22527	5400 - 57FF	53	35	54272 - 55295	D400 - D7FF
22	16	22528 - 23551	5800 - 5BFF	54	36	55296 - 56319	D800 - DBFF
23	17	23552 - 24575	5C00 - 5FFF	55	37	56320 - 57343	DC00 - DFFF
24	18	24576 - 25599	6000 - 63FF	56	38	57344 - 58367	E000 - E3FF
25	19	25600 - 26623	6400 - 67FF	57	39	58368 - 59391	E400 - E7FF
26	1A	26624 - 27643	6800 - 6BFF	58	3A	59392 - 60145	E800 - FBFF
27	1B	27647 - 28672	6C00 - 6FFF	59	3B	60416 - 61439	EC00 - EFFF
28	1C	28672 - 29696	70C0 - 73FF	60	3C	61440 - 62463	F000 - F3FF
29	1D	29696 - 30719	7400 - 77FF	61	3D	62464 - 63487	F400 - F7FF
30	1E	30720 - 31743	7800 - 7BFF	62	3E	63488 - 64511	F800 - FBFF
31	1F	31744 - 32767	7C00 - 7FFF	63	3F	64512 - 65535	FC00 - FFFF

MAP CELL FORMAT

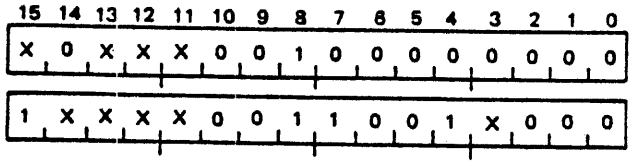


COMMAND VALUE



INTERRUPT TO "T"

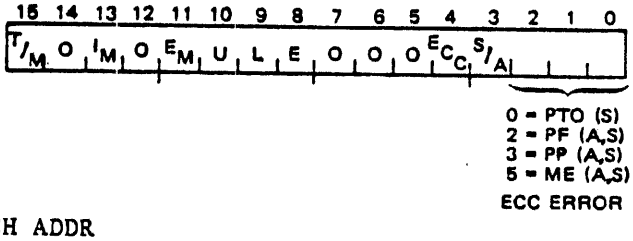
INTERRUPT TO "MI"



*MMS INT. ONLY. NO CHANGE ON NON-MMS INT.

STATUS

WORD 0



WORD 1 = IFETCH ADDR
WORD 2 = DATA ADDR
WORD 3 = PRECEDING IFETCH ADDR
WORD 4 = PRIOR COMMAND VALUE
(EXCEPT BITS 10, 9, 8
5, 4, 1, 0 = 0)

INSTRUCTIONS

X'1139'	CTRL 1, X'39'	= SET COMMAND VALUE
X'1339'	CTRL 3, X'39'	= SINGLE CYCLE MAP 0
X'1439'	CTRL 4, X'39'	= SINGLE CYCLE MAP 1
X'1639'	CTRL 6, X'39'	= INDEXED, REAL MEMORY ADDRESSING
X'1739'	CTRL 7, X'39'	= CLEAR MMS
X'1X79' (FOLLOWING CTRL 1)		= OUTPUT COMMAND VALUE
X'1X79' (NOT FOLLOWING CTRL 0 or 1)		= OUTPUT ECC MODE CONTROL

Figure 3-15. MMS Command Summary

COMMENT SHEET

Document No. _____
Title: _____
Page: _____

FROM:

NAME: _____

BUSINESS ADDRESS: _____

_____

_____

Does this publication meet your requirements? Yes ☐ No ☐

If no, please explain. _____

_____

_____

_____

_____

Do you wish a reply? Yes ☐ No ☐

COMMENTS: (Describe any errors, suggested ideas, additions, or deletions etc. Please include page number.)
All comments and suggestions become the property of General Automation, Inc.

_____

_____

_____

_____

_____

_____

_____

_____

No Postage Stamp Necessary If Mailed In the U.S.A.
(See Other Side)
Fold On Dotted Lines And Staple

YOUR COMMENTS, PLEASE . . .

This publication serves as a reference for systems analysts, programmers and operators of General Automation systems. Your answers to the questions on the back of this form help us produce better publications for your use.

FOLD

First Class
Permit No. 423
Anaheim, California

BUSINESS REPLY MAIL NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

General Automation, Inc.
1055 South East Street
Anaheim, California 92803

Attention: Publications Services

FOLD

CUT ALONG LINE

General Automation, Inc.
1055 South East Street
Anaheim, California 92803