Microprogramming 21MX Computers

operating and reference manual

HEWLETT hp PACKARD

HEWLETT-PACKARD COMPANY
11000 WOLFE ROAD, CUPERTINO, CALIFORNIA, 95014

MANUAL PART NO. 02108-90008
MICROFICHE PART NO. 02108-90009

Printed: AUG 1974
Changed: OCT 1974
Printed in U.S.A.

LIST OF EFFECTIVE PAGES

Pages	Effective Date
Title	August 1974
ii to vii	August 1974
1-1	October 1974
1-2	August 1974
2-1 to 2-2	August 1974
2-3 to 2-4	October 1974
3-1 to 3-2	August 1974
3-3	October 1974
3-4	August 1974
3-5 to 3-6	October 1974
3-7 to 3-9	August 1974
3-10 to 3-12	October 1974
3-13	August 1974
3-14 to 3-15	October 1974
3-16 to 3-24	August 1974
4-1 to 4-3	August 1974
4-4	October 1974
4-5 to 4-6	August 1974
4-7	October 1974
4-8 to 4-11	August 1974
4-12	October 1974
4-13	August 1974
4-14	October 1974
4-15 to 4-19	August 1974
4-20	October 1974
4-21 to 4-23	August 1974
4-24	October 1974
4-25	August 1974
4-26	October 1974
5-1 to 5-2	August 1974
5-3	October 1974
5-4 to 5-6	August 1974
5-7 to 5-8	October 1974
5-9 to 5-17	August 1974
6-1	October 1974
6-2	August 1974
6-3	October 1974
6-4 to 6-7	August 1974
A-1 to A-3	August 1974
B-1	August 1974
C-1	August 1974
C-2	October 1974
D-1 to D-4	August 1974
E-1 to E-19	August 1974
I-1 to I-4	August 1974

HEWLETT hp PACKARD

2100 series computers

MANUAL UPDATE PACKAGE

Manual Identification

02108-90008 Dated: AUG 1974
Microprogramming 21MX Computers
Operating and Reference Manual

Update Package Identification

Update Package 1
Dated: OCT 1974

PURPOSE

This update package presents changes to the current edition of the manual. The package consists of this cover letter and all new and changed pages (back-up pages are provided when necessary).

NEW AND CHANGED PAGE FORMATS

Update packages have the month and year in which the package was produced printed at the bottom outside corner of each page (for example, *OCT 1974*).

Each page containing added, deleted, or changed material has a vertical bar printed in the outside margin adjacent to the modified area.

Any page with only the date (no vertical bar) indicates that the information contained on the page is new or entirely revised.

MANUAL UPDATING INSTRUCTIONS

To update this manual, replace existing pages with the new pages provided in this package. Add new pages sequentially where necessary.

allowed by Hewlett-Packard instruction set microprograms), the microprogram must test for pending interrupts or they can be lost. When a pending interrupt is detected, the microprogram must yield control to the interrupt handler. For a discussion of microprogram interrupt handling, refer to sections 3-32 and 3-33 in this manual.

1-4. SUMMARY

The advantages of microprogrammed control are:

a. The user can use a fully-supported general purpose computer to aid in the generation and debugging of extensions to the computer's own instruction set.

b. The user can speed up the overall execution time of his software by microprogramming its most time consuming or repetitious routines.

c. The user can implement enhancements of the instruction set and special purpose processors produced by the manufacturer with ,little impact on his existing software.

block diagram is the "roadmap" that is used to determine possible data paths and to determine where logical decisions can be made. This diagram can be unfolded and referred to while reading other parts of the manual. Note that the four sections of the computer, illustrated in figure 2-1, are shown in more detail in the functional block diagram.

To read the functional block diagram, begin with a 101rrr or 105rrr instruction in the Instruction Register. The rrr specifies the octal Control Store entry point address according to the description in section 3-24, Mapping to a Module Address. This address is moved into the ROM Address Register (RAR). With a first address specified, the user microprogram begins execution. The contents of the Control Store location given in the ROM Address Register are moved into the ROM Instruction Register (RIR). The ROM Instruction Register now holds a 24 bit micro-instruction. The micro-instruction is decoded and the specified control functions are executed.

Successive micro-instruction addresses are determined in the following way. The ROM Address Register is incremented at the start of execution of each micro-instruction. When a jump is executed, the ROM Address Register is loaded with the jump target address. When a jump to subroutine is executed, the ROM Address Register is stored into the SAVE Register (save return address) and the jump target address is stored into the ROM Address Register. When a return from subroutine is executed (RTN), the SAVE Register contents are transferred into the ROM Address Register and the SAVE Register is cleared. Thus at the completion of execution of each micro-instruction, the ROM Address Register holds the address of the next micro-instruction.

2-5. DATA PATHS

The central data transfer path is the S-bus. The contents of all regesters except the following can be directed onto the S-bus: L-register, RAR, SAVE Register, Extend Register, and the Overflow Register. The following registers can receive data from the S-bus:

M-register
T-register
L-register
Counter Register
Display Register
Display Indicator
Instruction Register

The T-bus receives data only from the Rotate/Shifter (R/S) but can pass data to these registers:

A-register
B-register
Scratch Pad Registers (S1 through S12)
X-register
Y-register
P-register
S-register

The I/O-bus serves to transfer data to and from external devices under programmed control.

Note in Appendix D, the functional block diagram, that the arrows are significant. For example, the B-register contents can be sent to the S-bus and thence to the M-register. However, the contents of the B-register cannot be sent to S12 (Scratch Pad 12) without passing through the ALU.

2-6. MAIN MEMORY

The M-register is a 15 bit register which holds memory addresses for reading from or writing into Main Memory. When storing from the M-register, bit 15 is clear (0). The T-register or Transfer register holds the data being transferred to or from memory. Contents of both these registers are transferred to and from the S-bus. Four loader ROMs, selectable by Instruction Register bits 15 and 14, each can contain a 64 word Main Memory program which may be loaded into Main Memory and used to load Main Memory from a peripheral device or to perform any other function desired by the user.

Two flags are associated with memory: the A-register Addressable Flag (AAF) and the B-register Addressable Flag (BAF). These flags are required to allow the A- and B-registers to be addressed as locations 0 and 1, respectively, of Main Memory.

2-7. I/O SECTION

The Central Interrupt Register (CIR) is a 6 bit register associated with the I/O interrupt circuitry. It is loaded with the Select Code of the interrupting device under program control and passed to the S-bus. Whenever the Central Interrupt Register is loaded, an Interrupt Acknowledge (IAK) signal is issued to the I/O device.

The I/O-bus transfers data to and from external devices.

Two flags are associated with I/O: the Interrupt Pending flag and the I/O Skip Condition Met (Main Memory instructions SFS and SFC) flag.

The Interrupt Enable Register is used to disable or enable the recognition of all interrupts, except Memory Protect, Parity, and Power Fail interrupts.

- Field 5 begins in column 25 and contains a micro-order no longer than four characters.

- Field 6 begins in column 30 and contains a micro-order no longer than four characters (Word Type 1) or an operand (Word Types 2, 3, and 4).

- Field 7 begins in column 40 and contains comments only; comments may begin and be placed anywhere from column 40 to column 80 (if column 39 contains the last character of the field 6 operand, field 7 must begin in column 41).

3-10. CHARACTER SET

The characters that may appear in a source statement are as follows:

A through Z
0 through 9
. (period)
* (asterisk)
+ (plus)
- (minus)
(space)

Any ASCII character may appear in the comments field.

A space may only begin a field if no micro-order is specified in that field.

3-11. LABEL SYMBOL

A label may be one to eight characters consisting of A through Z, 0 through 9, and a period. The first character must be a letter.

Each label must be unique within the microprogram. Names which appear in $EXTERNALS micro-assembler control input statements (refer to section 5-5) may not be used as statement labels in the same microprogram.

.3-12. ASTERISK COMMENT

An asterisk in column one of the source statement indicates that the entire micro-assembler source statement is a comment.

3-13. MICRO-ORDERS: FIELDS 2 THROUGH 6

The micro-order fields define operations that are to be performed by the Control Section of the computer. The micro-orders applicable to each field are determined by the source statement Word Type. Section IV describes the micro-orders that apply to each Word Type and describes the operations that they specify.

3-14. OPERANDS IN FIELD 6

Word Types 2, 3, and 4 contain an operand in field 6.

In Word Type 2, the operand must be either a decimal or octal number. It cannot be an expression (refer to section 4-10 for definition of a Word type 2 operand).

In Word Types 3 and 4, the operand is a decimal number, octal number, or a number computed from an expression which can include a label (refer to section 4-16 for the definition of a Word Type 3 operand. Refer to section 4-20 for the definition of a Word Type 4 operand).

3-15. CODING THE FOUR WORD TYPES

The following sections describe how to code source statements in micro-assembly language. The reader should be familiar with Section IV of this manual before proceeding with these descriptions. Section IV describes the micro-orders that can be used with each Word Type. By referring to Section IV, the reader can see the options that are available to him as each Word Type is described. The reader will also need to refer to the functional block diagram in Appendix D.

3-16. CODING WITH WORD TYPE 1 — COMMON

This word type specifies data transfer and modification. The format of Word Type 1 is shown in section 4-1. As an example, a micro-instruction is developed that executes the following control functions:

- Store the A-register contents into the M-register

- Perform a memory protect check on the A-register contents

- Transfer the A-register contents to the ALU, increment this value in the ALU, and store the result into the P-register

a. Specify the register that is to be placed on the S-bus; the A-register is specified in the example:

OP	SPEC	ALU	STORE	S-BUS
				A

b. Specify the function of the ALU; the increment function is specified in the example:

OP	SPEC	ALU	STORE	S-BUS
		INC		A

c. Specify the Op field function; no Op field function is specified in the example. When no Op function is required, the standard operation is specified by either leaving the field blank or inserting NOP into the field:

OP	SPEC	ALU	STORE	S-BUS
NOP		INC		A

d. Specify a Special function, if required; a memory protect check is specified in the example:

OP	SPEC	ALU	STORE	S-BUS
NOP	MPCK	INC		A

e. Finally, specify where the resulting data is to be stored. Two store operations are required in the example. The unmodified A-register value on the S-bus must be stored into the M-register and the incremented A-register value on the T-bus must be stored into the P-register. The micro-order PNM performs both of these store operations and serves to illustrate that data stored from the S-bus is unmodified data and data stored from the T-bus can be modified by the ALU or R/S:

OP	SPEC	ALU	STORE	S-BUS
NOP	MPCK	INC	PNM	A

PNM is a unique micro-order. No other micro-order provides the ability to store into two registers in the same micro-instruction.

3-17. CODING WITH WORD TYPE 2 — IMMEDIATE DATA

This word type sends an 8 bit constant (immediate data) specified in the micro-instruction to a register. The format of Word Type 2 is shown in section 4-7. As an example, a micro-instruction is developed that specifies the following control function:

- Repeat the micro-instruction following this one ten times

a. Specify IMM in the Op Code field:

"IMM"	SPEC	MODIF	STORE	OPERAND
IMM				

b. Specify the octal or decimal data to be placed on the S-bus; an octal -12 is specified in the example(366B):

"IMM"	SPEC	MODIF	STORE	OPERAND
IMM				366B

This is necessary because use of the minus sign (−) is not allowed.

c. Specify one of the four possible data modifiers (refer to section 4-9); LOW (place the 8 bit operand in the lower half of the S-bus and ones in the upper half) is specified in the example:

"IMM"	SPEC	MODIF	STORE	OPERAND
IMM		LOW		366B

d. Specify where the resulting data is to be stored; the Counter Register is specified in the example:

"IMM"	SPEC	MODIF	STORE	OPERAND
IMM		LOW	CNTR	366B

e. Specify any special operations required; RPT (repeat the micro-instruction following this one the number of times specified in the Counter Register) is specified in the example:

"IMM"	SPEC	MODIF	STORE	OPERAND
IMM	RPT	LOW	CNTR	366B

3-18. CODING WITH WORD TYPE 3 — CONDITIONAL JUMP

This word type specifies a conditional branch in the microprogram. The format of Word Type 3 is shown in section 4-11. As an example, a micro-instruction is developed that specifies the following control function:

- Jump to the microprogram address labeled ERR2, if the last data on the T-bus was not zero.

a. Specify JMP and CNDX in the Op Code and Special fields:

"JMP"	"CNDX"	COND	JUMP SENSE	OPERAND
JMP	CNDX			

3-23. CONTROL STORE MODULES AVAILABLE TO USER

The 4096 words of ROM are divided into sixteen 256-word modules, module 0 through module 15. Modules 0, 1, 14, and 15 hold the 21MX Instruction Set and are not available to the user microprogrammer. Modules 12 and 13 are reserved exclusively for user microprograms. Any other Control Store space, not filled by a microprogrammed option, is available to the user microprogrammer. Figure 3-4 summarizes the allocation of Control Store.

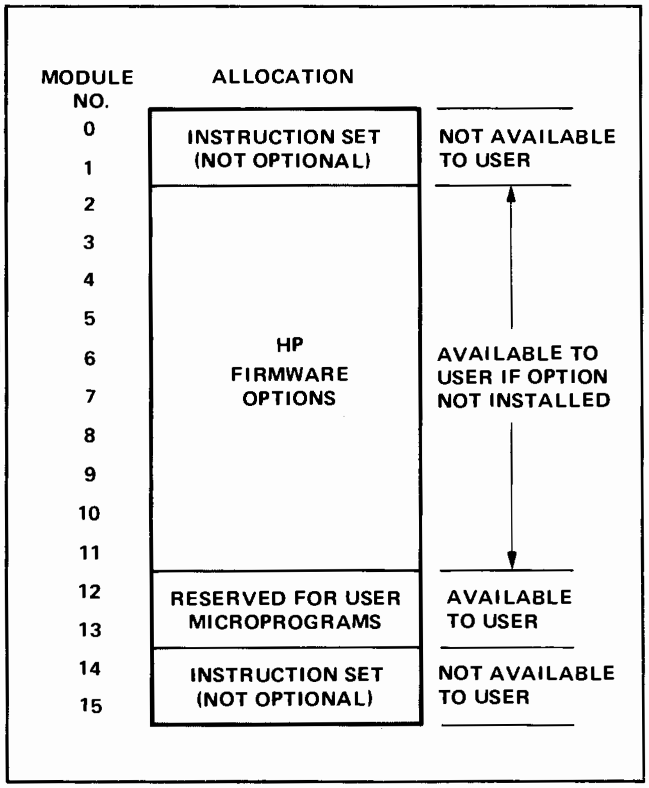

MODULE NO.
ALLOCATION
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
INSTRUCTION SET (NOT OPTIONAL)
NOT AVAILABLE TO USER
HP FIRMWARE OPTIONS
AVAILABLE TO USER IF OPTION NOT INSTALLED
RESERVED FOR USER MICROPROGRAMS
AVAILABLE TO USER
INSTRUCTION SET (NOT OPTIONAL)
NOT AVAILABLE TO USER

Figure 3-4. Allocation of Control Store by Modules

3-24. MAPPING TO A MODULE ADDRESS

Function codes available to the user are listed in table 3-1 together with the module address to which these function codes map. Some of these user function codes are assigned to the microprogrammed processors and options produced by Hewlett-Packard. The following function codes cannot be used:

105000 through 105137

105740 through 105777

101740 through 101777

If the HP 12977A Fast FORTRAN Processor is installed, the following function codes are not available to the user:

105140 through 105277

105700 through 105737

101700 through 101737

Note: If the function code maps to a Control Store module which is not present, the micro-instruction

JEAU PASS S S

is executed for each non-existent Control Store location. The ROM Address Register is incremented after each execution of the above micro-instruction until an installed module is encountered. No notification is given to the user or system that a non-existent module is being executed.

Table 3-1. User Function Code Mapping

Function codes $101rrr_8$ and $105rrr_8$ map to the module address given:

	RANGE OF rrr VALUES	MODULE	RANGE OF OCTAL ADDRESSES
$105rrr_8$ only	140 to 157	3	1400
	160 to 177	3	1400 to 1417
	200 to 217	4	2000
	220 to 237	4	2000 to 2017
	240 to 257	5	2400
	260 to 277	5	2400 to 2417
	300 to 317	6	3000
	320 to 337	6	3000 to 3017
	340 to 357	7	3400
	360 to 377	7	3400 to 3417
$101rrr_8$ or $105rrr_8$	400 to 417	8	4000
	420 to 437	8	4000 to 4017
	440 to 457	9	4400
	460 to 477	9	4400 to 4417
	500 to 517	10	5000
	520 to 537	10	5000 to 5017
	540 to 557	11	5400
	560 to 577	11	5400 to 5417
	600 to 617	12	6000
	620 to 637	12	6000 to 6017
	640 to 657	13	6400
	660 to 677	13	6400 to 6417
	700 to 717	2	1000
	720 to 737	2	1000 to 1017

3-25. MICROPROGRAMMING INPUT AND OUTPUT FUNCTIONS

Microprogramming Input and Output (I/O) functions requires more care than any other type of microprogramming, because there are strict timing dependencies. The microprogram described in section 3-40 is an example of I/O microprogramming.

To maintain integrity of the I/O system, every control signal which goes to the I/O devices is generated in a specific time period (T-period). All micro-instructions, except those containing READ or WRTE micro-orders, are executed in one I/O T-period, where T = 325 ns. READ and WRTE each require two I/O T-periods. An I/O time cycle consists of five T-periods labelled T2, T3, T4, T5, and T6. Specific I/O activity is restricted to certain T-periods in order to synchronize setting of data flags, latching of data, and resolving of multiple interrupt requests.

The microprocessor must synchronize with T2 before initiating an I/O cycle. Thereafter, special consideration must be given to the order and timing of the I/O micro-instructions given.

3-26. SYNCHRONIZING WITH THE I/O SYSTEM

To initiate an I/O cycle, the IOG micro-order must be specified. When this occurs, the processor "freezes" (ceases executing micro-instructions) until time T2. The next micro-instruction is executed during time T3, the next during T4, etc. IOG may occur with any micro-instruction which does not require some other Special or Jump Modifier (Field 3) micro-order.

Examples:

```
a. READ  IOG  INC   PNM  P
b.       IOG  PASS  IR   S3
```

3-27. I/O SIGNAL GENERATION

When IOG is specified, the I/O system generates control signals to the I/O devices starting at the next T2 time and according to the contents of the Instruction Register (IR).

IR bits 5-0 hold a Select Code (SC) signal (SC = the I/O slot number on the backplane or in I/O extenders) that determines which device will respond to the control signal. IR bits 11-6 determine which I/O signals are sent, as shown in table 3-2. The IR must be loaded prior to or during occurrence of the IOG to insure that the correct signals are generated to the proper SC. If Memory Protect is enabled, the IR must be loaded prior to issuing IOG (see section 3.34).

Select Codes 0, 1, 2, 3, 4, and 5 have special functions concerning, respectively, the interrupt system, the Front Panel, the Dual Channel Port Controller (DCPC), Power Fail, and Memory Protect/parity. The "Interrupt and Control summary" table in the Appendix of the **HP 21MX Computer Series Reference** manual (HP 02108-90002) holds a description of the effect of these select codes (S.C. in the table).

Table 3-2. I/O Control Signal Generation Determined by IR Bits 11-6

		IR*						
11	10	9	8	7	6	I/O SIGNAL	TIME	GENERAL USE
x	x	x	0	0	0	none	T3	Turns off the Run Flag on the CPU.
x	x	x	0	0	1	STF	T3	Set device flag.
x	x	1	x	x	x	CLF	T4	Clear device flag.
x	x	x	0	1	0	SFC	T3-T5	SKPF condition is true if and only if the device flag is clear.
x	x	x	0	1	1	SFS	T3-T5	SKPF condition is true if and only if the device flag is set.
–	–	–	–	–	–	IOI	T4-T5	Buffer the input data latch on the device onto the I/O-bus; this command must be stated explicitly in micro-code during these times.
x	x	x	1	1	0	IOO	T3-T4	Store the I/O-bus into the input data latch on the device.
0	x	x	1	1	1	STC	T4	Set device control flag.
1	x	x	1	1	1	CLC	T4	Clear device control flag.

*Bits marked with x are not significant for the I/O signal specified.

3-35. THE EFFECT OF THE DUAL CHANNEL PORT CONTROLLER ON MICROPROGRAMS

The Dual Channel Port Controller (optional hardware) steals full I/O cycles to perform direct transfers between external devices and Main Memory. This process is essentially transparent to the microprogram. The Dual Channel Port Controller (DCPC) is a hardware function that does not employ microcode. If the microprogram interferes with a DCPC cycle, the Control Processor freezes until DCPC completes its cycle. If DCPC takes a sequence of consecutive I/O cycles for input transfers, any attempted IOG, READ, or WRTE micro-orders will freeze the processor until DCPC is finished. If DCPC takes a sequence of consecutive I/O cycles for output transfers, the Memory Reference Group, the Alter/skip Group, and Shift Rotate Group macro-instructions can still proceed at between 40% and 60% normal execution rate; IOG will still freeze the Control Processor.

If DCPC takes as much as 50% of all I/O cycles, the overall efficiency of the basic instruction set execution is 60% to 70% for input or output transfers. Non-main Memory micro-instruction execution is only frozen 20% of each DCPC cycle. Thus arithmetic and logical micro-instructions execute at 80% efficiency, when DCPC takes every I/O cycle.

3-36. SUMMARY OF SPECIAL TIMING RULES

a. Always load the M-register before specifying WRTE in the OP micro-order field.

b. Load the M-register before or during micro-instructions containing READ in the OP field. Do not modify M-register until two micro-instructions after the READ.

c. Do not alter the T-register unless initiating a WRTE, since the T-register is internal to the Main Memory system and is used by DCPC and the CPU. The T-register is not intended to be a general purpose register, but to be used in referencing Main Memory.

d. Load the T-register with data to be written in the same instruction as WRTE appears, or DCPC could alter it before WRTE is executed.

e. The T-register must be placed on the S-bus no later than two micro-instructions after a READ is specified or the T-register will be disabled by the Memory system.

f. When an I/O cycle (using IOG) is in progress, a READ or WRTE **must not be initiated** before T6 in the cycle under either of the following conditions:

   1. An input or output routine (refer to sections 3-29 and 3-30) is in progress.

   2. A skip flag test of the I/O system is taking place.

g. Do not specify a READ or WRTE micro-order in the same micro-instruction that is transferring data from the T-register (T or TAB micro-order in the S-bus field). The reason is that if a freeze occurs as a result of such a READ or WRTE micro-order (see i. below) the data in the T-register will be invalid after the freeze.

   For example, a sequence of micro-instructions similar to the following must not take place:

```
READ   —    INC    PNM   P
—      —    PASS   S4    L
READ   —    INC    M     TAB
```

h. Do not start an I/O cycle (using IOG) before data is transferred from the T-register following a READ operation. The reason is that if the IOG results in a freeze (see i. below), the data in the T-register will be invalid.

   For example, a sequence of micro-instructions similar to the following must not take place:

```
READ   —    INC    PNM   P
—      IOG  PASS   S4    TAB
```

i. The following conditions always cause a micro-processor freeze:

   1. The CIR micro-order is in the S-bus field and either the I/O cycle time is not T6 or the Dual Channel Port Controller is stealing a full I/O cycle.

   2. The IOG micro-order is in the Special field and either the I/O cycle time is not T2 or the Dual Channel Port Controller is stealing a full I/O cycle.

   3. A T or TAB micro-order is in the S-bus field and a READ or WRTE micro-order memory cycle is still in progress.

   4. A READ or WRTE micro-order is in the Op field and one of the following conditions is true:

      (a) The semi-conductor Main Memory is being refreshed (two micro-instruction cycles are required every 32.5 microseconds for this purpose).

      (b) The Dual Channel Port Controller is stealing an I/O cycle and has not completed its memory reference.

      (c) A READ or WRTE memory cycle is still in progress.

j. Load the IR before issuing IOG unless there is no chance that Memory Protect is enabled (no Memory Protect on 2105).

3-37. SAMPLE MICROPROGRAMS

While reading the sample microprograms, the reader may find it useful to refer to the fold out functional block diagram in Appendix D. This diagram and the micro-order definitions in Section IV are the two basic sets of information used by the programmer in writing a microprogram.

3-38. SWAP MEMORY LOCATIONS

The sample microprogram illustrated in figure 3-5 swaps the contents of two Main Memory locations that are pointed to by the A- and B-registers (no indirect addresses).

Micro-instruction Commentary

```
READ          INC    M    A
```

a. Put the address in the A-register onto the S-bus.

b. Store the S-bus into the M-register.

c. Pass the S-bus through the ALU and increment data enabling the A- or B-register addressable test.

d. Read the location in Main Memory pointed to by the M-register (this requires 2 micro-instruction cycles).

```
      MPCK  PASS         M
```

a. Put the M-register onto the S-bus.

b. Pass the S-bus through the ALU (output not used).

c. Since READ requires two cycles, an instruction cycle is available before data is available from memory. And since the M-register holds the address of the location that will eventually be written into, this cycle is used for the memory protect check.

```
            PASS   S1    TAB
```

a. The read is complete and data from the memory location is in the T-register unless the AAF or BAF Flag is set. If AAF is set, the data is in the A-register. If BAF is set, the data is in the B-register.

b. Put memory data on the S-bus.

c. Pass S-bus through the ALU and R/S to the T-bus.

d. Store data on T-bus into Scratch Pad Register 1 (S1).

```
READ          INC    M    B
```

a. Put the address in the B-register onto the S-bus.

b. Store S-bus into the M-register.

c. Pass the S-bus through the ALU and increment data enabling the A- or B-register addressable test.

d. Read the Main Memory location pointed to by the M-register.

```
      MPCK  PASS         M
```

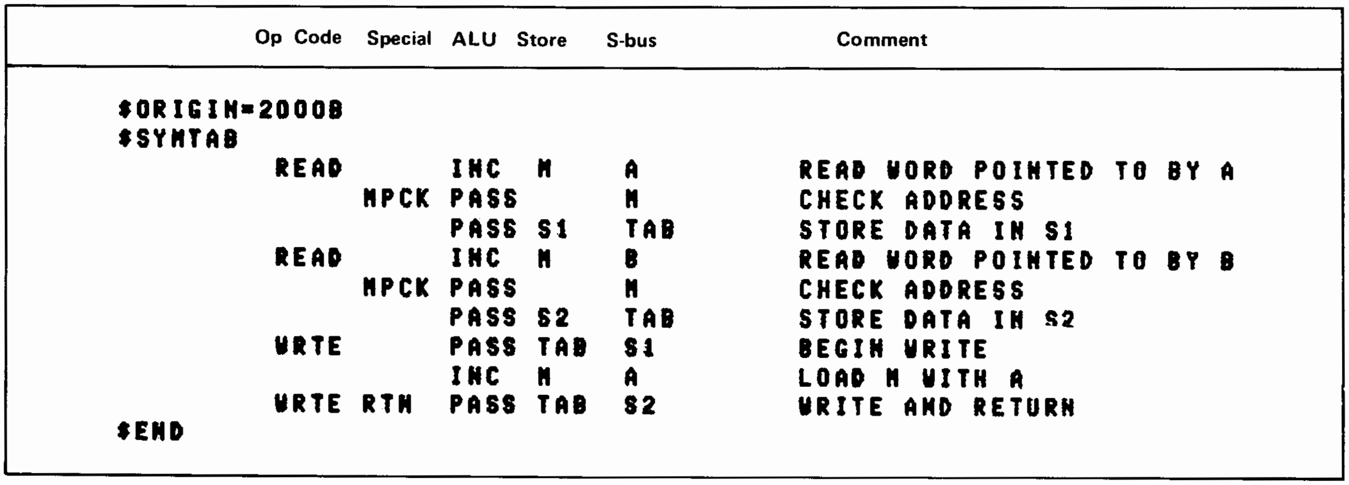
a. Put M-register (memory address) onto the S-bus.

b. Pass the S-bus data through the ALU.

c. Test the address for a Memory Protect violation.

```
            PASS   S2    TAB
```

a. Put memory data (T-, A-, or B-register contents) onto the S-bus.



```
                Op Code  Special  ALU   Store   S-bus          Comment

$ORIGIN=2000B
$SYMTAB
           READ          INC    M      A           READ WORD POINTED TO BY A
                  MPCK   PASS          M           CHECK ADDRESS
                         PASS   S1     TAB         STORE DATA IN S1
           READ          INC    M      B           READ WORD POINTED TO BY B
                  MPCK   PASS          M           CHECK ADDRESS
                         PASS   S2     TAB         STORE DATA IN S2
           WRTE          PASS   TAB    S1          BEGIN WRITE
                         INC    M      A           LOAD M WITH A
           WRTE   RTN    PASS   TAB    S2          WRITE AND RETURN
$END
```

Figure 3-5. Swap Microprogram

OP
DIV

Required micro-instruction mnemonic fields:

OP	SPECIAL	ALU	STORE	S-BUS
DIV	L1	SUB	B	B

Equivalent micro-instruction binary fields:

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP				ALU					S-BUS					STORE					SPECIAL				
0	1	0	1	0	0	1	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0

Meaning: Perform a divide step where the divisor is in the L-register and the 32 bit dividend is in the A- and B-registers (least significant bits in the A-register). This micro-order is repeated (16 times for a full word divisor) by specifying the Special micro-order RPT in the preceding micro-instruction. This performs the successive subtractions required in a divide algorithm.

The divide step is executed as follows:

a. Subtract the L-register from the B-register (ALU = B -L).

b. If borrow is required to complete the subtraction, the ALU Carry Out Flag is clear (0). This Carry Out result means that the divisor (L-register) is too big. The ALU result is not stored. The A-register and B-register are left shifted one bit and the divide step is complete.

c. If a borrow is not required to complete the subtraction, the ALU Carry Out Flag is set (1). This Carry Out result means that the divisor is small enough. The result of the subtraction is contained in the ALU and is left shifted one bit and stored back into the B-register. Bit 15 of the A-register shifts into bit 0 of the B-register and bit 0 of the A-register is set to 1 (the Carry Out result). The divide step is complete.

Usage: The base set divide operation is shown in the Basic Instruction Set microprogram in Appendix E at the label = DIV.

Initial Contents:

B-register		A-register	L-register
Dividend 16 Most Significant bits	. . .	Dividend 16 Least Significant bits	Divisor

After Repeat 16 Times of Divide Step:

Remainder Doubled	16 Bit Quotient of (B,A)/L	Divisor (unchanged)

OP
ENV

BIT NO.	23	22	21	20
CONTENT	1	0	1	0

Meaning: Enable the overflow test for the current ALU operation.

Usage: To detect an overflow condition (that is, set the Overflow register bit), ENV or ENVE (see below) must be specified as the OP Code of the micro-instruction in which the condition is to be tested. Overflow is set if the S-bus and L-register bits 15 are the same and bit 15 output from the ALU is different.

Caution: Caution is advised in the use of DEC (decrement) or INC (increment) in conjunction with ENV. The L-register is always compared.

OP
ENVE

BIT NO.	23	22	21	20
CONTENT	1	0	1	1

Meaning: Enable the overflow test and the extend test for the current ALU operation.

Usage: To detect an Overflow condition (that is, set the Overflow register bit), ENV (see above) or ENVE must be specified as the OP Code of the micro-instruction. To set the Extend Register as a result of the ALU operation, the ENVE micro-order must be specified as the OP code of the micro-instruction. The Extend Register bit is set if there is a carry generated by the ALU (ALU Carry Out = 1).

OP
LGS

Required micro-instruction mnemonic fields:

OP	SPECIAL	ALU	STORE	S-BUS
LGS	L1 or R1	PASS	B	B

OP	BIT NO.	23	22	21	20
NOP	CONTENT	0	0	0	0

Meaning: Standard Operation. No operation is specified for the Op Code field.

Usage: This is the default micro-order when the OP Code Field is left blank.

OP	BIT NO.	23	22	21	20
WRTE	CONTENT	0	1	1	1

Meaning: Write data from the T-register into the Main Memory address specified in the M-register. The CPU will freeze until Main Memory is not busy. Two micro-instruction times are required to complete the write.

Usage: The T-register should be loaded during the write instruction and must not be altered by the next sequential micro-instruction; otherwise the Dual Channel Port Controller data-transfers could destroy the data.

4-3. SPECIAL MICRO-ORDERS

SPECIAL	BIT NO.	4	3	2	1	0
CLFL	CONTENT	0	1	0	0	1

Meaning: Clear the CPU Flag.

SPECIAL	BIT NO.	4	3	2	1	0
COV	CONTENT	0	1	1	0	0

Meaning: Clear the Overflow Register bit.

SPECIAL	BIT NO.	4	3	2	1	0
FTCH	CONTENT	0	1	0	1	0

Meaning: Move the Main Memory address contained in the M-register (usually the address of the next macro-instruction to be executed) to the Memory Protect Violation Register. Clear out the Memory Protect Violation flag and reset the Indirect Counter.

Usage: This micro-order must be used during, or one micro-instruction after, the initiation of a READ from the address of the next macro-instruction to be executed. This micro-order must be used if the Memory Protect feature is installed on the computer.

SPECIAL	BIT NO.	4	3	2	1	0
ICNT	CONTENT	1	0	0	1	1

Meaning: Increment the Counter Register by one.

SPECIAL	BIT NO.	4	3	2	1	0
INCI	CONTENT	1	0	1	0	1

Meaning: Increment the Indirect Counter in the Memory Protect Option (if installed) by one.

Usage: Used by microprograms that implement indirect addressing. If INCI is executed three times within the same microprogram, the Interrupt Enable Flag is set to allow the CPU to recognize interrupts. Used to prevent multiple indirect addressing levels from holding off recognition of I/O interrupt requests.

SPECIAL	BIT NO.	4	3	2	1	0
IOFF	CONTENT	0	0	0	0	0

Meaning: Turn off the Interrupt Enable flag to disable recognition of normal interrupts (does not disable memory protect, parity, or power fail interrupts).

Usage: After three occurrences of INCI (see INCI Usage) in the SPECIAL Field, interrupts are again recognized and cannot be disabled until a FTCH micro-order occurs. The ION micro-order is normally used to re-enable interrupt recognition.

IOFF should be used with caution, since holding off interrupts could cause the loss of input and output data.

SPECIAL	BIT NO.	4	3	2	1	0
IOG	CONTENT	1	0	0	1	0

Meaning: Freeze the CPU until time period T2. Then execute the base set I/O macro-instruction that is in the Instruction Register.

Usage: Microprogrammed input and output require cooperation between the I/O Section and microprogram control. Familiarity with the I/O system is mandatory. See section 3-25 and the following sections for a more detailed description of I/O microprogramming.

SPECIAL	BIT NO.	4	3	2	1	0
ION	CONTENT	0	0	1	0	1

Meaning: Turn the Interrupt Enable flag on to enable recognition of interrupts. Allow the CPU to recognize standard device interrupts until the micro-order IOFF is executed.

Usage: After ION has been executed, the CPU can detect an interrupt from any I/O device in two ways:

a. If a JMP or RTN to location 0 of Control Store (the macro-instruction read and decode routine) is executed and an interrupt is pending or the Run flag is clear, execution is forced to location 4 in Control Store, which is the interrupt handler routine.

b. A test for interrupt pending or Run flag clear can be performed by the executing microprogram by executing INT, NHOI, or RUN in the Jump Condition field.

ION allows interrupts to be recognized. However interrupts are not generated by the interrupt system until a STF 0 I/O control command is executed. Refer to the discussion of the interrupt system in the **HP 21MX Computer Series Reference Manual.**

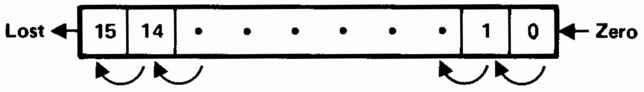
SPECIAL	BIT NO.	4	3	2	1	0
JTAB	CONTENT	1	1	0	1	1

Meaning: Perform a jump to a location within the Basic Instruction Set microprogram, based on the eight most significant bits (bits 15 through 8) of the Instruction Register. This is accomplished via a table look-up of the address in the main jump table for the basic instruction set (see figure 3-2).

The Save Register is cleared to 0. JTAB overrides the effects of JMP or JSB in the OP code field.

SPECIAL	BIT NO.	4	3	2	1	0
L1	CONTENT	0	0	0	1	0

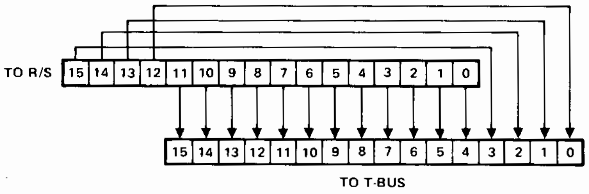
Meaning: Left one bit command to the Rotate/Shifter.



Usage: See MPY, DIV, CRS, LGS, ARS, LWF. Without one of the previous Op Codes, L1 performs a one bit logical left shift on data leaving the ALU.

SPECIAL	BIT NO.	4	3	2	1	0
L4	CONTENT	0	0	0	1	1

Meaning: Four bit circular left shift command to the Rotate/Shifter (R/S).



Usage: Used in conjunction with the shift and rotate operations.

SPECIAL	BIT NO.	4	3	2	1	0
MPCK	CONTENT	1	0	0	0	1

Meaning: Check the address placed on the S-bus for a memory protect violation.

Usage: An S-BUS micro-order must be used in conjunction with MPCK.

This check should be performed before any write to Main Memory (WRTE OP-code), if the memory protect feature is installed. Refer to section 3-27 for details on use of MPCK with the I/O system.

SPECIAL	BIT NO.	4	3	2	1	0
NOP	CONTENT	0	0	1	1	1

Meaning: No SPECIAL operation is performed.

Usage: This is the default operation if none is specified in the SPECIAL field.

SPECIAL	BIT NO.	4	3	2	1	0
RPT	CONTENT	0	1	1	0	1

Meaning: Repeat the following micro-instruction incrementing the Counter Register after each time the repeat is executed. When the lower four bits of the Counter Register are set, execute the following micro-instruction once. The lower four bits of the Counter Register are set at the completion of the repeat sequence. Thus, the repeat is executed the number of times specified in the lower four bits of the Counter Register in two's complement form.

ALU	BIT NO.	19	18	17	16	15
OP1	CONTENT	0	0	0	0	1

Meaning: Perform the following logical function in the ALU with the L-register and the S-bus:

(S+L) plus 1

where "+" means logical function "or".

ALU	BIT NO.	19	18	17	16	15
OP2	CONTENT	0	0	0	1	0

Meaning: Perform the following logical function in the ALU with the L-register and the S-bus:

(S+$\overline{L}$) plus 1

where "+" means logical function "or" and $\overline{L}$ means the ones complement of the L-register (not L).

ALU	BIT NO.	19	18	17	16	15
OP3	CONTENT	0	0	1	0	0

Meaning: Perform the following logical function in the ALU with the L-register and the S-bus:

S plus ($\overline{S \cdot L}$) plus 1

where "•" means logical function "and" and $\overline{L}$ means the ones complement of the L-register (not L).

ALU	BIT NO.	19	18	17	16	15
OP4	CONTENT	0	0	1	0	1

Meaning: Perform the following logical function in the ALU with the L-register and the S-bus:

(S+L) plus (S•$\overline{L}$) plus 1

where "•" means logical function "and", "+"means logical function "or", and $\overline{L}$ means the ones complement of the L-register (not L).

ALU	BIT NO .	19	18	17	16	15
OP5	CONTENT	0	0	1	1	1

Meaning: Perform the following logical function in the ALU with the L-register and the S-bus:

(S•$\overline{L}$)

where "•" means the logical function "and" and $\overline{L}$ means the ones complement of the L-register (not L).

ALU	BIT NO.	19	18	17	16	15
OP6	CONTENT	0	1	0	0	0

Meaning: Perform the following logical function in the ALU with the L-register and the S-bus:

S plus (S•L)

where "•" means the logical function "and".

ALU	BIT NO.	19	18	17	16	15
OP7	CONTENT	0	1	0	1	0

Meaning: Perform the following logical function in the ALU with the L-register and the S-bus:

(S+$\overline{L}$) plus (S•L)

where "+" means logical function "or", "•" means logical function "and", and $\overline{L}$ means the ones complement of the L-register (not L).

ALU	BIT NO.	19	18	17	16	15
OP8	CONTENT	0	1	0	1	1

Meaning: Perform the following logical function in the ALU with the L-register and the S-bus:

(S•L) minus 1

where "•" means the logical function "and".

ALU	BIT NO.	19	18	17	16	15
OP9	CONTENT	0	1	1	0	0

Meaning: Perform the following logical function in the ALU with the S-bus:

S plus S

ALU	BIT NO.	19	18	17	16	15
OP10	CONTENT	0	1	1	0	1

Meaning: Perform the following logical function in the ALU with the L-register and the S-bus:

(S+L) plus S

where "+" means the logical function "or".

ALU	BIT NO .	19	18	17	16	15
OP11	CONTENT	0	1	1	1	0

Meaning: Perform the following logical function in the ALU with the L-register and the S-bus:

$(S+\overline{L})$ plus S

where "+" means the logical function "or" and $\overline{L}$ means the complement of the L-register (not L).

ALU	BIT NO .	19	18	17	16	15
PASL	CONTENT	1	1	0	1	0

Meaning: Pass the L-register to the Rotate/Shifter.

ALU	BIT NO.	19	18	17	16	15
PASS	CONTENT	1	1	1	1	1

Meaning: Pass the S-bus data to the Rotate/Shifter.

ALU	BIT NO.	19	18	17	16	15
SANL	CONTENT	1	0	1	1	1

Meaning: Logical and of the S-bus and the complement of the L-register $(S \cdot \overline{L})$; pass the result to the Rotate/Shifter.

ALU	BIT NO.	19	18	17	16	15
SONL	CONTENT	1	1	1	0	1

Meaning: Logical or of the S-bus and the complement of the L-register $(S+\overline{L})$; pass the result to the Rotate/Shifter.

ALU	BIT NO.	19	18	17	16	15
SUB	CONTENT	0	0	1	1	0

Meaning: Subtract the L-register from the S-bus and pass the result to Rotate/Shifter.

ALU	BIT NO.	19	18	17	16	15
XNOR	CONTENT	1	1	0	0	1

Meaning: Logical exclusive nor of the L-register and the S-bus; $(\overline{L \ominus S})$ and pass it to the Rotate/Shifter ($\oplus$ means "exclusive or".)

ALU	BIT NO.	19	18	17	16	15
XOR	CONTENT	1	0	1	1	0

Meaning: Logical exclusive or of the L-register and the S-bus $(L \oplus S)$; pass the result to the Rotate/Shifter ($\oplus$ means "exclusive or".)

ALU	BIT NO.	19	18	17	16	15
ZERO	CONTENT	0	0	0	1	1

Meaning: Pass all zeros to the Rotate/Shifter.

ALU	BIT NO.	19	18	17	16	15
OP13	CONTENT	1	0	0	1	1

Meaning: Pass all zeros to the Rotate/Shifter.

4-5. STORE MICRO-ORDERS

STORE	BIT NO.	9	8	7	6	5
A	CONTENT	0	1	0	1	1

Meaning: Store the data on the **T**-bus in the A-register.

STORE	BIT NO.	9	8	7	6	5
B	CONTENT	0	1	0	1	0

Meaning: Store the data on the **T**-bus in the B-register.

STORE	BIT NO.	9	8	7	6	5
CAB	CONTENT	0	0	0	0	1

Meaning: Store the data on the **T**-bus in the A- or B-register according to the value of IR bit 11:

IR bit 11 set means B-register

IR bit 11 clear means A-register

STORE	BIT NO.	9	8	7	6	5
CM	CONTENT	0	1	1	0	1

Meaning: Store the data on the S-bus in the M-register, if the IR holds any Memory Reference instruction except a direct jump (JMP). Refer to the **HP 21MX Computer Series Reference Manual**, for a description of the Memory Reference instructions.

AAF or BAF is set as described under Usage for the M Store micro-order, whether or not the IR holds a Memory Reference instruction.

STORE	BIT NO.	9	8	7	6	5
CNTR	CONTENT	0	0	1	0	1

Meaning: Store the lower eight bits of the S-bus (bits 0-7) in the Counter Register.

STORE	BIT NO.	9	8	7	6	5
DSPI	CONTENT	0	0	1	1	1

Meaning: Store the lower six bits of the S-bus in the Display Indicator on the front panel.

Display Indicator Bit	5	4	3	2	1	0
Register Displayed	S	P	T	M	B	A

Usage: The six indicators on the front panel, labelled A, B, M, T, P and S are lit according to the bit(s) cleared in the Display Indicator. At power-up all bits are set until programmatically changed.

STORE	BIT NO.	9	8	7	6	5
DSPL	CONTENT	0	0	1	1	0

Meaning: Store the data on the S-bus in the Display Register on the Front Panel.

STORE	BIT NO.	9	8	7	6	5
IOO	CONTENT	0	0	1	0	0

Meaning: Direct the S-bus onto the I/O-bus.

Usage: This micro-order when used must be in the second and third instructions (T3 and T4) after IOG Special micro-order. See section 3-25 and the following sections for a description of I/O microprogramming.

STORE	BIT NO.	9	8	7	6	5
IR	CONTENT	0	1	0	0	0

Meaning: Store the data on the S-bus in the Instruction Register. Record the type of macro-instruction stored there in the Memory Protect hardware for use in determining error conditions during Instruction Register execution. See sections 3-28 and 3-34 for a description of Interfacing With Memory Protect feature.

STORE	BIT NO.	9	8	7	6	5
L	CONTENT	0	0	0	1	1

Meaning: Store the data on the S-bus in the L-register (Latch).

STORE	BIT NO.	9	8	7	6	5
M	CONTENT	0	1	0	0	1

Meaning: Store the data on the S-bus in the M-register.

Usage: An ALU micro-order (for example, INC) should also be specified in the micro-instruction. This will activate an A- or B-register addressable test. If bits 14 through 0 on the T-bus equal 1 or 2, the AAF or BAF, respectively, will be set. The M-register may be stored into immediately after a READ or WRTE Op micro-order.

STORE	BIT NO.	9	8	7	6	5
NOP	CONTENT	0	1	1	1	1

Meaning: No store operation is performed; this is the default micro-order when the Store field is left blank.

STORE	BIT NO.	9	8	7	6	5
P	CONTENT	1	1	1	1	0

Meaning: Store the data on the T-bus in the P-register (Program Address Register).

STORE	BIT NO.	9	8	7	6	5
PNM	CONTENT	0	1	1	1	0

Meaning: Store the data on the T-bus in the P-register (Program Address Register), and the data on the S-bus into the M-register (Memory Address Register).

Usage: Useful in microprograms which perform multiword READ operations from Main Memory, where the P-register points to the address in Main Memory to be read. In a single micro-instruction the microprogram can store P into the M-register via the S-bus and then increment P via the T-bus. An example of such an application is the following:

```
READ - -    INC    PNM    P
```

The A- or B-register addressable test is activated. See Usage under M micro-order, above.

STORE	BIT NO.	9	8	7	6	5
S	CONTENT	1	1	1	1	1

Meaning: Store the data on the T-bus in the S-register.

STORE		STORE	BIT NO.	9	8	7	6	5
S1	THRU	S12	CONTENT	1	n	n	n	n

nnnn is binary representation of decimal number 0 + 11

Meaning: Store the data on the T-bus in the indicated Scratch Pad Register S1 to S12.

STORE	BIT NO.	9	8	7	6	5
T	CONTENT	0	0	0	1	0

Meaning: Store the data on the S-bus in the T-register (Memory Data Register).

Usage: This micro-order should occur concurrently when a WRTE micro-order is used. The T-register is internal to the Memory System. It must not be used as a working register.

STORE	BIT NO.	9	8	7	6	5
TAB	CONTENT	0	0	0	0	0

Meaning: Store the data on the T-bus in the A-register if the AAF (A addressable Flag) is set; store the data on the T-bus in the B-register if the BAF (B addressable Flag) is set; store the data on the S-bus into the T-register (Memory Data Register) if neither AAF nor BAF is set.

Usage: Same as T micro-order.

STORE	BIT NO.	9	8	7	6	5
X	CONTENT	1	1	1	0	0

Meaning: Store the data on the T-bus in the X-register.

STORE	BIT NO.	9	8	7	6	5
Y	CONTENT	1	1	1	0	1

Meaning: Store the data on the T-bus in the Y-register.

4-6. S-BUS MICRO-ORDERS

S-BUS	BIT NO.	14	13	12	11	10
A	CONTENT	0	1	0	1	1

Meaning: Direct the data in the A-register onto the S-bus.

S-BUS	BIT NO.	14	13	12	11	10
ADR	CONTENT	0	1	0	0	0

Meaning: An address is formed on the S-bus using IR bits 0-9 and M-register bits 10-14; if IR bit 10 is clear, bits 10-14 of the address formed on the S-bus are clear. Bit 15 is always clear. IR bit 10 is the zero page/current page flag.

S-BUS	BIT NO.	14	13	12	11	10
B	CONTENT	0	1	0	1	0

Meaning: Direct the contents of the B-register onto the S-bus.

S-BUS	BIT NO.	14	13	12	11	10
CAB	CONTENT	0	0	0	0	1

Meaning: Direct the contents of the A- or B-register onto the S-bus according to the value of IR bit 11:

IR bit 11 set means B-register

IR bit 11 clear means A-register

S-BUS	BIT NO.	14	13	12	11	10
CIR	CONTENT	0	0	0	1	1

Meaning: At I/O time T6 place the contents of the Central Interrupt Register onto the S-bus and generate an IAK (Interrupt Acknowledge) signal to the I/O device. (See section 3-33 for CIR description in relation to Interrupt Handling).

Usage: This micro-order must be used after detection of an I/O interrupt to determine the select code of the interrupting device and to acknowledge that the interrupt is being serviced.

S-BUS	BIT NO.	14	13	12	11	10
CNTR	CONTENT	0	0	1	0	1

Meaning: Direct the contents of the Counter Register onto the S-bus. The 8 bit Counter Register is placed onto the low 8 bits of the S-bus; the upper 8 bits are set to ones.

S-BUS	BIT NO.	14	13	12	11	10
DSPI	CONTENT	0	0	1	1	1

Meaning: Direct the six bits of the display Indicator from the Front Panel to the S-bus. The upper 10 bits of the S-bus are set to ones.

Usage: See DSPI Store field definition for Display Indicator bit significance.

S-BUS	BIT NO.	14	13	12	11	10
DSPL	CONTENT	0	0	1	1	0

Meaning: Direct the contents of the Front Panel Display Register onto the S-bus.

There are five micro-order classifications in Word Type 3:

- "JMP" — Op Code used in conjunction with "CNDX" specifies Word Type 3, a conditional jump.
- "CNDX" — SPECIAL Code specifying Word Type 3.
- CONDITION — Condition that must be satisfied before jump is executed.
- JUMP SENSE — Optional code to invert the jump condition.
- OPERAND — Target address of jump.

All micro-order groups, except the OPERAND, are defined by the mnemonic, its binary equivalent, meaning, and, where necessary, by conventions in their use.

4-12. "JMP" MICRO-ORDER

"JMP"	BIT NO.	23	22	21	20
JMP	CONTENT	1	1	0	1

Meaning: Used in conjunction with the SPECIAL Code "CNDX", the CONDITION code specifies the condition under which a jump to the address specified in the OPERAND will take place. If the JUMP SENSE code "RJS" is specified, the CONDITION code specifies the condition under which no jump will take place.

4-13. "CNDX" MICRO-ORDER

"CNDX"	BIT NO.	4	3	2	1	0
CNDX	CONTENT	1	1	0	0	1

Meaning: Used in conjunction with the Op code "JMP", this micro-order specifies a conditional jump and Word Type 3.

4-14. CONDITION MICRO-ORDERS

The ALU and T-bus condition flags are set after each Word Type 1 or 2 micro-instruction. They are not changed during JMP or JSB micro-instructions (Word Types 3 and 4). Thus, several different jump tests can be made without losing the flag results.

CONDITION	BIT NO.	19	18	17	16	15
AL0	CONTENT	0	0	0	1	1

Meaning: Bit 0 of the last output from the ALU was set (tested before the Rotate/Shifter) by the last Word Type 1 or 2 micro-instruction.

CONDITION	BIT NO.	19	18	17	16	15
AL15	CONTENT	0	0	1	0	0

Meaning: Bit 15 of the last output from the ALU was set (tested before the Rotate/Shifter) by the last Word Type 1 or 2 micro-instruction.

CONDITION	BIT NO.	19	18	17	16	15
ASGN	CONTENT	0	1	1	1	0

Meaning: Alter/skip macro-instruction condition is not satisfied.

CONDITION	BIT NO.	19	18	17	16	15
CNT4	CONTENT	1	1	1	1	0

Meaning: The right (least significant) 4 bits of the Counter Register are all ones.

CONDITION	BIT NO.	19	18	17	16	15
CNT8	CONTENT	0	0	1	1	0

Meaning: All eight bits of the Counter Register are ones.

CONDITION	BIT NO.	19	18	17	16	15
COUT	CONTENT	0	0	0	1	0

Meaning: The ALU Carry Out Flag bit was set by the last ALU operation (tested before the Rotate/Shifter) of the last Word Type 1 or 2 micro-instruction.

CONDITION	BIT NO.	19	18	17	16	15
E	CONTENT	0	1	0	0	1

Meaning: The Extend Register bit is set.

CONDITION	BIT NO.	19	18	17	16	15
FLAG	CONTENT	0	1	0	0	0

Meaning: The CPU FLAG bit is set.

CONDITION	BIT NO.	19	18	17	16	15
FPSP	CONTENT	0	0	1	1	1

Meaning: A special signal is present issued by certain non-standard CPU Front Panels.

CONDITION	BIT NO.	19	18	17	16	15
INT	CONTENT	1	1	0	1	0

Meaning: An Interrupt is pending.

CONDITION	BIT NO.	19	18	17	16	15
IR2	CONTENT	0	1	1	1	1

Meaning: Instruction Register bit 2 is set.

CONDITION	BIT NO.	19	18	17	16	15
NDEC	CONTENT	1	0	0	1	1

Meaning: The "DEC M" (Decrement M-register) button on the Front Panel was **not** actuated.

CONDITION	BIT NO.	19	18	17	16	15
NHOI	CONTENT	0	1	1	0	0

Meaning: The RUN/HALT switch on the Front Panel is set to "Run" and there is no interrupt pending (i.e. no halt and no interrupt).

Usage: This micro-order is recommended for use in long microprograms. (85 microseconds or longer is the criterion used by Hewlett-Packard produced microprograms.) This is necessary since microprograms cannot be interrupted. A pending interrupt or halt condition is not detected unless a specific test is made for them.

CONDITION	BIT NO .	19	18	17	16	15
NINC	CONTENT	1	0	0	1	0

Meaning: The "INC M" (Increment M-register) button on the Front Panel was not actuated.

CONDITION	BIT NO.	19	18	17	16	15
NLDR	CONTENT	1	0	0	0	0

Meaning: The "IBL" (loader) button on the Front Panel was not actuated.

CONDITION	BIT NO.	19	18	17	16	15
NLT	CONTENT	1	0	1	0	1

Meaning: The "←" REGISTER SELECT LEFT button on the Front Panel was not actuated.

CONDITION	BIT NO.	19	18	17	16	15
NMLS	CONTENT	0	0	1	0	1

Meaning: Memory was not lost as a result of the last power down or power failure.

CONDITION	BIT NO.	19	18	17	16	15
NOP	CONTENT	1	1	1	0	1

Meaning: No condition test is made; no jump occurs.

Usage: This is the default micro-order if none is specified in the condition field.

CONDITION	BIT NO.	19	18	17	16	15
NRST	CONTENT	1	0	1	1	1

Meaning: The DISPLAY button on the Front Panel was not actuated.

CONDITION	BIT NO.	19	18	17	16	15
NRT	CONTENT	1	0	1	0	0

Meaning: The "→" REGISTER SELECT RIGHT button on Front Panel was not selected.

CONDITION	BIT NO.	19	18	17	16	15
NSFP	CONTENT	1	1	0	0	1

Meaning: A standard Front Panel is not installed on the CPU.

CONDITION	BIT NO.	19	18	17	16	15
NSNG	CONTENT	1	0	0	0	1

Meaning: The INSTR STEP (Instruction Step) button on the Front Panel was not actuated.

CONDITION	BIT NO .	19	18	17	16	15
NSTB	CONTENT	1	1	0	0	0

Meaning: None of the following Front Panel buttons were actuated:

INSTR STEP (Instruction Step)
"→" REGISTER SELECT RIGHT
"←" REGISTER SELECT LEFT
DISPLAY
IBL (Binary Loader)
INC M (Increment M-register)
DEC M (Decrement M-register)
STORE
RUN
PRESET

JUMP MODIFIER	BIT NO.	4	3	2	1	0
JEAU	CONTENT	1	1	1	1	1

Meaning: Enable the EAU jump table. According to the particular EAU macro-instruction held in the Instruction Register, the least significant three bits (0-2) of the OPERAND are replaced by EAU jump table bits (bits 4-9 and 11 of the Instruction Register actually determine the OPERAND address modification):

EAU Macro-instruction	Three LSB's of Address
RRR	000
ASR	001
LSR	010
(not used)	011
RRL	100
ASL	101
LSL	110
MPY	111

JUMP MODIFIER	BIT NO.	4	3	2	1	0
JIO	CONTENT	1	1	0	1	0

Meaning: Perform the JMP or JSB modifying OPERAND bits 2 and 3 according to the I/O instruction jump table (bits 6, 7, and 8 of the I/O macro-instruction in the Instruction Register actually determine the OPERAND address modification):

IR Contains I/O Macro-instruction	IR Bits 8 7 6	OPERAND Bits 3 & 2 Replaced By:
MIA or MIB	1 0 0	1 1
LIA or LIB	1 0 1	1 0
OTA or OTB	1 1 0	0 1
HLT	0 0 0	0 0
CLO or CLF	0 0 1	0 0
STO or STF	0 0 1	0 0
SFC or SOC	0 1 0	0 0
SFS or SOS	0 1 1	0 0
STC or CLC	1 1 1	0 0

JUMP MODIFIER	BIT NO.	4	3	2	1	0
JTAB	CONTENT	1	1	0	1	0

Meaning: Perform a jump to a location within the Basic Instruction Set microprogram based on the eight most significant bits of the Instruction Register. This is accomplished via a table look up of the address in the Main Jump Table for the basic instruction set. This micro-order is executed independently of word types; hence JMP or JSB need not be specified.

JUMP MODIFIER	BIT NO.	4	3	2	1	0
J30	CONTENT	1	1	1	0	1

Meaning: Replace the four Least Significant Bits of the OPERAND with bits 3 through 0 of the Instruction Register.

JUMP MODIFIER	BIT NO.	4	3	2	1	0
J74	CONTENT	1	1	1	0	0

Meaning: Replace the four Least Significant Bits of the OPERAND with bits 7 through 4 of the Instruction Register.

JUMP MODIFIER	BIT NO.	4	3	2	1	0
RTN	CONTENT	1	1	1	1	0

Meaning: Return to the address stored in the Save Register as a result of a subroutine jump (JSB); if the Save Register is equal to zero (no subroutine is active), return to address 0 of Control Store to initiate the reading of the next macro-instruction from Main Memory.

JUMP MODIFIER	BIT NO.	4	3	2	1	0
STFL	CONTENT	0	1	0	0	0

Meaning: Set the CPU Flag and then perform the JMP or JSB to the OPERAND address. No modification is made to the OPERAND address.

JUMP MODIFIER	BIT NO.	4	3	2	1	0
UNCD	CONTENT	1	1	0	0	0

Meaning: Perform the JMP or JSB to the OPERAND address. No modification is made to the OPERAND address.

Usage: This is the default micro-order if no JUMP MODIFIER is specified.

4-20. THE OPERAND MICRO-ORDER

OPERAND
An Address

BIT NO.	16	15	• • • • • • • • • •	6	5
CONTENT	Binary Address Equivalent				

The ADDRESS can be a decimal, octal or computed number:

Decimal number, d, in the range 0 to 4095

Octal number, kB, in the range 0B to 7777B where B signifies octal

Computed number, c, which is within the decimal or octal range, according to whether it is computed from octal or decimal values, of the form:

a. *+kB
b. *−kB
c. *+d
d. *−d
e. LABEL+kB
f. LABEL−kB
g. LABEL+d
h. LABEL−d
i. LABEL

where * means "this address" and LABEL means a micro-instruction label that is defined elsewhere in the microprogram.

Examples:

*+11B, *+9, HERE+5, START

4-21. PSEUDO INSTRUCTIONS

There are five pseudo instructions recognized by the micro-assembler: DEF, EQU, ONES, SKP, and ZEROES.

4-22. DEF

Character Column:

1 10 15 20 25 30 40 80

Fields:	Field 1	Field 2	Fields 3-5	Field 6	Field 7
Content:	LABEL (optional)	"DEF"	(blank)	ADDRESS	COMMENTS

The DEF statement creates a 24 bit micro-instruction word in ROM the contents of which is a 12 bit binary address defined by "ADDRESS" in the micro-assembler input record (Field 6). The binary address is associated in the microprogram with the optional LABEL, if defined.

The ADDRESS can be a decimal, octal or computed number:

Decimal number, d, in the range 0 to 4095

Octal number, kB, in the range 0B to 7777B, where B signifies octal

Computed number, c, which is within the decimal or octal range, according to whether it is computed from octal or decimal values, of the form:

a. *+kB
b. *−kB
c. *+d
d. *−d
e. LABEL+kB
f. LABEL−kB
g. LABEL+d
h. LABEL−d
i. LABEL

where * means "this address" and LABEL means a micro-instruction label that is defined elsewhere in the microprogram.

Examples of DEF statements:

Character Column:

1 10 30

Fields:	Field 1	Field 2	Field 6
Content:	AD1	DEF DEF DEF	SRF+150 ASGNOP 416B

4-23. EQU

Character Column:

1 10 15 20 25 30 40 80

Fields:	Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7
Content:	LABEL	"EQU"	(blank)	(blank)	(blank)	ADDRESS	COMMENTS

The EQU statement associates the stated LABEL with a 12 bit address. This statement does not result in an address being stored in ROM. The ADDRESS can be a decimal, octal or computed number:

Decimal number, d, in the range 0 to 4095

Octal number, kB, in the range 0B to 7777B, where B signifies octal

Computed number, c, which is within the decimal or octal range, according to whether it is computed from octal or decimal values, of the form:

a. *+kB

b. *−kB

c. *+d

d. *−d

e. LABEL+kB

f. LABEL−kB

g. LABEL+d

h. LABEL−d

i. LABEL

where * means "this address" and LABEL means a micro-instruction label that is defined in the micro-program before this statement.

Examples of EQU statements:

Character Column:

1 10 30

Fields:	Field 1	Field 2	Field 6
Content:	HALT	EQU	400B
	RELO	EQU	6000B
	START	EQU	RELO

4-24. ONES

Character Column:

1 10 15 20 25 30 40 80

Fields:	Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7
Content:	LABEL	"ONES"	(blank)	(blank)	(blank)	(blank)	COMMENTS

The ONES statement creates a 24 bit micro-instruction word in ROM consisting of ones in all 24 bits.

Example of a ONES statement:

Character Column:

1 10

Fields:	Field 1	Field 2
Content:	NEG 1	ONES

4-25. SKP

Character Column:	1	10	15	20	25	30	40	80
Fields:	Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7	
Content:	(blank)	"SKP"	(blank)	(blank)	(blank)	(blank)	COMMENTS	

The SKP statement commands the micro-assembler to skip to the Top of the next page (TOP OF FORM command) during the listing of the microprogram. No locations in ROM are used, when this statement is specified.

Example of a SKP statement:

Character Column:	1	10
Fields:	Field 1	Field 2
Content:		SKP

4-26. ZEROES

Character Column:	1	10	15	20	25	30	40	80
Fields:	Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7	
Content:	LABEL	"ZEROES"	(blank)	(blank)	(blank)	(blank)	COMMENTS	

The ZEROES statement creates a 24 bit micro-instruction word in ROM consisting of zeroes in all 24 bits.

Example of a ZEROES statement:

Character Column:	1	10	40
Fields:	Field 1	Field 2	Field 7
Content:	NULL	ZEROES	NO BITS

$INPUT

(Used by BCS systems only)

General Form: $INPUT = lun

The logical unit number, lun, must be octal and in the range 1 - 74.

Meaning: The logical unit number of the device through which all subsequent input (to the next $END statement) is to be read is "lun."

Purpose: When the assembly process is begun in BCS systems, the micro-assembler expects the first source statement to be entered through the system console device (logical unit number 5). The user may enter the whole source program through the system console device. Normally, however, the user enters a $INPUT command specifying the logical unit number of the card reader or paper tape reader from which the rest of the source program is to be read.

Example: $INPUT = 12

$LIST

General Form: $LIST = lun

The logical unit number, lun, must be octal and in the range 1 - 74.

Meaning: The logical unit number of the listing device is "lun".

Purpose: To cause the assembly listing to be printed on the device having the specified unit number. If omitted, logical unit number is assumed to be 6 (standard list device).

Example: $LIST = 16

$NOPUNCH

General Form: $NOPUNCH

Meaning: Suppress punching of binary object tape.

Purpose: To perform a micro-assembly for listing and diagnosis only.

Example: $NOPUNCH

$ORIGIN

General Form: $ORIGIN = nnn

The origin, nnn, must be octal and in the range 0 - 7777.

Meaning: Set microprogram origin at octal address nnn in Control Store.

Purpose: Every microprogram must have its program address origin defined. New origins may be specified within the microprogram.

Example: $ORIGIN = 427

$RCASE

General Form: $RCASE

Meaning: Punch a special 32-micro-instructions/record object tape.

Purpose: This special object tape is reserved for system maintenance. Refer to Section 5-6 Micro-Assembler Output for a description of this special object tape.

Example: $RCASE

$OUTPUT

General Form: $OUTPUT = lun

The logical unit number, lun, must be octal and in the range 1 - 74. This statement may come anywhere before the $END statement.

Meaning: lun is the logical unit number of the output device.

Purpose: To specify the device on which the micro-assembler object code is to be output. If this statement is omitted, logical unit of 4 is assumed.

Example: $OUTPUT = 10

$PASS 2

(Used by BCS systems only)

General Form: $PASS2 = lun

The logical unit number, lun, must be octal and in the range 1 - 74. If present, this must be the first statement in the source deck or tape.

Meaning: lun is the logical unit number of the magnetic tape unit onto which all subsequent micro-assembler input is to be written.

Purpose: To cause all source input to be recorded on magnetic tape for use as input to Pass 2 of the micro-assembler. If this control statement is omitted, the computer halts at the end of Pass 1 to allow the operator to reload the microprogram source into the "$INPUT" device.

Note: The only magnetic tape units supported by the micro-assembler are the HP 3030 and HP 7970.

Example: $PASS2 = 23

$SUPPRESS

General Form: $SUPPRESS

Meaning: Suppress all warning error messages.

Purpose: To cut down the volume of messages to the console device. Fatal error messages will still be printed.

Example: $SUPPRESS

$SYMTAB

General Form: $SYMTAB

Meaning: Print symbol table

Purpose: To provide the user with label names and corresponding octal addresses used in his microprogram.

Example: $SYMTAB

5-6. MICRO-ASSEMBLER OUTPUT

This section describes all forms of output from the micro-assembler. They are:

- Binary Object
- Symbol Table
- Source and Binary Microprogram Listing
- Error Messages

5-7. BINARY OBJECT OUTPUT

The Standard Object Tape output by the micro-assembler to paper tape or a disc file consists of one or more Instruction Records, the format of which is shown in Appendix A, Figure A-1. One Instruction Record holds up to 27 micro-instructions and five words of header information. Each micro-instruction requires 32 bits or two words in the format: an eight bit address and 24 bits for the micro-instruction. Hence the length of the record =

5 words of header
2**n** words for **n** micro-instructions (2 words for each micro-instruction)
5+2**n** words for one Instruction Record

No more than 27 micro-instructions are written into an Instruction Record. Hence the maximum length = 5+(2x27)=59 words. The last object record is a four word End Record. When the microprogram consists of more than 27 micro-instructions, a series of Instruction Records are produced with the last one holding 27 or less micro-instructions. For example, if 57 micro-instructions have been assembled, three Instruction Records and an End Record are required consisting of the following:

a. Instruction Record 1 holds 27 micro-instructions and consists of

5 words of header
54 words for 27 micro-instructions
59 words

b. Instruction Record 2 holds 27 micro-instructions and consists of

5 words of header
54 words for 27 micro-instructions
59 words

c. Instruction Record 3 holds 3 micro-instructions and consists of

5 words of header
6 words for 3 micro-instructions
11 words

d. The End Record consists of

4 words
133 words for the entire microprogram Binary Object.

The Standard Object format is accepted by all programs which accept standard relocatable format. Thus a Standard Object tape can be stored in a DOS-III file using the ":STORE,R,..." directive. However, if the DOS-III user wants the Binary Object stored automatically in a disc file by the micro-assembler, the DOS-III directive "STORE,B,. . ." must have previously been used to reserve a disc file.

The Micro-assembler can also produce a non-standard object as the result of the inclusion of the $RCASE control statement. This optional object is the HP ROM Simulator Object tape. The format of this tape is shown in Appendix A, Figure A-2.

5-8. SYMBOL TABLE LISTING

If the user has a $SYMTAB control statement in his microprogram source input, then the micro-assembler will print a symbol table on the device with logical unit number 6 or on the device defined by the $LIST control statement, if present.

An example of a symbol table is shown in Figure 5-2.

On the left are the symbols or labels in the microprogram. On the right is the value of the symbol; that is the six digit absolute octal address of the symbol. Where X follows the address, the symbol has been defined by a $EXTERNAL control statement.

c. Summon the Micro-assembler with statement

```
:PR,MICRO,[p1,p2,p3,p4,99]
```

where

p1 = the input device logical unit number

p2 = list device logical unit number

p3 = paper tape punch device logical unit number

p4 = maximum number of lines-per-page on the list device.

If 99 is entered for any of the above parameters, that parameter and all those that follow are defaulted to "standard" values.

d. The program title

```
MICRO-ASSEMBLER
```

is printed and Pass 1 begins. If a $SYMTAB control statement is in the source microprogram, the symbol table is printed at the conclusion of Pass 1. Pass 2 begins immediately (from disc) and the listing and relocatable object tape are output. Micro-assembly is complete.

Note: If Pass 2 fails to begin, check that the paper tape punch is turned on. The micro-assembler will cycle in a loop until the punch is turned on.

5-12. BCS OPERATION OF MICRO-ASSEMBLER

Before proceeding, the following items must be available:

- An absolute BCS binary tape.
- A reloctable object tape of the Micro-assembler program MICRO (HP 12978-160003).
- A source microprogram either on cards or paper tape.

For a detailed description of BCS usage, see the **Basic Control System** manual (HP 02116-9017).

The following procedure need be performed only once. When an absolute binary tape of the Micro-assembler is punched, it is used as described in the procedure "Executing the Micro-assembler."

Making an Absolute Micro-assembler tape:

a. Load the absolute BCS binary tape using the Basic Binary Loader.

b. Set the P-register to 2. Set bit 14 of the Switch Register and clear all other Switch Register bits.

c. Place the MICRO relocatable object tape in the paper tape reader. Check that the paper tape reader and the console device are on. Turn on the paper tape punch. Press PRESET and RUN on the CPU front panel. MICRO reads in and absolute binary tape is punched.

d. The message

```
*LOAD
```

is printed and the computer waits. Set Switch Register bits 2 and 14 leaving all others clear. Load BCS Library tape into the paper tape reader. Press RUN.

e. The BCS Library tape reads in and the rest of the absolute binary tape is punched. Linkage information is printed on the console device.

This is the absolute binary tape of MICRO, used for input to the next step.

Executing the Micro-assembler:

a. Load the MICRO absolute binary tape using the Basic Binary Loader.

b. When loading is complete, set P-register to 2. Press PRESET and RUN. The message

```
MICRO-ASSEMBLER
```

is printed followed by a request for the logical unit number of the source input device.

c. Enter the logical unit number followed by carriage return/line feed. Pass 1 now begins. If a $SYMTAB control statement is in the microprogram source, the symbol table is printed at the conclusion of Pass 1. (See Section 5-5 for a description of the $SYMTAB control statement.)

d. Turn on the paper tape punch.

e. Pass 2 begins immediately. If no $PASS2 control statement was included in the source, the message

```
RELOAD SOURCE, PRESS RUN
```

is printed. Reload the source microprogram into the input device and then press RUN on the front panel of the computer.

Note: If Pass 2 fails to begin, check that the paper tape punch is turned on. The micro-assembler will cycle in a loop until the punch is turned on.

If a teletype is used for both listing and punching, the computer halts (T-register = 102052) so that the operator can press the paper tape punch ON button to punch the microprogram object tape. The operator then presses RUN on the computer front panel.

When the paper tape is punched, another halt (T-register = 102053) occurs, so that the paper tape punch button can be set to OFF. Press RUN on the computer front panel.

f. Pass 2 completes micro-assembly. The microprogram object tape is complete. To assemble another microprogram proceed from step b.

5-13. MICRO DEBUG EDITOR

The Micro Debug Editor (MDE) makes it possible to load the object microprograms output from the Microassembler into a Writable Control Store module. It also provides the ability to debug microcode stored in the WCS and to "burn" microprograms into ROM chips.

Before using the Micro Debug Editor to debug microprograms, the Writable Control Store PCAs must be set to the required control store module numbers. This is accomplished by the installation of a module selection Jumper Assembly (HP Part Number 5060-8342). Refer to Section 6 of this manual for installation of the module selection Jumper Assembly and the WCS PCAs.

5-14. HARDWARE ENVIRONMENT

The BCS version requires the following minimum hardware:

a. HP 21MX Series Computer with 8K of Main Memory

b. A console device

c. A paper tape reader

d. One or more WCS PCA's, depending on the size of the microprogram to be debugged.

e. If a ROM program tape is to be punched, a paper tape punch is also required.

The DOS-III version of the MDE requires the same minimum hardware as the DOS-III system.

5-15. INITIALIZATION PROGRAM

When the Micro Debug Editor is to be run for debugging purposes (as opposed to being run merely to punch ROM program tapes), the user must supply an initialization program. The initialization program is an assembly language program that prepares the necessary parameters in

```
ASMB,R,B,L,T                         Assembly parameters
      NAM TEXT,6                     Program name (DOS-III)
      ENT TEST,MACRO                 Entry points

TEST  NOP
       .                             Any initialization procedure re-
       .                             quired by the microprogram
       .

MACRO OCT 105xxx                     (or 101xxx) Instruction that calls
                                     the user microprogram
      DEF P1
      DEF P2
         .                           Parameter addresses required by
         .                           the microprogram
         .

      DEF Px

      JMP TEST,I                     Return to calling program (MDE)

P1    (parameter 1 value)
P2    (parameter 2 value)
       .
       .                             Parameter values
       .

Px    (parameter x value)

      END
```

Figure 5-4. General Format of the Initialization Program

WRITABLE CONTROL STORE

SECTION VI

This section covers general information, installation, programming, and general theory of operation for the HP 12978A Writable Control Store Interface Kit. Options 001 and 002 for the interface kit are also covered in this section.

6-1. GENERAL INFORMATION

The Hewlett-Packard 12978A Writable Control Store Interface Kit provides the HP 21MX Computers with the necessary logic to dynamically change the instruction set of the computer. The printed-circuit assembly and flat cable assembly contained in the interface kits are shown in figure 6-1 and listed in table 6-1.

6-2. IDENTIFICATION

Hewlett-Packard uses five digits and a letter (12978A) for standard kit designations. If the designation of your kit does not agree with this number, there are differences between your kit and the kit described in this manual.

6-3. INTERFACE KIT CONTENTS

Table 6-1. Interface Kit Contents

INTERFACE KIT	CONTENTS	HP PART NO.
12978A	Writable Control Store PCA	12908-60006*
	Flat Cable Assembly 5 Connectors	5060-8393
	Microprogramming 21MX Computers	02108-90008
	Diagnostic Paper Tape	12908-60001
	Diagnostic Manual	12908-90013
*Only PCAs with a date code of 1436 or higher are suitable for 21MX applications.		

6-4. CONTENTS OF INTERFACE KIT OPTIONS

There are two 12978A Interface Kit Options. They contain material in addition to that contained in the basic interface kit. Option 001 provides all the software required for use of the writable control store in the DOS-III system. Option 002 provides all the software required for the use in the BCS system.

Table 6-2. Additional Material for Interface Options

OPTION	ADDITIONAL MATERIAL	HP PART NO.
12978A-001	DOS-III WCS Driver	24278-60001
	DOS-III WCS I/O Utility	24333-60001
	DOS-III Micro-assembler	12978-16001
	DOS-III Micro Debug Editor	12978-16002
	DOS WCS Driver Manual	12908-90004
12978A-002	BCS WCS Driver	24277-60001
	BCS WCS I/O Utility	24283-60001
	BCS Micro-assembler	12978-16003
	BCS Micro Debug Editor	12978-16004
	BCS WCS Driver Manual	12908-90003

6-5. SPECIFICATIONS

Table 6-3 lists the characteristics and specifications of the writable control store PCA.

6-6. INSTALLATION

6-7. UNPACKING AND INSPECTION

If the shipping carton is damaged upon receipt, request that the carrier's agent be present when the kit is unpacked. Inspect the kit for damage (cracked, broken parts, etc.). If the kit is damaged and fails to meet specifications, notify the carrier and the nearest HP Sales and Service Office immediately. (Sales and Service Offices are listed at the back of this manual.) Retain the shipping container and the packing material for the carrier's inspection. The HP Sales and Service Office will arrange for the repair or replacement of the damaged item without waiting for any claims against the carrier to be settled.

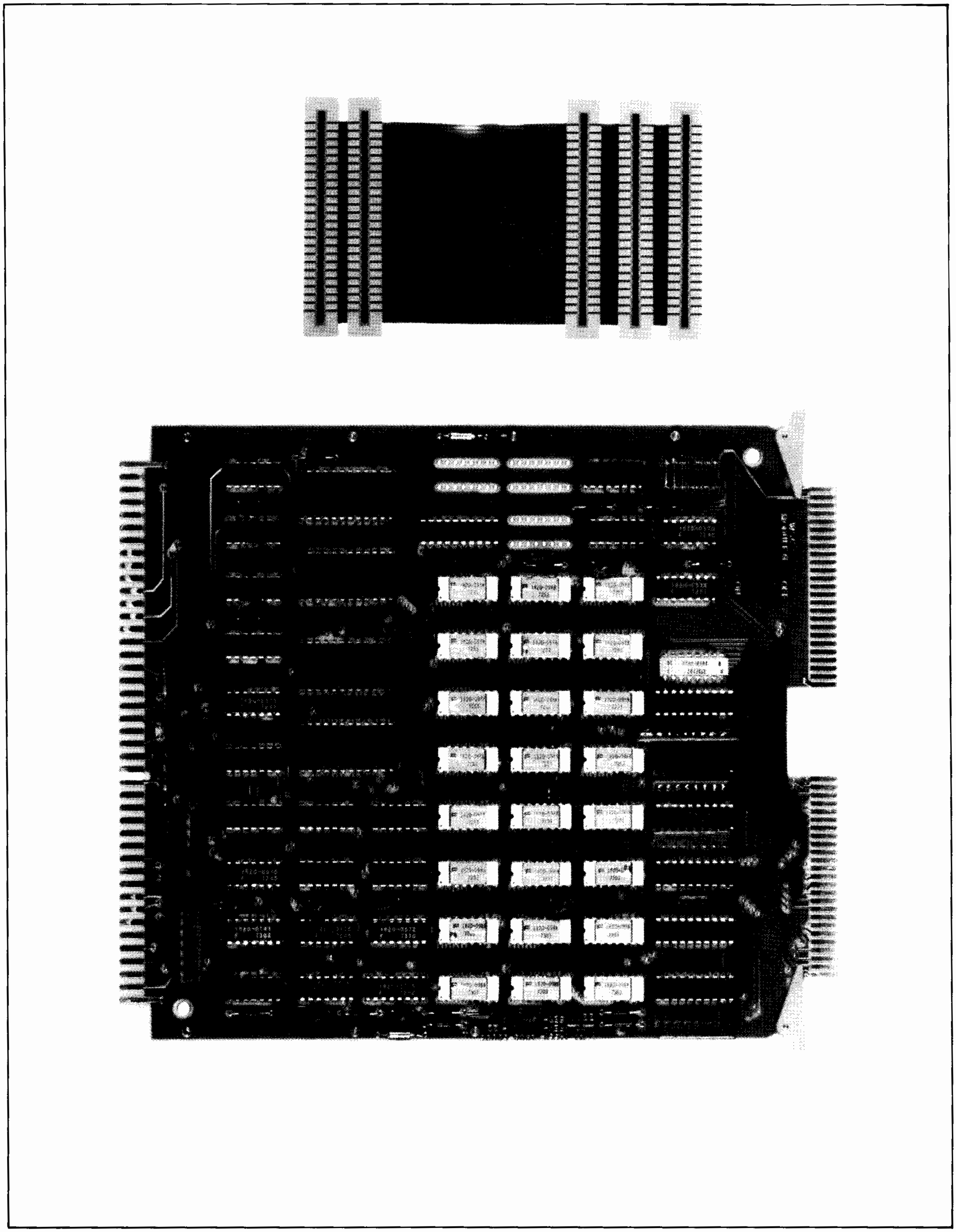


Figure 6-1. Writable Control Store Interface Kit

Table 6-3. Writable Control Store PCA Specifications

CAPACITY

Words Available: 256 per module

Maximum WCS Modules: one per HP 2105; two per HP 2108

Word Size: 24 bits

MICRO-INSTRUCTION TIME

Access: 162 ns.

Full Micro-instruction Cycle: 325 ns.

INSTALLATION

One writable control store PCA requires the use of one Input/Output slot (slot 10). Writable control store may be used as any module, except module 0.

DATA STORAGE

Input/Output Group instructions or an HP 21MX Dual Channel Port Controller are used to load the WCS.

DATA READBACK

Input/Output Group instructions only are used to read data from the WCS.

INTERFACE CURRENT SUPPLIED BY COMPUTER

0.15A (−2V supply); 4.6A (+5V supply)

PCA DIMENSIONS

Width: 7-3/4 inches (196.8 mm)

Height: 8-11/16 inches (220.7 mm)

PCA WEIGHT

Net Weight: 18 oz (511.2 gm) (card and cable only)

Shipping Weight: 4 lb (2.27 kg)

PCA INPUT LEVELS

"1" state: 1.9 volts minimum

"0" state: 1.1 volts maximum

PCA OUTPUT LEVELS

"1" state: 2.4 volts minimum

"0" state: 0.7 volts maximum

6-8. INSTALLATION

Install the writable control store kit as follows:

a. Ensure that the computer operates properly prior to installing the writable control store interface kit.

b. Turn off power at the computer.

c. Remove the bottom and back access covers from the computer.

d. On the writable control store remove the appropriate jumper wires from TB1 to select the desired module number (see figure 6-2 for pin number configuration). Refer to table 6-4 for the desired module number and jumper removal.

e. On the writable control store PCA place the WCS module 0 enable switch S1 in the OFF position.

f. Place the first writable control store PCA in slot number 10 (select code 10) of the I/O section of the computer. Any additional writable control store PCAs should be placed in slot 11.

Note: When WCS PCAs are installed, computer software must be reconfigured because of the changed I/O slot usage. If adding WCS PCA(s) will overburden the Power Supply of the computer, it may be necessary to move some I/O PCAs to an I/O Extender, HP 12979A.

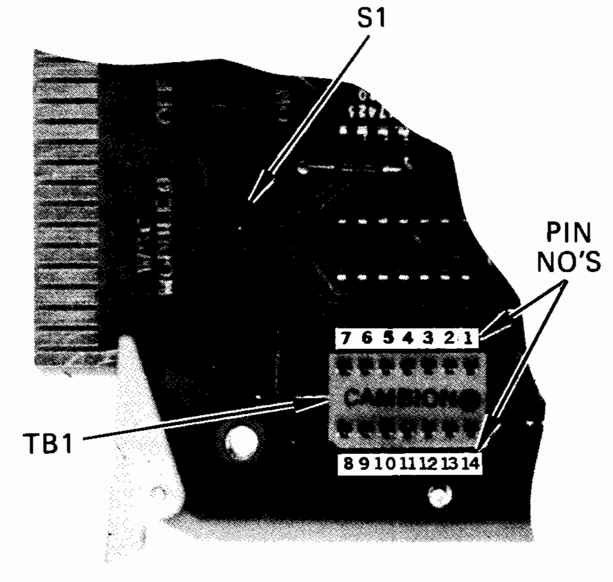


Figure 6-2. WCS Terminal Board for Selecting Module Number Position

Table 6-4. WCS PCA Jumper Removal on Terminal Board for Various Module Selections

MODULE	JUMPERS TO BE REMOVED
0	None
1	Pins 6,9
2	Pins 5,10
3	Pins 6,9; 5,10
4	Pins 4,11
5	Pins 6,9; 4,11
6	Pins 5,10; 4,11
7	Pins 6,9; 5,10; 4,11
8	Pins 3,12
9	Pins 6,9; 3,12
10	Pins 5,10; 3,12
11	Pins 6,9; 5,10;, 3,12
12	Pins 4,11; 3,12
13	Pins 6,9; 4,11; 3,12
14	Pins 5,10; 4,11; 3,12
15	Pins 6,9; 5,10; 4,11; 3,12

g. Remove the ROM-CPU Interconnect assembly, part no. 5060-8344. Install the connectors of the flat cable assembly, part no. 5060-8393

1. on J1 of the ROM Control PCA 1, A7
2. on J2 of the CPU A1
3. on J1 of each WCS PCA

as shown in sideview on figure 6-3.

Note: If an I/O PCA is installed immediately above the WCS (refer to figure 6-3) that requires a cable (hood) connector on the back, then it may be necessary to double the flat cable assembly back or cut it to make room for the I/O cable connector.

h. Replace the bottom and back access covers on the computer.

i. Turn on power at the computer and perform the diagnostic test as outlined in the Diagnostic Program Procedures (part no. 12908-90009) shipped with the 12978A Interface Kit. If the diagnostic program is completed without error, the PCA is installed and operating properly. If the diagnostic program indicates errors, halt the computer, turn off power, and recheck all of the above installation procedures. Correct where necessary, then recheck and repeat the operating procedures of the diagnostic.

6-9. RESHIPMENT

If an item of the kit is to be shipped to Hewlett-Packard for service or repair, attach a tag to the item identifying the owner and indicating the service or repair to be accomplished. Include the model number of the kit. Package the item in the original factory packaging material, if available. If the original material is not available, standard factory packaging material can be obtained from a local Hewlett-Packard Sales and Service Office. If standard factory packaging material is not used, wrap the item in Air Cap TH-240 Cushioning (or equivalent) manufactured by Sealed Air Corp., Hawthorne, N.J. and place in a corrugated carton (200 pound test material). Seal the shipping carton securely and mark it "FRAGILE" to ensure careful handling.

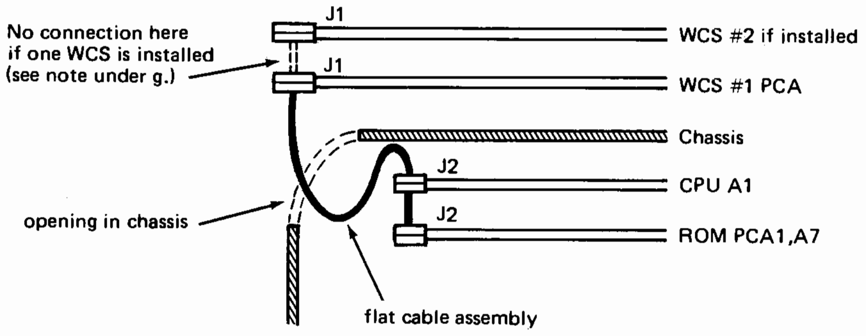


Figure 6-3. Installation of Flat Cable Assembly

MICRO-ORDER SUMMARY

APPENDIX

C

Table C-1. Summary of User Micro-orders

MICRO-ASSEMBLER SOURCE (CARD)	OP	SPECIAL	ALU	JMP COND	IMMEDIATE MODIFIER	STORE	RJS	S-BUS
COLUMN NO.	10	15	20	20	20	25	25	30
BITS (ROM)	23-20	4-0	19-15	19-15	19-18	9-5	14	14-10
Corresponding Bit Pattern								
00000	*NOP	IOFF	INC	TBZ	HIGH	TAB	†RJS	TAB
00001	ARS	SRG2	OP1	ONES	LOW	CAB		CAB
00010	CRS	L1	OP2	COUT	CMHI	T		T
00011	LGS	L4	ZERO	AL0	CML0	L		CIR
00100	MPY	R1	OP3	AL15		IOO		IOI
00101	DIV	ION	OP4	[redacted]		CNTR		CNTR
00110	LWF	SRG1	SUB	CNT8		DSPL		DSPL
00111	WRTE	[redacted]	OP5	[redacted]		DSPI		DSPI
01000	ASG	STFL	OP6	FLAG		IR		ADR
01001	READ	CLFL	ADD	E		M		M
01010	ENV	FTCH	OP7	OVFL		B		B
01011	ENVE	SOV	OP8	RUN		A		A
01100	JSB	COV	OP9	NHOI		[redacted]		LDR
01101	JMP	RPT	OP10	SKPF		CM		[redacted]
01110	IMM	SRGE	OP11	ASGN		PNM		[redacted]
01111		*NOP	DEC	IR2		*NOP		*NOP
10000		[redacted]	CMPS	NLDR		S1		S1
10001		MPCK	NOR	NSNG		S2		S2
10010		IOG	NSAL	NINC		S3		S3
10011		ICNT	OP13	NDEC		S4		S4
10100		SHLT	NAND	NRT		S5		S5
10101		INCI	CMPL	NLT		S6		S6
10110		[redacted]	XOR	NSTR		S7		S7
10111		SRUN	SANL	NRST		S8		S8
11000		**UNCD	NSOL	NSTB		S9		S9
11001		CNDX	XNOR	NSFP		S10		S10
11010		JIO	PASL	INT		S11		S11
11011		JTAB	AND	SRGL		S12		S12
11100		J74	ONE	RUNE		X		X
11101		J30	SONL	*NOP		Y		Y
11110		RTN	IOR	CNT4		P		P
11111		JEAU	*PASS	[redacted]		S		S

*default micro-order

**JMP default

†If no 'RJS', then bit 14 = 1

▨ means not normally used by user microprogrammer.

■ means included here for completeness only; reserved for exclusive use of system microprogrammers.