HP 12731A
MEMORY EXPANSION
MODULE

THEORY OF OPERATION

NOTE

This document is part of the HP 1000 M, E, and F-Series Computers Engineering and Reference Documentation and is not available separately.

Table of Contents

INTRODUCTION TO MEMORY EXPANSION

CHAPTER 1

1.1 INTRODUCTION

This document provides the theory of operation for the 12731A Memory Expansion Module (MEM) and the dynamic mapping instructions (DMI). Discussion is conducted on the functional block diagram and detailed circuit operation levels.

1.2 RELATED DOCUMENTATION

The 12731A MEM interacts with all functions of the HP 1000 M, E, and F-Series Computers. In order to understand the operation of the MEM, it is essential to understand CPU timing, assembly language decoding and microprogramming, memory operation, DCPC timing and control, memory protect, and the interrupt system. A suggested course of study is:

a. CPU theory of operation with emphasis on timing, memory addressing, and the interrupt system (Section I).

b. Memory controller theory of operation (Section II).

c. Dual Port Channel Controller theory of operation with emphasis on control and timing (Section III).

d. Memory Protect theory of operation paralleled with MEM theory of operation (Section IV).

e. HP 1000 E-Series and F-Series Computer Microprogramming Reference Manual, part no. 02109-90004.

1.3 GENERAL DESCRIPTION

The 12731A memory expansion module provides added capabilities to the HP 1000 computer memory system. The basic bus structure of the HP 1000 computer allows addressing of only 32K 16-bit words (2 to the 15th) of logical and physical memory. The 12731A combined with the dynamic mapping instructions allows physical memory to be expanded to 1M words (2 to the 20th). This expansion is accomplished by "mapping" logical memory into physical memory on a page by page basis (1 page = 1024 words). This is accomplished through four dynamically alterable maps which provide the additional address bits necessary to address 1M words. A programmable fence enables MEM to divide the base page into mapped and unmapped portions. In addition to the memory expansion and base page capabilities, the MEM combined with the memory protect module provide read and/or write protection for every 1K page of memory, special protection of the unmapped portion of the base page, and a "protected mode" in which many privileged DMS instructions are not allowed.

2.1 MEM CONTROL

The MEM circuitry is controlled by a set of 38 machine-language instructions. These instructions are implemented in microcode ROMs which are part of the Dynamic Mapping System. The microcode instructions are decoded by the CPU which generates control signals to the MEM to carry out varies DMS functions. Data and status information is transmitted and received by the MEM via the 16-bit S-bus.

2.2 MAP REGISTERS

The map registers are an array of 128 registers divided into four dynamically alterable maps. Each map consists of 32 12-bit wide registers. Each register stores a 10 bit page address and 2-bits to specify read and/or write protection on a page by page basis. The purpose of the memory expansion maps is to provide the HP 1000 with the capability to address more physical memory than the basic bus structure would allow. The basic 15-bit memory address bus of the 1000 provides address for logical memory of 32K words. This area is addressed as 32 pages of 1024 words (1K) each. See Figure 2-1.

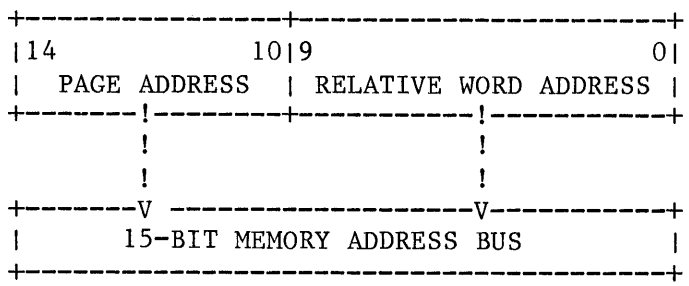


Figure 2-1. Basic Memory Address Scheme

Using the Dynamic Mapping System, the 5-bit page address is used to address one of the 32 registers within a "MEM map" instead of addressing a page in memory. The 10-bit page address contained in the addressed map register is then combined with the 10-bit relative word address to form a 20-bit memory address which is then applied to memory (See Figure 2-2). In this way the 32K word logical memory address space can be "mapped" into 1M word of physical memory.

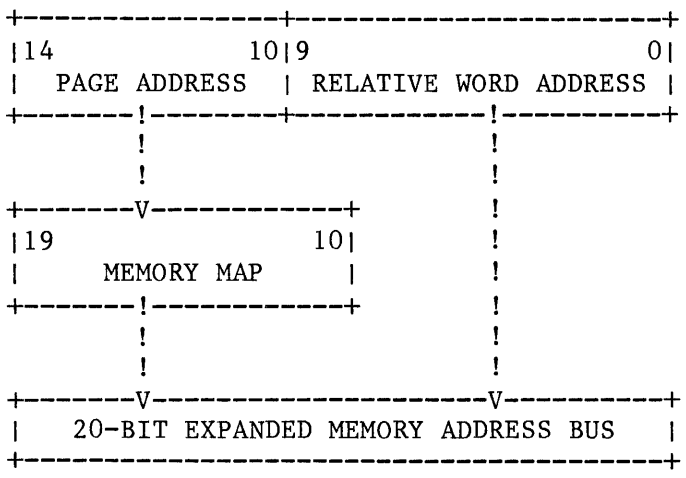
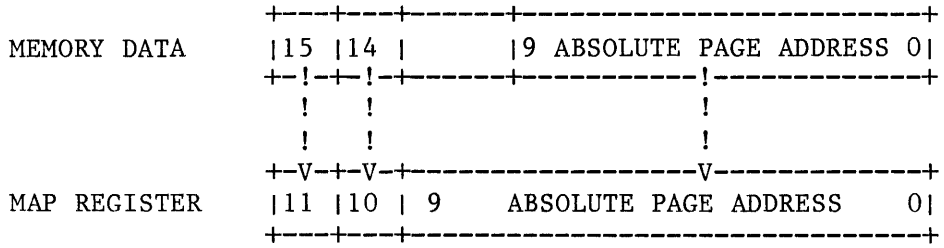


Figure 2-2. DMS Address Translation

2.2.1 MAP REGISTER LOADING

The registers in each map are arranged as a set of 32 contiguous addresses. Map registers are written from or read to the 16-bit S-bus under program control. Conversion of the 16-bit word data format to and from the 12-bit map register format is illustrated in Figure 2-3.



Bit 11 set = Read Protection
Bit 12 set = Write Protection

Figure 2-3. Basic Word Format VS Map Register Format

(HP 1000 M/E/F-SERIES ERD)

Because the maps are dynamically loaded, contiguous maps do not neccessarily contain contiguous page addresses. This allows a contiguous block of logical memory to be mapped into non-contiguous physical memory. For example, suppose system map registers 0,1, and 2 were loaded with 0000B, 0077B, and 0010B, then any reference to logical page 0 would be mapped to physical page 0 through map register 0. However, any reference to loagical page 1 would be mapped to physical page 77 through map register 1, and so on. In this way, logical memory can be distributed throughout physical memory either contigously or non-contigously. (See Figure 2-4.)

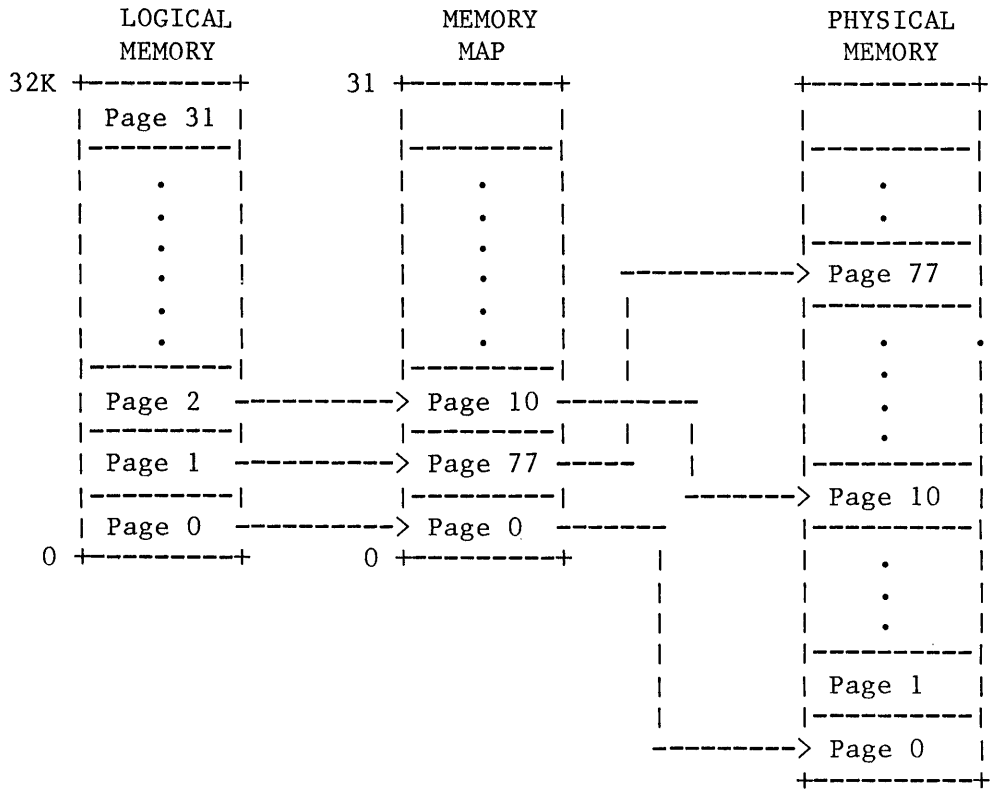


Figure 2-4. Logical to Physical Memory Mapping

2.2.2 MAP CONTROL AND SELECTION

The MEM includes four seperate memory maps: the User Map, System Map, and two Dual-Channel Port Controller Maps (Figure 2-5). The dynamic mapping scheme allows only one map to be selected at a time. This selection is controlled by either the hardware or programmatically. All four maps are loaded or read under program control, but only the system or user map can be enabled for address mapping programmatically.

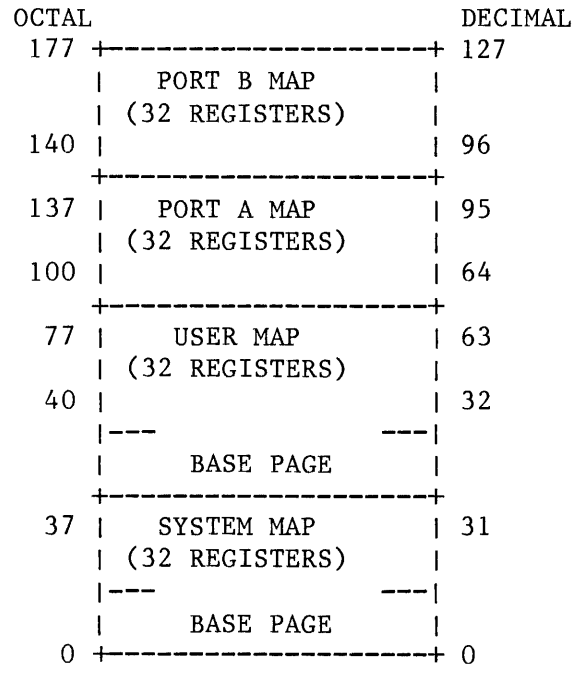


Figure 2-5. Memory Maps and Segmentation

2.2.2.1 USER AND SYSTEM MAP SELECTION AND CONTROL

Once the maps have loaded, the user or the system map can be selected using various dynamic mapping instructions. The execution of one of these instructions turns on the dynamic mapping system and enables the desired map. The selected map is then used to perform all mapping until one of five events occur:

a. DYNAMIC MAPPING SYSTEM DISABLED. DMS can be disabled by execution of a "disable" DMS instruction or by the POPIO signal issued by the CPU. POPIO is issued whenever power is first applied to the computer or whenever the PRESET button is depressed in the HALT mode.

b. ALTERNATE MAP SELECTED. Certain DMS instructions perform memory references using the alternate map. The alternate map is whichever of the two maps is not currently enabled. For example, if the system map is currently enabled the user map is the alternate map. The DCPC maps are never the alternate map. After one of these instructions completes execution, control is returned to the enabled map.

c. PROGRAMATIC MAP SELECTION. If the user map is enabled and an "enable system map" instruction is executed, the user map will be disabled and the system map will be enabled. The reverse is also true.

d. COMPUTER INTERRUPT ACKNOWLEDGE. Whenever the computer acknowledges the occurrance of an interrupt, the MEM hardware automatically selects the system map. This occurs whether the mapping system is enabled or not. The status of the mapping system is saved so that it can be restored after the interrupt is handled by the computer.

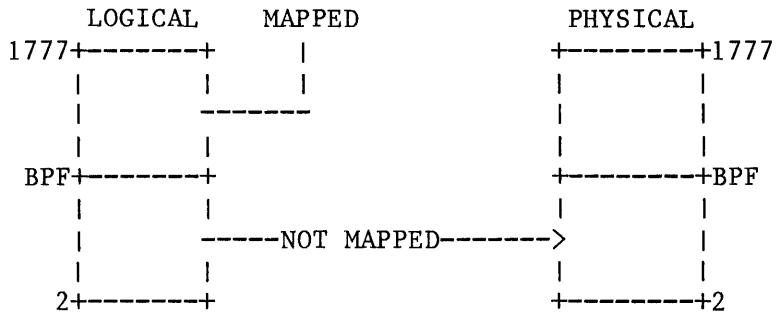
e. A DMA CYCLE REQUESTED. Whenever a DMA cycle is requested by the DCPC the port A or port B map is automatically selected by the hardware. After the transfer is complete control is returned to the enabled map.

2.2.2.2 DUAL CHANNEL PORT CONTROLLER MAPS

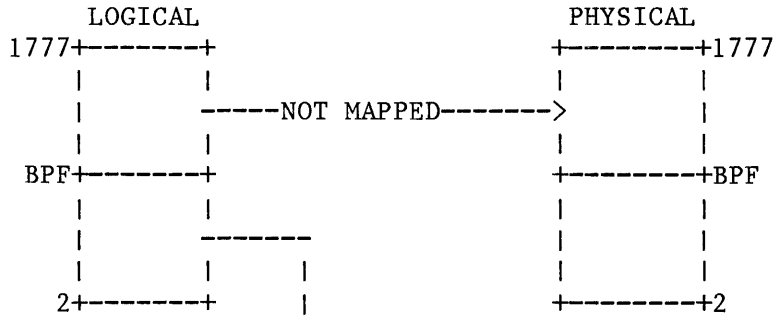
The port A and port B maps are utilized exclusively by the DCPC and cannot be enabled programmatically. The port A map is dedicated to DCPC channel 1, and port B map is dedicated to DCPC channel 2. The maps are enabled by the DCPC hardware whenever a DMA cycle is requested and released immediately after the cycle is complete.

2.3 BASE PAGE FENCE

An added feature of DMS is the ability to divide logical page 0 (base page) into a mapped and unmapped section. The feature is implemented by the base page fence register on the MEM. The fence register is an 11-bit register used to delineate the division point of the base page. Bits 0-9 determine the division address and bit 10 determines whether the portion above or below the fence is to be mapped. Figure 2-6 illustrates the two possible arrangements.



A. MEMORY BELOW BPF NOT MAPPED



B. MEMORY ABOVE BPF NOT MAPPED

Figure 2-6. Base Page Fence Mapping Feature

The base page fence only applies when logical page 0 is addressed by any of the four maps. The fence register is loaded from the S-bus programmtically. It is cleared at power-up, under program control, or by pressing PRESET on the front panel.

2.4 MEM STATUS REGISTER

The MEM status register allows the programmer to determine the status of the MEM logic. The status can be read programmatically or from the front panel of an E-or F-Series computer by examining the f-register. Figure 2-7 is the format of the status register.

BIT	SIGNIFICANCE
15	0=MEM disabled at last interrupt 1=MEM enabled at last interrupt
14	0=System map selected at last interrupt 1=User map selected at last interrupt
13	0=MEM disabled currently 1=MEM enabled currently
12	0=System map selected currently 1=User map selected curently
11	0=Protected mode disabled currently 1=Protected mode enabled currently
10	Portion mapped *
9	Base page fence bit 9
8	Base page fence bit 8
7	Base page fence bit 7
6	Base page fence bit 6
5	Base page fence bit 5
4	Base page fence bit 4
3	Base page fence bit 3
2	Base page fence bit 2
1	Base page fence bit 1
0	Base page fence bit 0

*Bit 10 Mapped address (M)

Figure 2-7. Status Register Format

2.5 PROTECTED MODE

MEM works in combination with memory protect and the memory controller to provide memory protection features necessary to a real time computer environment. The protected mode is enabled whenever the memory protect card is enabled, (STC 05). Note that mapping is still possible in the unprotected mode, (DMS enabled, memory protect disabled).

2.6 MEM VIOLATION REGISTER

The MEM violation register allows the programmer to determine the type of MEM violation that occured in the protected mode and the status of the MEM hardware at the time of the last interrupt. The violation register can be read programmatically by various DMS instructions. Figure 2-8 illustrates the format for the violation register.

BIT	SIGNIFICANCE
15	Read violation*
14	Write violation*
13	Base page violation*
12	Privileged instruction violation*
11	Reserved
10	Reserved
9	Reserved
8	Reserved
7	0=ME bus disabled at violation 1=ME bus enabled at violation
6	0=MEM disbled at violation 1=MEM enabled at violation
5	0=System map disabled at violation 1=System map enabled at violation
4	Map address bit 4
3	Map address bit 3
2	Map address bit 2
1	Map address bit 1
0	Map address bit 0

*Significant when associated bit is set

Figure 2-8. Violation Register Format

2.7 MEM VIOLATIONS

The MEM violations are designed to safeguard DMS. The four types of

violations are read protect, write protect, base page, and privileged instruction. All MEM violations cause an interrupt to select code 5. MEM violations can be distinguished from memory protect or parity error violations by testing the flag on select code 5. The flag will be set only for MEM violations.

2.7.1 READ PROTECT VIOLATIONS

If the computer is in the protected mode and bit 11 of the accessed system map or user map register is 1, then any attempt to read the addressed memory location will result in a read violation and the read will not occur. If the computer is in the unprotected mode or the read is a DCPC read, no violation will occur and the read will take place. In either case, bit 15 of the violation register will be set.

2.7.2 WRITE PROTECTION VIOLATIONS

If the computer is in the protected mode, and bit 10 of the accessed system map or user map register is 1, then any attempt to write to the addressed memory location will result in a write violation and the write will not occur. If the computer is in the unprotected mode, or the write is a DCPC write, no violation will occur and the the write will take place. In either case bit 14 of the violation register will be set.

2.7.3 BASE PAGE VIOLATION

If the computer is in the protected mode, any attempt to write to the unmapped portion of the base page programmatically will result in an base page violation and the write will not occur. If the computer is in the unprotected mode or the base page write is a DCPC write, no base page violation will occur. Base page mapping remains the same however. For example, if a DCPC input transfer were aimed at logical memory addresses 0 to 77777B which happened to map to physical addresses 100000B to 177777B, and the base page fence was set at 1000B, then the input data would be written into physical addresses 100000B to 100777B, 1000B to 1777B, and 102000B to 177777B. Any of the above cases bit 13 of the violation register will be set.

2.7.4 PRIVILEGED INSTRUCTION VIOLATION

If the computer is in the protected mode, any of the following will result in a privileged instruction violation.

a. Any attempt to load a map register.

b. Any attempt to load any register other than the MEM address register with the user map enabled.

If the computer is in the unprotected mode the loads will occur. In either case bit 12 of the violation register will be set.

(HP 1000 M/E/F-SERIES ERD)

DETAILED THEORY OF OPERATION	CHAPTER 3

NOTE: Throughout this section references are made to the schematic diagrams for the MEM hardware, 12731-60001-51,52, and 53, to the Dynamic Mapping Instructions contained in your computer operating and reference manual, and to the DMS microcode given in Appendix A. In order to follow the explainations of circuit operation it is recommended to have these items available. Also, low true signals are shown on the schematic drawing with an overscore or "not" true indicator, however, in text the "not" indicator is expressed as a minus sign following the nmemonic (e.g., P4NF-)

3.1 MEM CONTROL

The 12731A is controlled by a set of 38 assembly language instructions implemented by microroutines. These microroutines perform all register loading and reading, as well as enabling most MEM functions.

3.1.1 INSTRUCTION DECODING

The DMS microinstructions contain standard micro-orders for controlling the S-bus, M-bus, and other computer hardware, as well as three special micro-orders used exclusively by the memory expansion module. The three micro-orders, "MESP" (in the special field), "MEU" (in the store field), and "MEU" (in the S-bus field) are decoded by the micromachine to generate three signals used to control the MEM hardware. In the Special field, MESP generates MESP-. In the Store field, MEU generates MEST-, and in the S-bus field, MEU generates MEEN-. These three signals are in turn decoded by U112 (D,E11), a 3-to-8 line decoder to produce MEM control lines CL0- through CL7-. Table 3-1 indicates which control line signal is generated by each combination of micro-orders and Table 3-2 indicates the functions performed by each signal. These eight signals used in conjuction with S-bus bits 8 thru 15 implement all functions on the MEM. Only word type I microinstructions are used by DMS.

Table 3-1. MEM Signals Invoked by Microcode

LABEL	OP	SPEC	ALU	STORE	S-BUS	INST. TYPE	CONTROL SIGNAL
&	&	MESP	&	MEU	MEU	Q0	CL0-
&	&	MESP	&	MEU	$	Q1	CL1-
&	&	MESP	&	$	MEU	Q2	CL2-
&	&	MESP	&	$	$	Q3	CL3-
&	&	*	&	MEU	MEU	Q4	CL4-
&	&	*	&	MEU	$	Q5	CL5-
&	&	*	&	$	MEU	Q6	CL6-
&	&	*	&	$	$	--	CL7-

&-Any legal code
*-Any legal code except MESP
$-Any legal code except MEU

Table 3-2. MEM Control Signal Functions

SIGNAL	FUNCTION
CL0-	Generated by Q0 microinstruction 1. Enable SYS/USR map register to S-bus. The map register is determined by MEAR bit 5. 2. Store S-bus into PORTA/PORTB map register. The map register is determined by bit MEAR 7. The map register address is specified by MEAR bits 0 through 4.
CL1-	Generated by Q1 microinstruction Store S-bus into a map register. The map is determined by MEAR bits 5 and 6. The map register address is specified by MEAR bits 0 through 4.
CL2-	Generated by Q2 microinstruction Enable a map register to the S-bus. The map is determined by MEAR bits 5 and 6. The map register address is specified by MEAR bits 0 through 4. S-bus bits 10 through 13 are always low.

(HP 1000 M/E/F-SERIES ERD)

Table 3-2. MEM Control Signal Functions (Continued)

SIGNAL	FUNCTION
CL3-	Generated by Q3 microinstruction Select the opposite, (alternate), program map.
CL4-	Generated by Q4 microinstruction 1. Set the "status command " flag through the next microcycle. 2. Reset to the currently selected program map.
CL5-	Generated by Q5 microinstruction Load the appropriate MEM register (other than map registers) from the S-bus. a. MEM State register (2-bits) b. MEM Fence register (11-bits) c. MEM Address Register (8-bits) The correct register is selected by S-bus bits 13 through 15.
CL6-	Generated by Q6 microinstruction Enable MEM register (other than map register) onto the S-bus. a. Normally enables the Violation register b. If proceeded by CL4- microinstruction, enable Status register.
CL7-	No MEM cotrol specified, NOP for MEM.

3.2 MAP LOGIC

The MEM maps consist of twenty-four 16 x 4-bit static RAMs, U11-14, U21-24,

U31-34, U41-44, U51-54, and U61-64 (schematic sheet 51). The RAMs are arranged in blocks of three to form eight 12-bit wide register files of 16 registers each. Each RAM in a file shares a common enable line and write line, and the file is read or writen to as a single unit. Data is written to the registers from the S-bus through U25 and U55 (E1,2). Bits 0 through 10 specify the physical page address, and bits 14 and 15 specify read and write protect.

The system and user maps share a commom data output bus as do the port A and port B maps. These two buses are multiplexed through U15, U45, and U65 (D5,7) to drive the ME-Bus for address mapping. It is also possible to put the map register contents on the S-bus through U37 and U77 (G5,7) by executing a Q0 or Q2 microinstruction.

3.3 MAP SELECTION AND ADDRESSING LOGIC

The maps are selected by the map select logic, sheet 53 B-26,27, and 28 through F-26,27,and 28. This logic allows the lower or upper half of a map to be addressed either internally or externally. The maps are addressed internally for loading from or reading to the S-bus, and externally for address mapping purposes.

3.3.1 INTERNAL MAP SELECTION

Internally the maps are selected and addressed by the Memory Expansion Address Register (MEAR) output bits C0 through C6 (U75,76 at F,G22). The function to be performed is determined by MEAR output bit C7 and control lines CL0-, CL1-, and CL2-. The map selection is determined by bits 4 through 6 and the register addressing is performed by bits 0 through 3. Table 3-3 summarizes the internal map control.

Table 3-3. Internal Control of the Maps

CONTROL SIGNAL	MEAR BITS * C7	C6	C5	C4	FUNCTION
CL1-	&	0	0	0	Load lower half of SYS map
CL1-	&	0	0	1	Load upper half of SYS map
CL1-	&	0	1	0	Load lower half of USR map
CL1-	&	0	1	1	Load upper half of USR map
CL1-	&	1	0	0	Load lower half of PORTA map
CL1-	&	1	0	1	Load upper half of PORTA map
CL1-	&	1	1	0	Load lower half of PORTB map
CL1-	&	1	1	1	Load upper half of PORTB map

(HP 1000 M/E/F-SERIES ERD)

Table 3-3. Internal Control of the Maps (Continued)

CONTROL SIGNAL	MEAR BITS * C7	C6	C5	C4	FUNCTION
CL2-	&	0	0	0	Read lower half of SYS map
CL2-	&	0	0	1	Read upper half of SYS map
CL2-	&	0	1	0	Read lower half of USR map
CL2-	&	0	1	1	Read upper half of USR map
CL2-	&	1	0	0	Read lower half of PORTA map
CL2-	&	1	0	1	Read upper half of PORTA map
CL2-	&	1	1	0	Read lower half of PORTB map
CL2-	&	1	1	1	Read upper half of PORTB map
CL0-	0	&	0	0	Read lower half of SYS map, Load lower half of PORTA map
CL0-	0	&	0	1	Read upper half of SYS map, Load upper half of PORTA map
CL0-	0	&	1	0	Read lower half of USR map, Load lower half of PORTA map
CL0-	0	&	1	1	Read upper half of USR map, Load upper half of PORTA map
CL0-	1	&	0	0	Read lower half of SYS map, Load lower half of PORTB map
CL0-	1	&	0	1	Read upper half of SYS map, Load upper half of PORTB map
CL0-	1	&	1	0	Read lower half of USR map, Load lower half of PORTB map
CL0-	1	&	1	1	Read upper half of USR map,

& - Either 0 or 1
* - MEAR bits C0 through C3 address each register file

An example of internal addressing and control of the maps is the DMI assembly language instruction SYA, (load/store the system map per A). This instruction transfers the contents of 32 consecutive memory locations to the 32 system map registers. The starting address in memory is specified by bits 0 through 14 in the A-register, and bit 15 specifies that the data is to be moved from memory to the map registers.

The first step in an SYA instruction is to set up the MEAR to select the system map and point to the first map register, register 0. This is accomplished in microcode by forming the control word 0010000000000000 on the S-bus and executing a Q5 microinstruction which drives CL5- low. The complement of CL5- is NANDed with S13 to generate LOAD- low to the MEAR. The next P5B- loads bits 0-7 of the S-bus into the MEAR. A read of the memory location specified by the A-register bits 0-14 then takes place and the data is placed on the S-bus. The data is stored into map register 0 by executing a Q1 microinstruction which drives CL1- low. CL1- low in turn will generate MEADDR high, (U104-6 at D12). MEADDR and it's complement are used to enable the MEAR to address the maps as follows: MEADDR high is applied to the map

select logic U81 and U84 to enable MEAR bits 5 & 6 to select one of the four maps. The complement of MEADDR generates the signal LOCAL high (U103-10), which qualifies MEAR bit 4 to generate M4-. M4- and it's complement are applied to the map select logic U72 and U73 to select either the upper or lower half of a map. The complement of LOCAL enables driver U74 to generate map address bits M0-, M1-, M2-, and M3- which are applied directly to the eight map register files. The map register is now fully addressed. At P4 of the Q1 microinstruction P4B strobes U109-8 (D13) to assert WE1- low and the data on the S-bus is written into system map register 0. At P5 the rising edge of P5B- clocks the MEAR and increments the address, and the next read of memory takes place to load register 1. This continues until all 32 map registers are loaded.

The proceedure is the same to store the maps to memory except a Q2 microinstruction is used to enable the map register contents to the S-bus, and no map register write occurs.

A special case of internal control is the Q0 microinstruction. This instruction allows the contents of either a system or user map register to be transfered to a port A or port B map register in a single microinstruction. The system or user map is selected and addressed by the MEAR in the same manner as a Q1 or Q2 microinstructions the only difference being that MEADDR is genrerated by CL0-. In addition to generating MEADDR, CL0- along with MEAR bit 7 select either the port A or port B map through the map select logic. At the beginning of the Q0 microinstruction, the contents of the selected system or user map register is placed on the S-bus, and driven back through U25 and U55 to the inputs of the port A or port B map. At P4, P4B strobes U104-12 (D12) to assert WE2- low and the data is stored into the corrosponding port register. This feature allows the contents of one program map to be loaded into a port map quickly by a single DMS instruction.

3.3.1.1 EXTERNAL SELECTION OF THE PROGRAM MAPS

The two program maps are selected for address mapping by executing one of the DMS map enable instructions (See Appendix A for list of the DMS instructions). The actual selection is accomplished by loading the 2-bit MEM state register, U124 (E,F12). One bit of the state register selects the map to be used for address mapping, system or user, and the other bit turns the mapping system on or off. Refer to paragraph 3-34 NON-MAP registers for a detailed explaination of loading the state register. Once loaded, bit 0 of the state register, U124-8, drives the program map enable logic located at E-13 and F-13. A detailed explaination of this logic will be discussed later. For now it is sufficient to say that the output U87-13 determines the map selection. This signal, USR, and it's complement, SYS, drive the map selection logic U92-13 and U92-3. As an example, if the system map is selected by the state register, then USR is low and SYS is high. If DMA is not in progress and the MEAR is not selected, NDMA and NME high, then SYS high will generate SYST high. SYST high will generate PROGEN- low. PROGEN-

drives the select line for the output multiplexors U15, U45, and U65 (D5-7). Driven low it will enable the system/user output bus to the ME-Bus. SYST high also serves as a qualifier for SY1- and SY0-. SY1- or SY0- will be driven low to select either the upper or lower system map register file by M4-, U94-8 (G24). For internal selection of a register file the sense of M4- was determined by MEAR bit 4. However, in this case MEAR bit 4 is prevented from generating M4- because LOCAL is low, and the sense of M4- is determined by MB14, U94-5. LOCAL low also disables MEAR bits 0 through 3 by driving U74-1 high, and enables MB10 through MB13 to address the selected register file through multiplexor U35 (F9). Note that the A inputs of U35 are permanently enabled and LOCAL low only serves as an output enable signal. The selection is now complete, the map is enabled by the state register, and the desired map register can be addressed by MB 10 through MB 14 when the mapping system is enabled. This is accomplished by the second bit of the state register, U124-6 (MAPON). MAPON high serves as the second qualifier for MEBEN- through U123-13 (C17). The other qualifier is the MEM fence logic which will be disscussed later. MEBEN- low instructs the memory controller to use the 10-bit ME-bus for the upper memory bits instead of M-bus bits 10 through 14.

Once a program map has been enabled and the mapping system turned on, all memory references will be mapped through the selected map until one of the following events takes place:

a. The mapping system is turned off,or the other program map is selected programmatically. The mapping system can be turned off by executing one of the "disable MEM" instructions. These instructions will re-load the state register to set MAPON low. This will set MEBEN- high instructing the memory controller not to use the ME-bus for addressing. The other program map can also be selected by re-loading state register bit 0.

b. The alternate map is selected. In many applications it is desirable to switch from one program map to the other, read or store a word of data, and return to the other map. To do this in assembly language would require three instructions, and would be time consumming. To improve performance, the MEM hardware provides this capability with Q3 and Q4 microinstructions and the program map enable logic located at E-13 and F-13. An example best explains how this capability works.

   In this example, the mapping system is on, the user map is selected, and the following conditions exsist: U117-6 is high as a result POPIO going high at power-up. U117-2 is low and U117-3 is high as a result of CL3- and CL4- both being inactive, high. Under these conditions, the flip-flop does not change states when clocked by P5B-. U124-8 is also high, therefore the output of the EXCLUSIVE-OR gate, U107-6 is low. U107-6 is inverted by U87-13 which drives USR high, selecting the user map. This condition will prevail until the state register is re-loaded, or a Q3 microinstruction is executed. When a Q3 microinstruction is executed CL3- will be driven low and change the J-K inputs of the flip-flop from 01 to 10. The next rising edge of P5B- will cause the flip-flop to toggle and U107-6 will go high. This will result in USR going low, SYS going high,

and the system map being selected. CL3- will return to the inactive state during the next microcycle, and the flip-flop will remain in the new state until a Q4 microinstruction is executed. When a Q4 microinstruction is executed, CL4- will be driven low and change the J-K inputs of the flip-flop from 01 to 11. The next risisng edge of P5B- will set pin 6 high, and U107-6 will go low again. USR will be driven high and the user map will once again be selected.

c. An interrupt is acknowledged by the CPU. Whenever a device generates an interrupt to the computer, the computer acknowledges the interrupt by issuing the signal IAK. At this point it is desirable to have control of the computer switched to the operating system to handle the interrupt. This is accomplished by switching to the system map on the MEM. When IAK U87-11, goes high it immediately forces USR low and selects the system map, if it was not already selected. IAK high also drives U124-10 and U117-5 low presetting both flip-flops. Note also that the rising edge of IAK clocks flip-flops U97A and U97B saving the status of the MEM system. This information can be retrived later to restore the MEM.

d. The DCPC requests a cycle. Unlike the program maps the Port A and Port B maps cannot be enabled for address mapping programmatically. Instead they are turned on by the DCPC whenever a transfer is performed. At the beginning of a cycle, the signal DMAEN- goes low. DMAEN- low drives DMA high through U102-12 (G26). DMA high serves as a qualifier for the port A and port B map select logic, U82-11 (E27) and U82-8 (F27). The selection of the port A or Port B map is determined by MB15. DMAEN- also disables the MEAR from addressing the maps by driving U75-7 (G21) low and U103-9 (G23) high. This allows the map registers to be addressed by M-bus bits 10 through 14.

3.4 NON-MAP REGISTERS

In addition to the map registers there are several other MEM registers. These registers and associated logic provide additional capabilities and functions for the MEM.

3.4.1 THE BASE PAGE FENCE REGISTER

The base page fence register and associated logic provide the capability to divide the logical base page into a mapped and unmapped portion. This is accomplished by comparing the base page address with a known address stored in the base page fence register and turning the mapping system off if the address is in the area to be unmapped. The unmapped area can be defined as either the area above or below the address contained in the fence register.

3.4.1.1 LOADING THE BASE PAGE FENCE REGISTER

The fence register, U16 and U57 (F18), is loaded from the S-bus by the LFA/B microroutine in a two step process. The first step is to set the status command flip-flop, U117 (D-14). This will drive ENFNC, U117-9, high providing one of the two qualifiers necessary for FCLK, U106-12, the fence register clock. The status command flip-flop is set by executing a Q4 microinstruction which will drive CL4- low forcing the J-K inputs to the status command flip-flop to 00. The next rising edge of P5B- will clock the flip-flop and set ENFNC high. ENFNC will be reset by the next P5B-, therefore the load must be completed during the next micocycle. During the next microcycle the data to be loaded into the fence register is placed on the S-bus and a Q5 microinstruction is executed driving CL5- low. CL5- low is NORed with CL5OK- low to issue CL5A high, the second qualifier for FCLK. CL5OK-, U127-6 (E15), is low whenever the system map is selected, SB15 is high or when the MEM is not in the protected mode, (CTL5- high). The proteccted mode is disscussed later in this chapter. At 5 of the Q5 microcycle the falling edge of P4B will create a rising edge on FCLK and the data will be loaded into the fence register. The fence is cleared by POPIO-, U86-16 (G12), low issued at power-up or by pressing the preset button on the computer front panel.

3.4.1.2 TURNING THE MAPPPING SYSTEM OFF WITH THE BPF

Bits 0 through 9 of the fence register form an address which divides each page into a mapped and unmapped section. This address is compared to the lower ten bits of the M-bus on every memory access by comparators U27 and U47 (C18-20). If the memory address is above the fence, then U105-8 is high, if it is below the fence, then U105-8 is low. U105-8 is EXCLUSIVE-ORed with MPBLF the fence sense bit. If MPLBF and U105-8 are either both high or both low, then U107-11 is low. If the access is on the base page, BSPG- low, and the mapping system on, MAPON high, then OFA will be driven low forcing MEBEN- high turning off the mapping system.

3.4.2 THE STATE REGISTER

The 2-bit state register is used to select one of the program maps and turn the mapping system on or off.

3.4.2.1 LOADING THE STATE REGISTER

The state register, U124 (F12), is loaded with SB8- and SB9- driven from the

S-bus by U25 (E1). The actual load is performed by a Q5 microinstruction with S-bus bit 14 high. Executing a Q5 microinstruction drives CL5- low to U116-5 (E16). If CL5OK- is low, then CL5A goes high at U106-10 (F112), and the next P4B will load the register. The state register will be initilized by POPIO to select the system map and turn the mapping system off at power-up or whenever the PRESET button on the front panel is pressed.

3.4.3 THE STATUS REGISTER

The status register, U17 and U67 (F19,20), is not a true register but a tri-statedriver which enables the 11-bit fence register, the 2-bit state register, and the 2-bits of state information saved by the last interrupt on to the S-Bus.

3.4.3.1 READING THE STATUS REGISTER

The status register is read to the S-bus by executing a Q4, then Q6 microinstruction. Executing a Q4 microinstruction forces CL4- low, FIL82 ABORTED setting the status flag flip-flop at P5 of the Q4 microcycle. ENFNC- low serves as one qualifier for RDSTAT, U125-4 (D14), the status register enable local. ENFNC- will remain low throughout the next microcycle which must be a Q6 microinstruction. Executing a Q6 microinstruction will drive CL6- low at U125-5 setting RDSTAT high enabling the status register on to the S-bus throughout the microcycle.

3.5 MEM PROTECTION MODE LOGIC

The MEM works in conjuction with the memory protect and memory controller to provide memory protection features necessary to a real time computer. The protected mode is entered by turning the mapping system on and executing a STC 05 assembly instruction which sets CTL5- low to the MEM.

3.5.1 READ PROTECT VIOLATION LOGIC

The read protect logic allows memory to be read protected on a page by page basis. This protection is controlled by the read protect bit, bit 11 of each map register. Flip-flop U95 (C12), samples the read protect line, RPRO continuously. Whenever a read takes place, READ- goes low and clocks U95. If RPRO is high, then U95-8 will go low. If the protected mode is enabled,

MAPON low and CTL5- low, then U95-8 going low will force METDIS- low. METDIS- low disables the the memory controller from putting data onto the S-bus. If the CPU tries to access the data by issuing TEN- low, then a Read protect viloation will occur, and MEV- will go low the next P4. MEV- low informs the memory protect module that an MEM violation has occurred. Note that DMAENB high will disable the protection logic, this allows DMA reads and writes without MEM violations.

3.5.2 WRITE PROTECT LOGIC

The write protect logic allows memory to be write protected on a page by page basis. This protection is controlled by the write protect, bit 10 of each map register.The write protect line, WPRO, is monitored by U125-8 continuousy. If a write protected page is accessed in the protected mode, and MPCNDB- is low, then a write violation will occur and MEV- will go low the next P4. MPCNDB- goes low from the memory protect module informing MEM to check for write violations.

3.5.3 BASE PAGE VIOLATIONS

If the computer is in the protected mode, any attempt to write to the unmapped portion of the base page will result in a base page violation. BSPG- is monitored continuously by U126-6 (C13). If an access to the unmapped portion to the base page occurs, then BSPG- goes low forcing MEBEN- high. If a write to the this location is attempted, and no memory protect violation takes place, then MPCNDB- will go low and a BPV- will go high. The next P4 will force MEV- low informing the memory protect module that a memory expansion violation has occured.

3.5.4 PRIVLEDGED INSTRUCTION VIOLATIONS

If the computer is in the protected mode, any of the following will result in a priviledged instruction violation.

a. Any attempt to load a map register

b. Any attempt to load any register other than the MEM address register with the user map enabled.

Executing a Q0 or a Q1 microinstruction will drive PRIV high and the next P4

will drive MEV- low if the computer is in the protected mode. The actual write of a map register is prevented by CTL5- low at U104-2 and U104-10(D13) disabling WE1- and WE2- from issuing the write pulse to the map register RAMs. Executing a Q5 microinstruction with CL50K- high will generate PRIV high if the fence register is being loaded or if the state register is being loaded. CL50K-high will prevent the actual load from taking place.

3.6 MICROPROGRAMMING RULES FOR DYNAMIC MAPPING INSTRUCTIONS

Programming the MEM using the special DMS microinstructions Q0 through Q7 requires that certain rules be followed in order to prevent illegal conditions from occuring in the CPU.

a. A read or a write cycle (including "dummy" reads) must be in progress from the CPU before coding "MESP" in the microcode. This is to lock-out DMA interference at T6 which would otherwise "NOP" the "MESP" operation. The "MESP" may appear in the line following the read, write or JMP30. It may also appear in lines that correspond to a two or three T-period read.

b. "MEU" and "MESP" micro-orders may not appear in the same line as a read, write, or JMP30. This is to prevent the ME-bus from changing at the time it is being latched at the memory controller.

c. "MEU" micro-orders in either the S-bus or store field must not appear in the line immediately following any read, write, or JMP30 micro-orders. The reason is the same as in rule number 2.

d. A CPU read or write must be in progress in order to code "MEU in either the S-bus or store fields. DMA lock-out in this case is also necessary as in rule number 1, but this time the reason is to inhibit the micro-order from executing and interfering with the ME-bus during the DMA cycle.

SIGNAL GLOSSARY	CHAPTER 4

4.1 INTRODUCTION

This section defines the input and output signals used for MEM operation. Many of these signals are assigned new names or mnemonics on the MEM board and are described as necessary within the text. All discussion assumes an E-Series computer with high speed memory. Most signals are the same for the E/F-Series and the M-Series computers although the timing may be somewhat different. Refer to the appropriate theory of operation to determine these differences.

4.1.1 INPUT SIGNALS

P4NF- P4 NON FREEZABLE, low true

ORIGIN: CPU

FUNCTION: P4NF- is a negitive true basic clock signal used to trigger events throughout the MEM.

TIMING: P4NF- is generated syncronously by the CPU clock circuitry every microcycle. It is uneffected by freeze conditions or program flow. The signal has a duration of 35ns.

P5NF- P5 NON FREEZABLE, low true

ORIGIN: CPU

FUNCTION: P5NF- is a negitive true basic clock signal used to trigger events throughout the MEM. The rising edge of P5NF- signals the beginning of a new microcycle.

TIMING: P5NF-is generated by the CPU clock circuitry every microcycle. It is uneffected by freeze conditions or program flow. It has a duration of 35ns.

+----------+

FRZFF — FREEZE FLIP-FLOP, high true

ORIGIN: CPU

FUNCTIN: FRZFF is a positive true signal used to inhibit the the basic clocking circuitry of the MEM whenever a freeze condition occurs on the CPU.

TIMING: FRZFF is generted on the CPU at various times. See the CPU theory of operation for a more detailed explanation of freeze conditions.

+----------+

POPIO — POWER ON PRESET I/O, high true

ORIGIN: CPU

FUNCTION: POPIO is a positive true signal used to initialize the MEM to a known state. POPIO disables MEM, selects the system map, clears the MEM status, fence and violation registers.

TIMING: POPIO is generated by the CPU at power up or whenever the PRESET button on the front panel is depressed in the HALT mode.

+----------+

IAK — INTERRUPT ACKNOWLEDGE, high true

ORIGIN: CPU

FUNCTION: IAK is a positive true signal used to select the system map and save the status of the MEM whenever the CPU acknowledges an interrupt.

TIMING: IAK is generated by the CPU during T6 and has a duration of 105 ns.

+----------+

CTL5- — CONTROL 5, low true

ORIGIN: Memory Protect

FUNCTION: CTL5- is a negative true signal used by the MEM logic to enable the "protected mode" circuitry.

(HP 1000 M/E/F-SERIES ERD)

TIMING: CTL5- is generated on the MP module whenever the control flip-flop is set by a STC 05 instruction. See the MP theory of operation for more detail on the CTL5- signal.

+----------+

MPCND- MEMORY PROTECT CONDITIONAL, low true

ORIGIN: Memory Protect

FUNCTION: MPCND is a negitive true signal which signals the MEM logic that a memory write is in progress above the memory protect fence set on the MP. The signal enables the MEM to check for base page or write MEM violations.

TIMING: MPCND is generated on the MP whenever a memory write Occurs that does not violate memory protect rules.

+----------+

READ- MEMORY READ, low true

ORIGIN: CPU or DCPC

FUNCTION: READ is a negative true signal which indicates that a memory read is beginning. It is used to enable the read protect logic on the MEM.

TIMING: READ- is generated on the CPU at the end of a microcycle. It has a duration of 35ns. READ- is generated on the DCPC during T2 of an output transfer and lasts the entire microcycle.

+----------+

TEN- T-REGISTER ENABLE, low true

ORIGIN: CPU or DCPC

FUNCTION: TEN- is a negative true signal used to inform the MEM logic that data is about to be enabled from the T-register to the S-bus. TEN- is a qualifier for a read MEM violation.

TIMING: TEN- is generated on the CPU whenever a TAB micro-order is in the S-bus field after a read. It is generated by the DCPC during T3 of an output transfer.

+----------+

DMAEN- DMA ENABLE, low true

ORIGIN: DCPC

FUNCTION: DMAEN- is a negative true signal used to inform the MEM logic that a DMA cycle is beginning. This signal performs several functions on the MEM. DMAEN- disables the MEM violation logic and selects either the port A or port B map according to bit 15 of the M-bus. It also disables the MEM violation and memory expansion address registers.

TIMING: DMAEN- is generated on the DCPC from P4T6 to P4T4 during both input and output transfers.

+----------+

MEEN- MEMORY EXPANSION ENABLE, low true

ORIGIN: CPU

FUNCTION: MEEN- is one of three negative true signals issued by the micro machine. The MEM decodes these signals to generate one of six control signals which perform various functions on the MEM.

TIMING: MEEM- is generated whenever the MEU micro-order Is in the S-bus field of any microinstruction.

+----------+

MEST- MEMORY EXPANSION STORE, low true

ORIGIN: CPU

FUNCTION: MEST- is one of three control signals issued by the micro machine. The MEM decodes these three signals to generate one of six control signals which perform various functions on the MEM.

TIMING: MEST- is generated whenever the MEU is in the STORE field of any microinstruction.

+----------+

MESP- MEMORY EXPANSION SPECIAL, low true

ORIGIN: CPU

FUNCTION: MESP- is one of three negative true control signals issued

by the micro machine. The MEM decodes these signals to generate one of six control signals which perform various functions on the MEM.

TIMING: MESP- is generated whenever the MESP micro-order is in the SPECIAL field of any microinstruction.

4.2 OUTPUT SIGNALS

+----------+

METDIS- MEM T-REGISTER DISABLE, low true

DESTINATION: Memory Controller

FUNCTION: METDIS- is a negative true signal used to prevent the T-register contents from being enabled to the S-bus after a memory read. The signal is active whenever a read violation occurs in the protected mode.

+----------+

MEV- MEMORY EXPANSION VIOLATION, low true

DESTINATION: Memory Protect

FUNCTION: MEV- is a negative true signal issued whenever a MEM violation occurs in the protected mode. The signal sets the testable flag on select code 5 and causes an interrupt to be generated to that same location.

+----------+

MEBEN- MEMORY EXPANSION BUS ENABLE, low true

DESTINATION: Memory Controller

FUNCTION: MEBEN- is a negative true signal which tells the memory controller to use the ME-bus as the upper ten bits of the current memory address. MEBEN- is active whenever the mapping system is on and the memory access is not in the unmapped portion of the base page.

+----------+

```
PAGE 0003 RTE MICRO-ASSEMBLER REV.2040 800521

0001                    MICMXE,L,C,T
0002                    *
0003                    ********************************************************************
0004                    * (C) COPYRIGHT HEWLETT-PACKARD COMPANY 1979.  ALL RIGHTS      *
0005                    * RESERVED.  NO PART OF THIS PROGRAM MAY BE PHOTOCOPIED,       *
0006                    * REPRODUCED OR TRANSLATED TO ANOTHER PROGRAM LANGUAGE WITHOUT*
0007                    * THE PRIOR WRITTEN CONSENT OF HEWLETT-PACKARD COMPANY.        *
0008                    ********************************************************************
0009                    *
0010                    $NOLIST
0012                    $LIST
0013                    *
0014                             ORG                  %20000
0015                    *
0016                    *
0017                    *
0018                    *
0019                    *
0020                    *
0021                    ************************************************************************
0022                    *                                                                      *
0023                    *        MEMORY EXPANSION UNIT MACRO INSTRUCTIONS                      *
0024                    *        ------ --------- ---- ----- ------------                      *
0025                    *        1978-03-09-1811
0026                    ********************************************************
0027                    *
0028                    INDIRECT EQU                  %251
0029                    HORI     EQU                  %006
0030                    *
0031                    ********************************************************
0032                    *
0033                    *   REGISTER ASSIGNMENTS
0034                    *
0035                    *      S3 :: P-REGISTER
0036                    *      S4 :: MEM CONTROL WORD; MEM ADDRESS REGISTER
0037                    *      S5 :: WORDS AND MAP DATA IN LOOP EXECUTION; MASKS AND CONSTANTS
0038                    *      S6 :: GENERAL PURPOSE SCRATCH
0039                    *
0040                    ************************************************************************
```

```
0042                    *
0043                    *
0044                    *
0045                    *
0046                    *
0047                    ********************************************************************************
0048                    *       ENTRY JUMP TABLE
0049                    ********************************************************************************
0050                    *
0051                    *        MACRO JUMP POINT AND MNEMONIC     BINARY CODE
0052                    *
0053 20000   324 002047 JTABL     JMP                    XMM        1000X011110X0000
0054 20001   377 102047           RTN        CMPS CAB    CAB        QUICK SELF TEST
0055 20002   324 010207           JMP                    MBI        1000X01111000010
0056 20003   324 010147           JMP                    MBF        1000X01111000011
0057 20004   324 011047           JMP                    MBW        1000X01111000100
0058 20005   324 012707           JMP                    MWI        1000X01111000101
0059 20006   324 012647           JMP                    MWF        1000X01111000110
0060 20007   324 013547           JMP                    MWW        1000X01111000111
0061 20010   324 014507           JMP                    SY*        1000X01111001000
0062 20011   324 015047           JMP                    US*        1000X01111001001
0063 20012   324 014607           JMP                    PA*        1000X01111001010
0064 20013   324 014747           JMP                    PB*        1000X01111001011
0065 20014   324 016247           JMP                    SSM        1000X01111001100
0066 20015   324 016507           JMP                    JRS        1000X01111001101
0067 20016   370 036747           RTN                               1000X01111001110
0068 20017   370 036747           RTN                               1000X01111001111
0069 20020   324 002047           JMP                    XMM        1000X011110X0000
0070 20021   324 002057           JMP   STFL             XMM        1000X01111010001
0071 20022   324 004607           JMP                    XM*        1000X01111010010
0072 20023   370 036747           RTN                               1000X01111010011
0073 20024   324 005547           JMP                    XL*        1000X01111010100
0074 20025   324 006007           JMP                    XS*        1000X01111010101
0075 20026   324 006247           JMP                    XC*        1000X01111010110
0076 20027   324 006647           JMP                    LF*        1000X01111010111
0077 20030   010 022447 RS*                   PASS MEU   MEU        1000X01111011000
0078 20031   230 022040 RV*       READ RTN   PASS CAB   MEU        1000X01111011001
0079 20032   324 007047           JMP                    DJP        1000X01111011010
0080 20033   324 007647           JMP                    DJS        1000X01111011011
0081 20034   324 007147           JMP                    SJP        1000X01111011100
0082 20035   324 007547           JMP                    SJS        1000X01111011101
0083 20036   324 007247           JMP                    UJP        1000X01111011110
0084                    *
0085 20037   345 007165 UJS       IMM   DCNT HIGH S4    %103       S4:=USER, CNTR:=16B
0086 20040   324 007704           JMP   RJ30             JS*        START READ ON DEF.
0087                    ********************************************************************************
```

```
0089                        *
0090                        *
0091                        *
0092                        *
0093                        *
0094                        ********************************************************************************
0095 20041  010 033107  XMM                 PASS S3   M         S3 := M; SAVE M
0096 20042  230 070547          READ        PASS CNTR X          CNTR :=  COUNT
0097 20043  360 001602          RTN  CNDX   ALZ                  TEST FOR ZERO COUNT
0098 20044  342 000507          IMM         LOW  L    %200       L  := 1111111110000000
0099 20045  234 007147          READ        SANL S4   A          MASK LOW 7 BITS OF A-REG
0100 20046  347 076507          IMM         HIGH L    %337       L  := 1101111111111111
0101 20047  011 047147                      SONL S4   S4         ADD CONTROL BIT (13)
0102 20050  010 046447                      PASS MEU  S4         MEM ADDR REG := S4
0103 20051  010 011707                      PASS P    B          P  := B(TABLE ADDRESS)
0104 20052  010 070747                      PASS      X          SET ALU FLAGS FROM X
0105 20053  334 003402          JMP  CNDX   FLAG      XMS        TEST FOR XMS INSTRUCTION
0106 20054  327 104142          JMP  CNDX   AL15      READMAP    TEST FOR NEGATIVE COUNT
0107 20055  227 174725  MELOOP1 READ DCNT   INC  PNM  P          READ NEXT WORD; P := P+1
0108                        *
0109 20056  230 001207          READ        PASS S5   TAB        S5 := MAP DATA - DUMMY READ
0110 20057  007 106147                      INC  A    A          A := A+1
0111 20060  010 050452               MESP   PASS MEU  S5         MAP REG := DATA
0112 20061  000 071607                      DEC  X    X          X := X-1
0113 20062  320 003302          JMP  CNDX   ALZ       XMM.RTN    IF DONE THEN BUG OUT
0114 20063  324 042642          JMP  CNDX   CNT4 RJS  MELOOP1    LOOP FOR 16X
0115 20064  335 002642          JMP  CNDX   NINT      MELOOP1    TEST FOR NO INTERRUPT
0116 20065  000 045107  XMM.EXIT            DEC  S3   S3           ELSE SERVICE INTERRUPT
0117                        *
0118 20066  010 074207  XMM.RTN             PASS B    P          RESET B-REG
0119 20067  227 144700  P.RTN   READ RTN    INC  PNM  S3         P  := NEXT INSTRUCTION; START READ
0120                        *
```

```
0122                        *
0123                        *
0124                        *
0125                        *
0126                        *
0127                        ********************************************************************************
0128 20070   327 103342   XMS       JMP  CNDX AL15       P.RTN      TEST FOR X<0 ... NOP
0129 20071   230 036747   MELOOP2   READ                             FOR DCPC
0130 20072   007 106147                  INC  A     A          A := A+1
0131 20073   010 010452             MESP PASS MEU   B          MAP REG := DATA
0132 20074   007 110225             DCNT INC  B     B          B  := B + 1; INC CNTR
0133 20075   000 071607                  DEC  X     X          X := X-1
0134 20076   320 003342             JMP  CNDX ALZ       P.RTN      IF DONE THEN BUG OUT
0135 20077   324 043442             JMP  CNDX CNT4 RJS  MELOOP2    LOOP FOR 16X
0136 20100   335 003442             JMP  CNDX NINT      MELOOP2    TEST FOR NO INTERRUPT
0137 20101   000 045107                  DEC  S3    S3         RESET P REGISTER FOR RESTART
0138 20102   227 144700             READ RTN  INC  PNM  S3            SERVICE INTERRUPT
0139                        *
0140                        READMAP   EQU                      *
0141 20103   227 174726   MELOOP3   READ ICNT INC  PNM  P          P := P+1 - DUMMY READ
0142 20104   007 106147                  INC  A     A          A := A+1
0143 20105   010 023212             MESP PASS S5    MEU        S5 := MAP REG
0144 20106   210 050036             WRTE MPCK PASS TAB  S5         WRITE DATA INTO TABLE
0145 20107   007 171607                  INC  X     X              X := X-1
0146 20110   320 003302             JMP  CNDX ALZ       XMM.RTN        IF DONE THEN BUG OUT
0147 20111   324 044142             JMP  CNDX CNT4 RJS  MELOOP3    LOOP FOR 16X
0148 20112   335 004142             JMP  CNDX NINT      MELOOP3    TEST FOR NO INTERRUPT
0149 20113   324 003247             JMP                 XMM.EXIT   ELSE SERVICE INTERRUPT
```

```
0151                    *
0152                    *
0153                    *
0154                    *
0155                    *
0156                    ********************************************************************************
0157 20114  357 077147  XM*       IMM        CMHI S4    %337       S4 := 0010000000000000
0158 20115  150 002762             LWF  L1    PASS       CAB        T-BUS := A/B; FLAG := A/B(15)
0159 20116  321 145042  PA.PB     JMP  CNDX  ALO  RJS   SY.US      TEST FOR PORT.A MAP
0160 20117  341 176507             IMM        LOW  L     %177       L  := 1111111101111111
0161 20120  231 047147             READ       SONL S4    S4         S4 := 0010000010000000
0162 20121  334 045202  SY.US     JMP  CNDX  FLAG RJS   XFER       TEST FOR SYSTEM MAP
0163 20122  343 076507             IMM        LOW  L     %337       L  := 1111111111011111
0164 20123  231 047147             READ       SONL S4    S4         S4 := 00100000X0100000
0165 20124  010 046447  XFER                  PASS MEU   S4         MEM ADDR REG := S4(7-0)
0166 20125  340 100547             IMM        LOW  CNTR  %40        CNTR := 32
0167 20126  230 036765  XFERLOOP  READ DCNT                       DUMMY READ
0168 20127  010 036747                        PASS                  FOR MEB DELETE WITH DUMMY READ
0169 20130  010 022452                   MESP PASS MEU   MEU        MEM PORT REG := MEM PROG REG
0170 20131  326 145302             JMP  CNDX  CNT8 RJS   XFERLOOP  IF NOT DONE THEN LOOP
0171 20132  230 036740  RTN*      READ RTN                          RETURN
0172                    ********************************************************************************
```

```
0174                        *
0175                        *
0176                        *
0177                        *
0178                        *
0179                        ********************************************************************************
0180 20133  300 012447  XL*       JSB                       INDIRECT  GET OPERAND ADDR FROM INSTR + 1
0181 20134  010 036752                 MESP                            SWITCH MAP STATE
0182 20135  230 036747            READ                                 START CROSS LOAD START CROSS LOAD
0183 20136  010 000052                 MESP PASS CAB  TAB               CAB := DATA, RESET MAPS
0184 20137  227 174700            READ RTN  INC  PNM  P                RETURN TO FETCH
0185                        ********************************************************************************
0186 20140  300 012447  XS*       JSB                       INDIRECT  GET OPERAND ADDR FROM INSTR + 1
0187 20141  010 036752                 MESP                            SWITCH MAP STATE
0188 20142  210 002036            WRTF MPCK PASS TAB  CAB
0189 20143  010 022447                      PASS MEU  MEU               RESET MAP STATE
0190 20144  227 174700            READ RTN  INC  PNM  P                START NEXT INST READ - EXIT
0191                        ********************************************************************************
0192 20145  300 012447  XC*       JSB                       INDIRECT  GET OPERAND ADDR FROM INSTR + 1
0193 20146  010 002512                 MESP PASS L    CAB               L  := A/B; SET ALTERNATE MAP
0194 20147  230 036747            READ                                 READ REAL OPERAND
0195 20150  010 001012                 MESP PASS S1   TAB               S1 := DATA, RESET MAPS
0196 20151  227 174707            READ      INC  PNM  P                START READ FOR NEXT INST.
0197 20152  014 140747                      XOR       S1                COMPARE DATA
0198 20153  360 001002            RTN  CNDX ALZ                        RTN-DUN'T SKIP IF EQUAL
0199 20154  227 174700            READ RTN  INC  PNM  P                P  := INSTR + 2; RETURN
0200                        ********************************************************************************
0201 20155  344 016507  LF*       IMM       HIGH L    %007             L  := 0000011111111111
0202 20156  232 003147            READ      AND  S4   CAB               S4 := A/B(10-0) BEWARE THE READ
0203 20157  010 022447                      PASS MEU  MEU               SEND "FENCE" DIRECTIVE
0204 20160  370 046447            RTN       PASS MEU  S4                MEM FENCE := S4
0205                        ********************************************************************************
```

```
0207                        *
0208                        *
0209                        *
0210                        *
0211                        *
0212                        ********************************************************************************
0213 20161  345 001147  DJP       IMM        HIGH S4    %100       S4 := 0100000011111111
0214 20162  324 007307             JMP                   JP*
0215                        *
0216 20163  345 005147  SJP       IMM        HIGH S4    %102       S4 := 0100001011111111
0217 20164  324 007307             JMP                   JP*
0218                        *
0219 20165  345 007147  UJP       IMM        HIGH S4    %103       S4 := 0100001111111111
0220 20166  230 036747  JP*       READ
0221 20167  304 017377             JSB  IOFF             OPGET      GET OPERAND ADDR FROM INSTR + 1
0222 20170  010 046447  JMPSTAT             PASS MEU    S4         MEM STATUS IS SET HERE
0223 20171  227 133736             READ MPCK INC  P      M          CHECK TARGET ; START INST READ
0224 20172  370 036747             RTN                  RETURN
0225                        ********************************************************************************
0226 20173  345 005166  SJS       IMM  1CNT HIGH S4    %102       S4 := 0100001011111111
0227 20174  324 007704             JMP  RJ30             JS*        START READ ON DEF.
0228                        *
0229 20175  345 001144  DJS       IMM  RJ30 HIGH S4    %100       S4 := 0100000011111111
0230                        *
0231                        ORG                   20176B     !!!DO NOT MOVE--INDEXED ENTRY FROM UJS,SJS!!!
0232 20176  304 017377  JS*       JSB  IOFF             OPGET      GET OPERAND ADDR FROM INSTR + 1
0233 20177  010 046447                  PASS MEU    S4         MEM STATUS IS SET HERE
0234 20200  210 074036             WRTE MPCK PASS TAB  P          WRITE RETURN ADDR AT TARGET
0235 20201  007 133707                  INC  P      M          P := TARGET ADDRESS
0236 20202  227 174700  JS*EXIT   READ RTN  INC  PNM  P          P  := TARGET + 1
0237                        ********************************************************************************
```

```
0239                        *
0240                        *
0241                        *
0242                        *
0243                        *
0244                        ********************************************************************************
0245 20203  010 036752  MBF          MESP                                SET ALTERNATE MAP
0246 20204  304 012407  MBI     JSB                          BYTEADJ     ADJUST FOR FULL WORD PROCESSING
0247 20205  304 013007          JSB                          X.LOOP-1    MOVE BYTES IN PAIRS
0248 20206  010 070747                   PASS                X           ALU FLAGS := X CONDITIONS
0249 20207  320 052142          JMP  CNDX ALZ  RJS  B.RESET    TEST FOR INTERRUPTED MOVE
0250 20210  334 052202          JMP  CNDX FLAG RJS  B.RESET+1  TEST FOR NO ODD BYTE
0251 20211  344 000507          IMM       HIGH L    %000        L := 0000000011111111
0252 20212  230 026747          READ      PASS      CNTR        ALO := IR(0); START DCPC READ
0253 20213  321 150642          JMP  CNDX ALO  RJS  *+2         TEST FOR MBI INSTRUCTION
0254 20214  010 036752               MESP                       SET ALTERNATE MAP
0255 20215  230 006647          READ      PASS M    A           M  := SOURCE ADDRESS, RESET MAPS
0256 20216  010 006162               L1   PASS A    A           FORM BYTE ADDRESS IN A
0257 20217  014 001152               MESP SANL S4   TAB         S4 := AAAAAAAA00000000
0258 20220  324 011507          JMP                          MB*
0259                        ********************************************************************************
0260 20221  344 000512  MBW     IMM  MESP HIGH L    %000        SET THE OPPOSITE MAP L := BYTE MASK
0261 20222  304 012407          JSB                          BYTEADJ     ADJUST FOR FULLWORD PROCESSING
0262 20223  304 013647          JSB                          W.LOOP-1    MOVE BYTES IN PAIRS
0263 20224  010 070752               MESP PASS              X           ALU := X; SELECT ALTERNATE MAP
0264 20225  320 052142          JMP  CNDX ALZ  RJS  B.RESET    TEST FOR INTERRUPTED MOVE
0265 20226  334 052202          JMP  CNDX FLAG RJS  B.RESET+1  TEST FOR NO ODD BYTE
0266 20227  230 006647          READ      PASS M    A           M  := SOURCE ADDRESS
0267 20230  010 006162               L1   PASS A    A           FORM BYTE ADDRESS IN A
0268 20231  014 001147                    SANL S4   TAB         S4 := AAAAAAAA00000000
0269                        *
0270 20232  230 010647  MB*     READ      PASS M    B           M  := DESTINATION ADDRESS
0271 20233  010 010222               L1   PASS B    B           FORM BYTE ADDRESS IN B
0272 20234  012 000507                    AND  L    TAB         L  := 0000000088888888
0273 20235  010 147147                    IOR  S4   S4          S4 := AAAAAAAABBBBBBBB
0274 20236  210 046036          WRTE MPCK PASS TAB  S4          WRITE DATA INTO DESTINATION
0275 20237  007 106147                    INC  A    A           A  := A + 1
0276 20240  010 022447                    PASS MEU  MEU         RESET SELECTED MAP
0277 20241  007 110207                    INC  B    B           B  := B + 1
0278 20242  227 144700          READ RTN  INC  PNM  S3
0279                        ********************************************************************************
0280 20243  150 071622  B.RESET LWF  L1   PASS X    X           RESET X IN BYTES
0281 20244  010 022447                    PASS MEU  MEU         RESET SELECTED MAP
0282 20245  227 144707          READ      INC  PNM  S3          EXIT
0283 20246  010 006162               L1   PASS A    A           RESET A FOR EVEN BYTE ADDRESS
0284 20247  370 010222          RTN  L1   PASS B    B           RESET B FOR EVEN BYTE ADDRESS
0285                        ********************************************************************************
0286 20250  010 033116  BYTEADJ      CLEL PASS S3   M           SAVE M FOR NEXT INST FETCH
0287 20251  010 006164               R1   PASS A    A           A := SOURCE WORD ADDRESS
0288 20252  010 010224               R1   PASS B    B           B := DESTINATION WORD ADDRESS
0289 20253  150 071624          LWF  R1   PASS X    X           X := WORD COUNT. FLAG := ODD BYTE
0290 20254  370 070747          RTN       PASS      X           SET ALU FLAGS FOR TESTING X
```

```
0292                        *
0293                        *
0294                        *
0295                        *
0296                        *
0297                        ********************************************************************************
0298 20255  010 036752  MWF             MESP                            FLIP THE MAP SO IT WILL COME OUT RIGHT
0299 20256  010 033107  MWI                     PASS S3     M           SAVE M FOR NEXT INST
0300 20257  010 070747                         PASS        X           ALU FLAGS := X CONDITIONS
0301 20260  320 014402              JMP  CNDX  ALZ         MW*         TEST FOR X=0
0302 20261  230 006647  X.LOOP      READ       PASS M      A           READ SOURCE WORD
0303 20262  007 106147                         INC  A      A           INCR. SOURCE ADDR.; SWITCH MAPS
0304 20263  010 010652             MESP       PASS M      B           M.P. CHECK,M := DEST ADDR
0305 20264  010 001147                         PASS S4     TAB         S4 := DATA
0306 20265  210 046036              WRTE MPCK  PASS TAB    S4          WRITE DATA INTO DESTINATION
0307 20266  007 110207                         INC  B      B           INCREMENT DESTINATION ADDRESS
0308 20267  000 071612             MESP       DEC  X      X           DECREMENT COUNT; SWITCH MAPS
0309 20270  320 014402              JMP  CNDX  ALZ         MW*         TEST IF MOVE COMPLETE
0310 20271  335 013042              JMP  CNDX  NINT        X.LOOP      TEST FOR NO INTERRUPT
0311 20272  324 014347              JMP                    MWINT
0312                        ********************************************************************************
0313 20273  010 033112  MWW             MESP       PASS S3     M           SAVE M FOR NEXT INST FETCH
0314 20274  010 070747                         PASS        X           T-BUS := X
0315 20275  320 014402              JMP  CNDX  ALZ         MW*         TEST FOR X=0
0316 20276  230 006647  W.LOOP      READ       PASS M      A           READ SOURCE WORD
0317 20277  007 106147                         INC  A      A           INCREMENT SOURCE ADDRESS
0318 20300  010 001147                         PASS S4     TAB         S4 := DATA
0319 20301  010 010647                         PASS M      B           M.P.CHECK; M := DEST ADDRESS
0320 20302  210 046036              WRTE MPCK  PASS TAB    S4          WRITE DATA INTO DESTINATION
0321 20303  007 110207                         INC  B      B           INCREMENT DESTINATION ADDRESS
0322 20304  000 071607                         DEC  X      X           DECREMENT COUNT
0323 20305  320 014402              JMP  CNDX  ALZ         MW*         TEST IF MOVE COMPLETE
0324 20306  335 013702              JMP  CNDX  NINT        W.LOOP      TEST FOR NO INTERRUPT
0325 20307  000 045107  MWINT               DEC  S3     S3          SET P COUNTER FOR INTERUPT EXIT
0326 20310  010 022447  MW*               PASS MEU    MEU         RESET SELECTED MAP; RETURN
0327 20311  227 144700              READ RTN   INC  PNM    S3          START INST FETCH; EXIT
0328                        ********************************************************************************
```

```
0330                        *
0331                        *
0332                        *
0333                        *
0334                        *
0335                        ********************************************************************************
0336 20312  357 077147  SY*       IMM         CMHI S4    %337       S4 := 0010000000000000
0337 20313  324 015207             JMP                    MAPMOVE
0338                        ********************************************************************************
0339 20314  355 175164  PA*       IMM  R1     CMHI S4    %176       S4 := 0100000010000000
0340 20315  010 047164             R1     PASS S4    S4         S4 := 0010000001000000
0341 20316  324 015207             JMP                    MAPMOVE
0342                        ********************************************************************************
0343 20317  342 077147  PB*       IMM         LOW  S4    %237       S4 := 1111111110011111
0344 20320  324 015107             JMP                    US*+1      L  := 1101111111111111
0345                        *                                         S4 := 0010000001100000
0346                        ********************************************************************************
0347 20321  343 077147  US*       IMM         LOW  S4    %337       S4 := 1111111111011111
0348 20322  347 076507             IMM         HIGH L     %337       L  := 1101111111111111
0349 20323  014 147147                         XOR  S4    S4         S4 := 0010000000100000
0350 20324  230 033107  MAPMOVE   READ        PASS S3    M          S3 := M - DUMMY READ
0351 20325  010 046447                         PASS MEU   S4         MEM ADDR REG := S4
0352 20326  340 100547             IMM         LOW  CNTR  32                := 32
0353 20327  010 003707                         PASS P     CAB        P  := A/B
0354 20330  327 115742             JMP  CNDX   AL15       MELOOP5    AL15=1 => READ MAPS
0355                        *
0356 20331  227 174725  MELOOP4   READ DCNT   INC  PNM   P          READ NEXT WORD; P := P + 1
0357 20332  230 001207             READ        PASS S5    TAB        S5 := MAP DATA - DUMMY READ
0358 20333  010 074047                         PASS CAB   P          A OR B := P
0359 20334  010 050452                   MESP  PASS MEU   S5         MAP REG := DATA
0360 20335  326 155442             JMP  CNDX   CNT8 RJS   MELOOP4    LOOP FOR 32X
0361 20336  227 144700             READ RTN    INC  PNM   S3         P  := INSTR + 1
0362                        *
0363 20337  227 174725  MELOOP5   READ DCNT   INC  PNM   P          DEC CNTR P := P + 1  -DUMMY READ
0364 20340  010 074047                         PASS CAB   P          A OR B := P
0365 20341  010 023212                   MESP  PASS S5    MEU        S5 := MAP DATA
0366 20342  210 050036             WRTE MPCK   PASS TAB   S5         WRITE DATA INTO TABLE
0367 20343  326 155742             JMP  CNDX   CNT8 RJS   MELOOP5    LOOP FOR 32X
0368 20344  227 144700             READ RTN    INC  PNM   S3         P  := INSTR + 1
0369                        ********************************************************************************
```

```
0371                        *
0372                        *
0373                        *
0374                        *
0375                        *
0376                        ********************************************************************************
0377 20345  300 012447  SSM       JSB                       INDIRECT  GET OPERAND ADDR FROM INSTR + 1
0378 20346  010 022447                      PASS MEU        MEU       SEND "STATUS" DIRECTIVE
0379 20347  010 023007                      PASS S1         MEU       WRITE STATUS WORD INTO MEMORY
0380 20350  210 040036            WRTE MPCK PASS TAB        S1
0381 20351  324 010107            JMP                       JS*EXIT
0382                        ********************************************************************************
0383 20352  230 036747  JRS       READ
0384 20353  304 017377            JSB  IOFF                 OPGET     GET STATUS WORD FROM ^.INSTR+1
0385 20354  150 001222            LWF  L1     PASS S5       TAB       FLAG := STAT(15); S5(15) := STAT(14)
0386 20355  345 007147            IMM          HIGH S4      %103      S4 := 0100001111111111
0387 20356  230 074647            READ         PASS M       P         SET M FOR SECOND OPERAND
0388 20357  304 017377            JSB  IOFF                 OPGET     GET TARGET ADDR FROM INSTR+2
0389 20360  334 017102  ON.OFF    JMP  CNDX FLAG            SY.USR    TEST IF MEM WAS ON
0390 20361  345 003147            IMM          HIGH S4      %101      IF OFF, S4 := 0100000111111111
0391 20362  230 050747  SY.USR    READ         PASS         S5        AL15 := STAT(14) -DUMMY READ
0392 20363  327 107402            JMP  CNDX AL15            JMPSTAT   TEST STAT(14) FOR USER SELECTED
0393 20364  345 004507            IMM          HIGH L       %102      IF SYS, L  := 0100001011111111
0394 20365  232 047147            READ         AND  S4      S4          THEN  S4 := 010000X011111111
0395 20366  324 007407            JMP                       JMPSTAT   SET STATUS OF MEM; ALSO SET P
0396                        ********************************************************************************
0397 20367  340 006547  OPGET     IMM          LOW  CNTR 003B         SET COUNTER FOR MAXIMUM INDIRECTS
0398 20370  230 000665            READ DCNT PASS M          TAB       M := T/A/B  DEC INDIRECT CNTR
0399 20371  367 140002            RTN  CNDX AL15 RJS                  RETURN IF INDIRECT RESOLVED
0400 20372  326 157402            JMP  CNDX CNT8 RJS        *-2       CONTINUE IF IND. LEVEL <= 3
0401 20373  230 036743            READ ION                            RE-ENABLE INTERRUPT RECOGNITION
0402 20374  323 157342            JMP  CNDX HOI  RJS        OPGET     TEST FOR HALT OR INTERRUPT
0403 20375  320 012747            JMP                       INDIRECT+6 PROCESS PENDING INTERRUPT
0404                        ********************************************************************************
0405                                END
```

END OF PASS 2: NO ERRORS

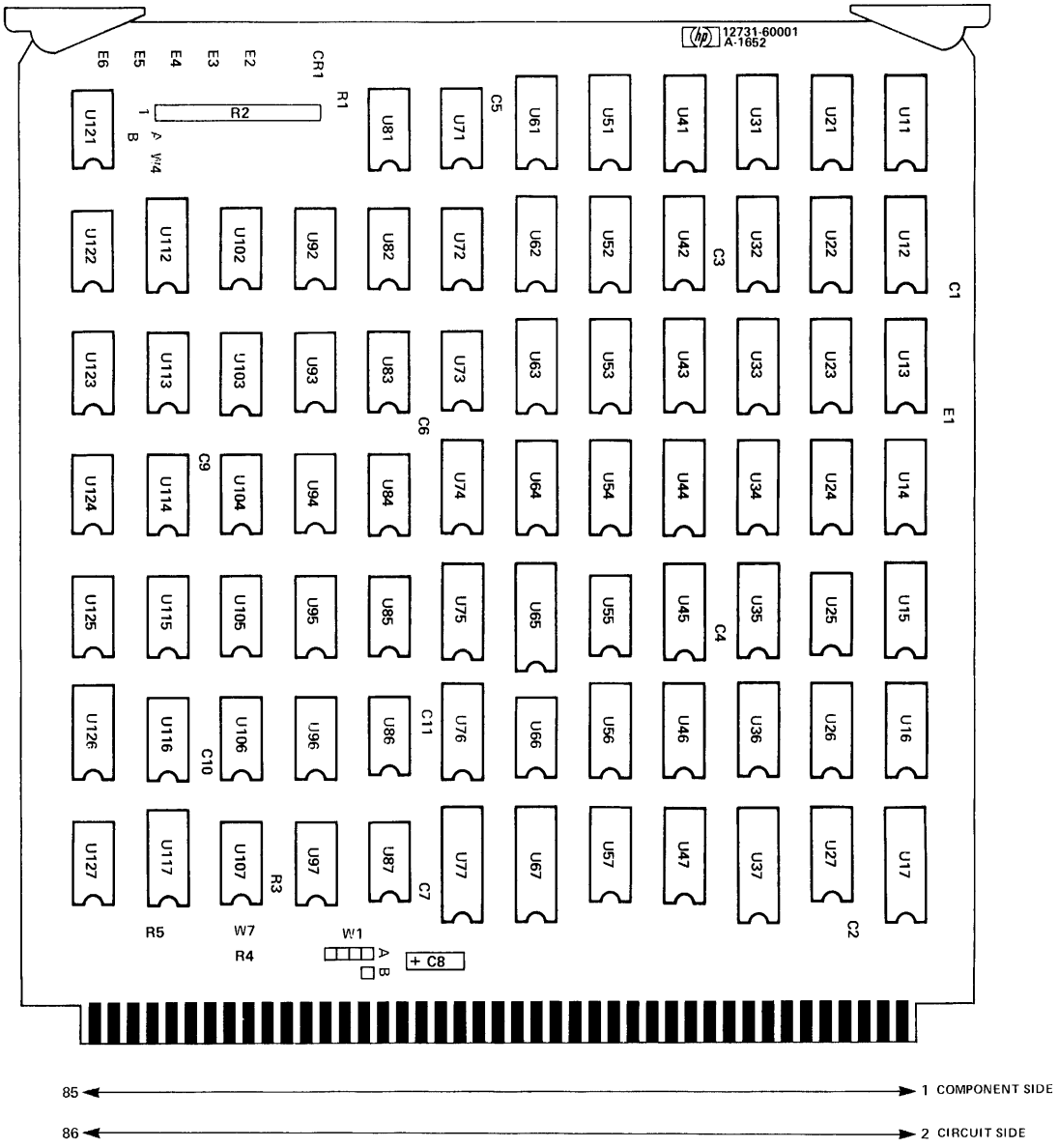
```
PAGE 0002 MICRO XREF  REV.2013 801031

SYMBOLS=0059  REFERENCES=0081  SOURCE LINES=0405

B.RESET   0280  0249  0250  0264  0265
BYTEADJ   0286  0246  0261
DJP       0213  0079
DJS       0229  0080
HORI      0029  **NOT REFERENCED**
INDIRECT  0028  0180  0186  0192  0377  0403
JMPSTAT   0222  0392  0395
JP*       0220  0214  0217
JRS       0383  0066
JS*       0232  0086  0227
JS*EXIT   0236  0381
JTABL     0053  **NOT REFERENCED**
LF*       0201  0076
MAPMOVE   0350  0337  0341
MB*       0270  0258
MBF       0245  0056
MBI       0246  0055
MBW       0260  0057
MELOOP1   0107  0114  0115
MELOOP2   0129  0135  0136
MELOOP3   0141  0147  0148
MELOOP4   0356  0360
MELOOP5   0363  0354  0367
MW*       0326  0301  0309  0315  0323
MWF       0298  0059
MWI       0299  0058
MWINT     0325  0311
```

```
MWW         0313  0060
ON.OFF      0389  **NOT REFERENCED**
OPGET       0397  0221  0232  0384  0388  0402
P.RTN       0119  0128  0134
PA*         0339  0063
PA.PB       0159  **NOT REFERENCED**
PB*         0343  0064
READMAP     0140  0106
RS*         0077  **NOT REFERENCED**
RTN*        0171  **NOT REFERENCED**
RV*         0078  **NOT REFERENCED**
SJP         0216  0081
SJS         0226  0082
SSM         0377  0065
SY*         0336  0061
SY.US       0162  0159
SY.USR      0391  0389
UJP         0219  0083
UJS         0085  **NOT REFERENCED**
US*         0347  0062  0344
W.LOOP      0316  0262  0324
X.LOOP      0302  0247  0310
XC*         0192  0075
XFER        0165  0162
XFERLOOP    0167  0170
XL*         0180  0073
XM*         0157  0071
XMM         0095  0053  0069  0070
```

```
XMM.EXIT  0116  0149
XMM.RTN   0118  0113  0146
XMS       0128  0105
XS*       0186  0074
```



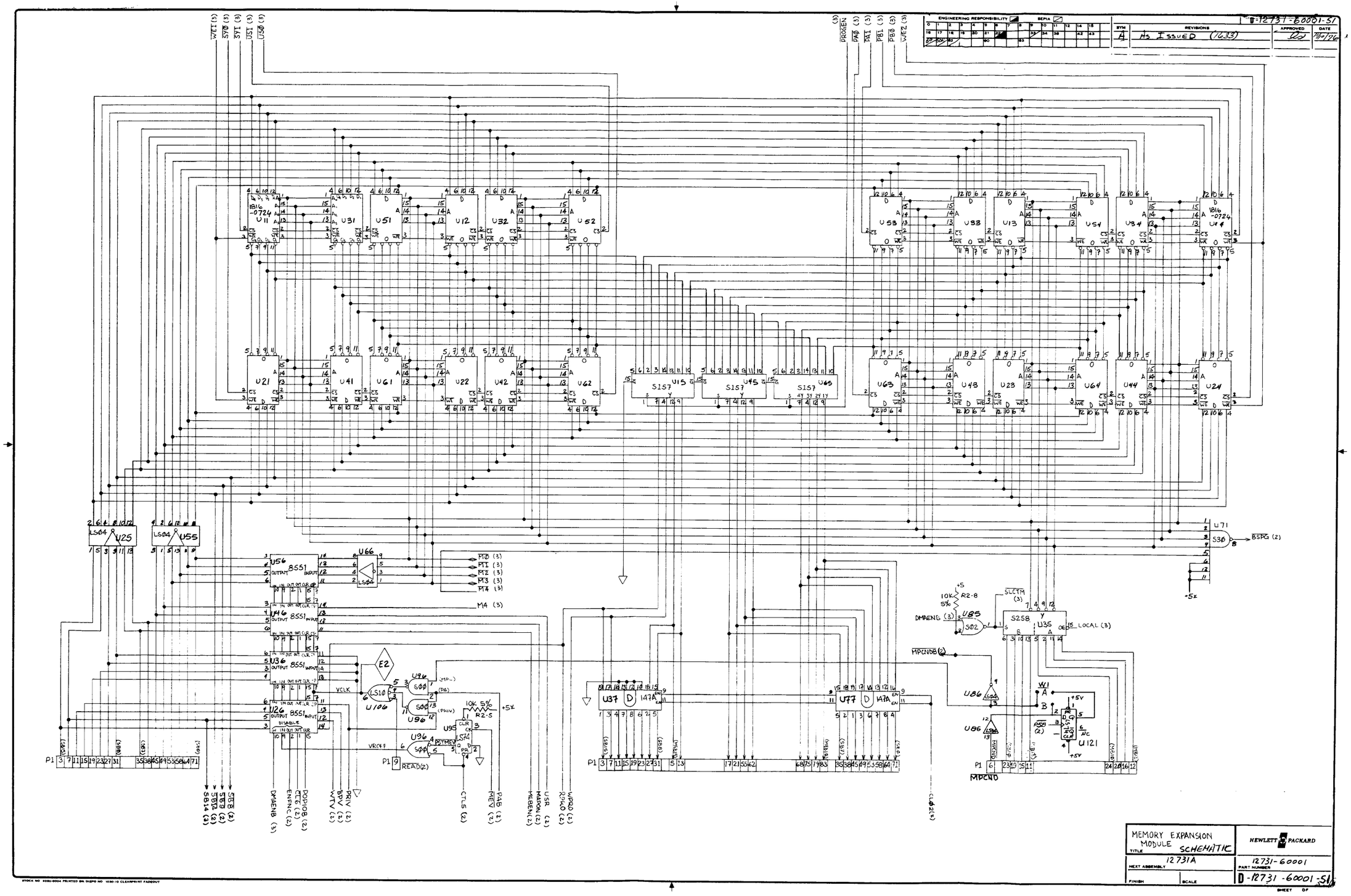
12731A Memory Expansion Module Assembly
12731-60001

ITEM NO	REFERENCE DESIGNATOR FIRST SIX	PART DESCRIPTION	PARENT OPTION	PART NUMBER	COMP OPTION	LOC	QUANTITY PER
01 03 05		R/L					
00	C1,3,5	CAP .47UF-20+80%		0160-0174		U	3
01	C2,4,6,7,9,10	CAP .01UF		0160-2055		U	6
00	C11	CAP 820PF 5%		0160-3539		U	1
00	C8	CAP 2.2UF 10%		0180-0197		U	1
00	E1-6	STUD SOLDER TERM		0360-0294		U	6
00	R5	RES 1000 5% .25		0683-1025		U	1
00	R4	RES 1500 5% .25		0683-1525		U	1
00	R1	RES 330 5% .25		0683-3315		U	1
00	R3	RES 470 5% .25		0683-4715		U	1
		SOCKET PC SINGLE		1251-1556		U	4
00	W1	JMPR PLUG .3"C-C		1258-0124		U	1
		PIN GRV .062X.25		1480-0116		U	2
00	R2	RES NET 8X10K		1810-0055		U	1
01 03	U11-14,21-24,31-34 41-44,51-54,61-64	IC RAM 16X4		1816-1005		U	24
00	U126	IC SN7423N		1820-0538		U	1
01	U26,36,46,56	IC DM8551N		1820-0574		U	4
01	U72,73,82,96,123	IC SN74S00N		1820-0681		U	5
		IC SN74S04N		1820-0683		U	3

PART NO CONT

ITEM NO.	REFERENCE DESIGNATOR (FIRST SIX)	PART DESCRIPTION	PARENT OPTION	PART NUMBER	COMP. OPTION	LOC	QUANTITY PER
		PART NO CONT		1820-0683			
01	U93,102,113						
		IC SN74S10N		1820-0685		U	4
01	U83,92,104,122						
		IC SN74S20N		1820-0688		U	2
00	U81,84						
		IC SN74S64N		1820-0691		U	2
01	U94,105						
		IC SN74S74N		1820-0693		U	1
00	U95						
		IC SN74S86N		1820-0694		U	1
00	U107						
		IC SN74161N		1820-0716		U	2
00	U75,76						
		IC HP147A		1820-0755		U	4
01	U17,67,37,77						
		IC SN74S157N		1820-1077		U	3
01	U15,45,65						
		IC SN74LS74N		1820-1112		U	3
01	U97,121,124						
		IC SN74LS02N		1820-1144		U	2
01	U116,125						
		IC SN74LS174N		1820-1196		U	2
01	U16,57						
		IC SN74LS00N		1820-1197		U	1
00	U114						
		IC SN74LS04N		1820-1199		U	4
01	U25,55,66,86						
		IC SN74LS10N		1820-1202		U	1
00	U106						
		IC SN74LS27N		1820-1206		U	1
00	U127						
		IC SN74S138N		1820-1240		U	1
00	U112						
		IC DM8098N		1820-1255		U	1
00	U74						
		IC SN74LS109N		1820-1282		U	1
				PART NO CONT			

ITEM NO	REFERENCE DESIGNATOR (FIRST SIX)	PART DESCRIPTION	PARENT OPTION	PART NUMBER	COMP OPTION	LOC	QUANTITY PER
		PART NO CONT		1820-1282			
00	U117						
		IC SN74S258N		1820-1309		U	1
00	U35						
		IC SN74S85N		1820-1321		U	2
00	U27,47						
		IC SN74S02N		1820-1322		U	3
01	U85,87,103						
		IC SN74S30N		1820-1323		U	1
00	U71						
		IC SN74S38N		1820-1451		U	1
00	U115						
		DIODE-LIGHT EMIT		1990-0327		U	1
00	CR1						
		LABEL-AL COLOR		7120-5480		L	1
		WIRE JUMPERS		8159-0005		U	2
00	W2,4						
		EXTRACTOR-PC		5040-6001		W	1
		EXTRACTOR-BLACK		5040-6068		W	1
		BOARD-ETCHED		12731-80001		W	1



8-12731-60001-51

ENGINEERING RESPONSIBILITY — SEMA

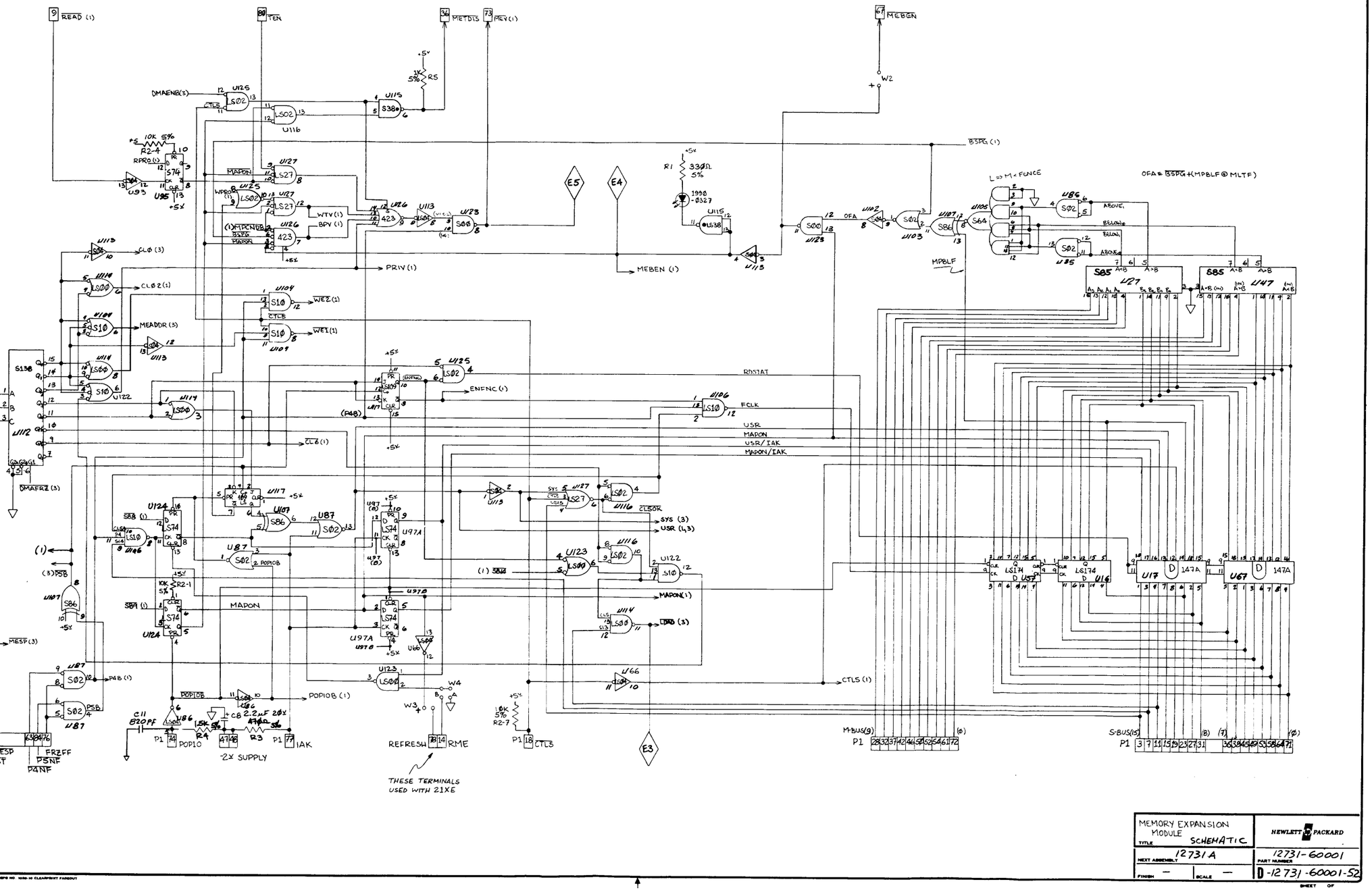
SYM	REVISIONS	APPROVED	DATE
A	H3 ISSUED (1633)	JCJ	11/1/76

TITLE: MEMORY EXPANSION MODULE SCHEMATIC	HEWLETT PACKARD
NEXT ASSEMBLY: 12731A	PART NUMBER: 12731-60001
FINISH	SCALE
	D-12731-60001-51
	SHEET OF

STOCK NO 5080-0004 PRINTED ON DIEPO NO 1030-10 CLEARPRINT FADEOUT

SYM	REVISIONS	APPROVED	DATE
A	As ISSUED (1633)		
B	ECO 22-3807, ADD C11, DATE CODE IS 1652		7-9-79
C	U122 WAS 74LS10 IN ERROR		5-3-79

ENGINEERING RESPONSIBILITY — SEPIA

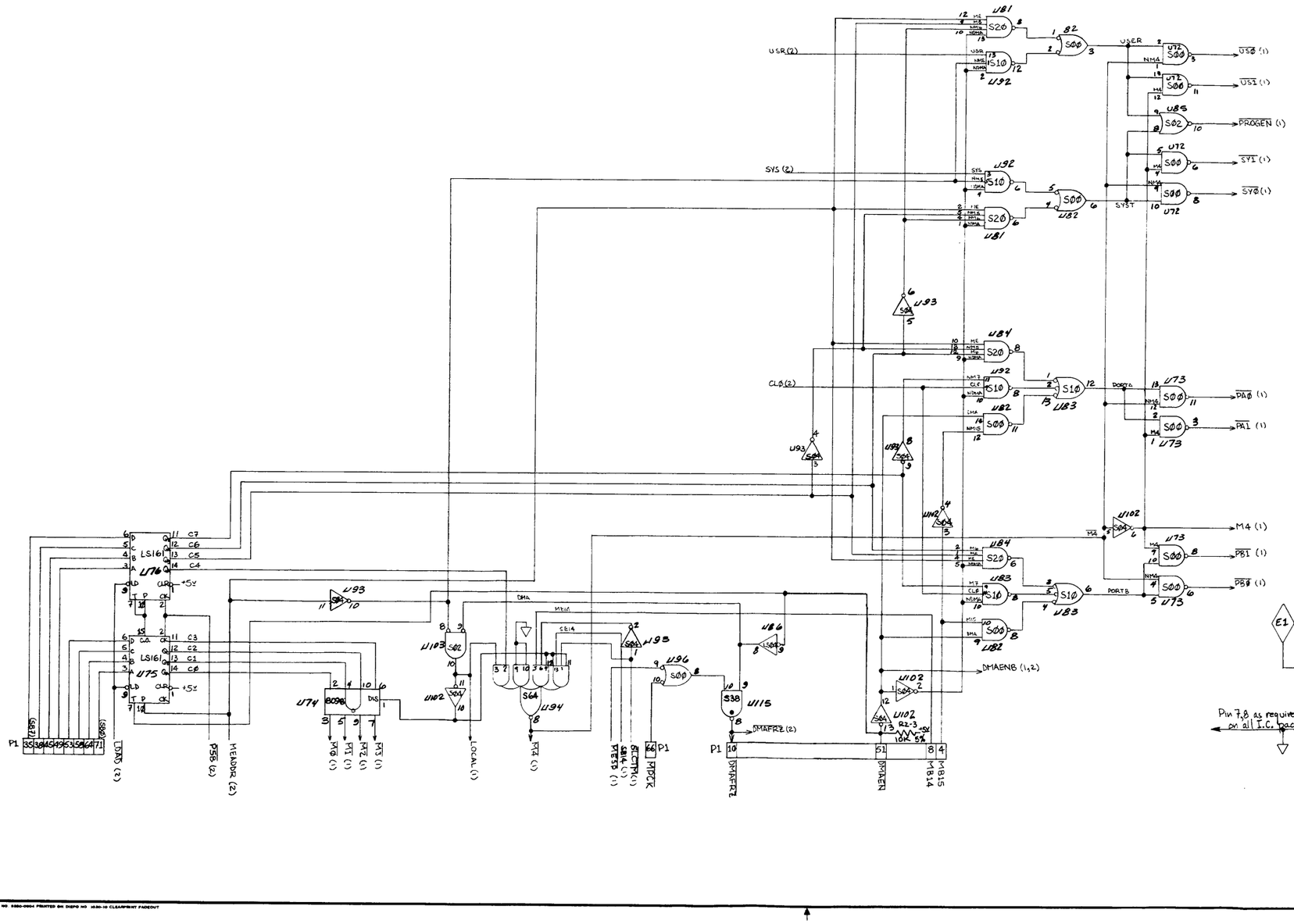


MEMORY EXPANSION MODULE SCHEMATIC	HEWLETT PACKARD
NEXT ASSEMBLY 12731A	PART NUMBER 12731-60001
FINISH — SCALE —	D-12731-60001-52
	SHEET OF

ENGINEERING RESPONSIBILITY	SEPIA

D-12731-60001-53

DTH	REVISIONS	APPROVED	DATE
A	AS ISSUED (1633)	[illegible]	11/01/76



JUMPER	21MX	21XE
W1	A	B
W2	IN	IN
W3	OUT	OUT
W4	A*	A*

* USE WITH 12892B MEMORY PROTECT RME JUMPER MUST CORRESPOND.

TITLE: MEMORY EXPANSION MODULE SCHEMATIC	HEWLETT PACKARD
NEXT ASSEMBLY: 12731A	PART NUMBER: 12731-60001
FINISH	SCALE	D-12731-60001-53
	SHEET OF

STOCK NO. 5060-0904 PRINTED ON DIEPO NO. 1030-10 CLEARPRINT FADEOUT