

NAKED MINI® 4

PROCESSOR ACCEPTANCE TEST

(PAT4)

93802-90



Engineering Reference Number
90-93802-00A4

NAKED MINI Division
HERTFORD HOUSE
DENHAM WAY, MAPLE CROSS
RICKMANSWORTH, HERTFORDSHIRE WD3 2X
Telephone Rickmansworth 71211 Telex 9226

© 1979 Computer Automation, Inc.
Printed in U.S.A.

NAKED MINI® 4

PROCESSOR ACCEPTANCE TEST

(PAT4)

93802-90



ComputerAutomation®

Engineering Reference Number
90-93802-00A4

NAKED MINI Division
HERTFORD HOUSE
DENHAM WAY, MAPLE CROSS
RICKMANSWORTH, HERTFORDSHIRE WD3 2X

Telephone Rickmansworth 71211

Telex 9226

© 1979 Computer Automation, Inc.
Printed in U.S.A.

REVISION HISTORY

<u>Revision</u>	<u>Issue Date</u>	<u>Comments</u>
A1	August 1977	Original issue
A2	September 1977	Adds optional instruction set programs
A3	May 1978	Production release
A4	July 1979	Corrects several documentation errors. Updates manual to reflect software compatibility with OMEGA4 PLUS loader.

Computer Automation, Inc., reserves the right to make changes in specifications at any time and without notice. The information furnished by Computer Automation, Inc., in this publication is believed to be accurate and reliable. However, no responsibility is assumed by Computer Automation, Inc., for its use; nor for any infringements of patents or other rights of third parties resulting from its use. No license is granted under any patents or patent rights of Computer Automation, Inc.

Computer Automation, Inc., welcomes comments and suggestions for improvements to our manuals. Should difficulty be encountered with the information in the manual or if you have a suggestion for manual improvement, please send the comments or suggestions to Computer Automation, Inc., Technical Publications, Department 11-0715, 18651 Von Karman, Irvine, California, 92713. Please provide as much information as possible to help evaluate the comment or suggestion.

CONTENTS

SECTION 1	INTRODUCTION	1
SECTION 2	ENVIRONMENT	1
	2.1 HARDWARE REQUIRED	1
	2.2 SOFTWARE REQUIRED	2
	2.3 ADDITIONAL HARDWARE SUPPORTED	3
SECTION 3	REFERENCE DOCUMENTATION	3
SECTION 4	GENERAL DESCRIPTION	4
SECTION 5	PROGRAM OPERATION	5
	5.1 LOADING AND START UP	5
	5.1.1 Default Mode -- Address :100	5
	5.1.2 Demand Mode -- Address :101	6
	5.1.3 Restart Mode -- Address :102	7
	5.2 RUN INDICATION	7
	5.3 TERMINATION	8
	5.3.1 Normal Termination	8
	5.3.2 Error Termination	8
	5.4 CONSOLE INTERRUPT TESTING	8
	5.5 POWER FAIL/POWER UP TESTING	8
	5.6 UNEXPECTED TRAPS	9
SECTION 6	TEST DESCRIPTIONS	9
	6.1 TEST 1, INSTRUCTION DIAGNOSTIC (IDP4)	9
	6.1.1 Basic Instruction Testing	9
	6.1.2 Single Word Memory Reference Testing	9
	6.1.3 Register to Register Testing	10
	6.1.4 Shift/Rotate and Bit Manipulation Test	10
	6.1.5 Immediate Literal Testing	10
	6.1.6 Conditional Jump Testing	10
	6.1.7 Multiple Word Memory Reference Testing	10
	6.1.8 Stack Testing	10
	6.1.9 Trap Emulation Testing	10
	6.2 TEST 2, UNIMPLEMENTED INSTRUCTION TEST (PAT4)	11

6.3	TEST 3, MEMORY TEST (MEM4)	11
6.3.1	High Memory Addressability Test	12
6.3.2	Address Test	12
6.3.3	Spiral Test	12
6.3.4	Byte Test	12
6.3.5	Low Memory Test	13
6.3.6	Error Halts	14
6.4	TEST 4, REAL-TIME CLOCK TEST (RTC4)	15
6.5	TEST 5, I/O TEST (IOT4)	16
6.6	TEST 6, FLOATING POINT INSTRUCTION TEST (FPT4)	16
6.7	TEST 7, LIST INSTRUCTION TEST (LST4)	17
6.8	TEST 8, CHARACTER/NUMERIC INSTRUCTION TEST (CNM4)	17
6.9	TEST A, AUTOLOAD TEST (ALT4)	18
SECTION 7	OPERATION SUMMARY	19
7.1	DEFAULT MODE	19
7.2	DEMAND MODE	19
7.3	RESTART MODE	20
7.4	AUTOLOAD TEST (ALT4)	21
APPENDIX A	OS:LNK MEMORY MAP	21
APPENDIX B	MOVE UTILITY	24
APPENDIX C	OMEGA4 PLUS DISK UTILITY PROGRAM	24
C.1	INTRODUCTION	24
C.2	PHYSICAL DEVICE USE	25
C.3	DISK UTILITY PROGRAM COMMANDS	27
C.3.1	Designating the List Output Device: the P Command	28
C.3.2	Copying a File or an Entire Disk: the C Command	29
C.3.3	Deleting a File: the D Command	31
C.3.4	Formatting a Disk: the F Command	31
C.3.5	Labeling a Disk: the L Command	33
C.3.6	Displaying a Disk's Files: the V Command	33

C.3.7	Dumping the Absolute AUs of a Disk: the X Command	34
C.3.8	Renaming a File: the R Command	36
C.3.9	Transferring Control to Another Address or Program: the J Command	36
C.3.10	Invoking Another Program Module: the E Command	36
C.4	OMEGA4 PLUS DISK UTILITY PROGRAM COMMAND SUMMARY	37
C.5	OMEGA4 PLUS DISK UTILITY PROGRAM MESSAGE SUMMARY	37

FIGURES

<u>Figure</u>		<u>Page</u>
C-1	Standard IBM Sector-Ordering Sequences for Floppy Disks . .	32
C-2	8-Column Dump Listing	35

SECTION 1. INTRODUCTION

The NAKED MINI@4 Family Processor Acceptance Test program verifies the operational capability of NAKED MINI@4 Family computers by means of tests that check for complete instruction repertoire, memory integrity, and processor features (Power Fail, Real-Time Clock, AutoLoad, Console Interrupt, Traps).

SECTION 2. ENVIRONMENT

The minimum hardware and software requirements to run this program are listed in this section.

2.1 HARDWARE REQUIRED

The following hardware is required to run PAT4.

- LSI 4/10, LSI 4/30, or LSI 4/90 processor with Programmer's Console.
- Paper tape reader or floppy disk drive for program loading.
- I/O Test Module (Model Number 14547-00). Required only for Test 5 (IOT4) and Test 8 (NM4).
- Memory as follows for paper tape or floppy diskette (Minimum of 16K with diskette)

PROGRAM	MEMORY TEST RELOCATING	MEMORY REQUIRED
PAT410 } PAT430 } PAT490 }	N	8K
	Y	16K
PAT410 OPT	N	16K
	Y	32K
PAT430 OPT } PAT490 SCI } PAT490 BUS }	N	24K
	Y	48K

2.2 SOFTWARE REQUIRED

The following PAT4 program software is required, depending on the instruction set.

Paper Tape

	<u>Engineering Number</u>	<u>Marketing Number</u>
• LSI 4/10 basic	00-93806-40	63-45018-10
• LSI 4/10 optional	00-93806-43	63-45018-20
• LSI 4/30 basic	00-93806-42	63-45018-30
• LSI 4/30 optional	00-93806-45	63-45018-40
• LSI 4/90 basic	00-93806-44	63-45018-50
• LSI 4/90 scientific (floating point and list)	00-93806-47	63-45018-60
• LSI 4/90 business (character/numeric)	00-93806-48	63-45018-70

NAKED MINI 4 PROCESSOR ACCEPTANCE TEST (PAT4)

90-93802-00

Floppy Diskette

	<u>Engineering Number</u>	<u>Marketing Number</u>	<u>Filename</u>
• LSI 4/10 basic	00-93806-02	63-45018-11	PAT410BASIC.BIN
• LSI 4/10 optional	00-93806-03	63-45018-21	PAT410OPT.BIN
• LSI 4/30 basic	00-93806-04	63-45018-31	PAT430BASIC.BIN
• LSI 4/30 optional	00-93806-05	63-45018-41	PAT430OPT.BIN
• LSI 4/90 basic	00-93806-06	63-45018-51	PAT490BASIC.BIN
• LSI 4/90 scientific (floating point and list)	00-93806-07	63-45018-61	PAT490SCI.BIN
• LSI 4/90 business	00-93806-08	63-45018-71	PAT490BUS.BIN

2.3 ADDITIONAL HARDWARE SUPPORTED

The following hardware is optional.

- Additional memory, up to 64K
- ASR-33 TTY or equivalent (to use DEBUG4)
- Processor option card

SECTION 3. REFERENCE DOCUMENTATION

Refer to the following documents for supplementary information.

	<u>Engineering No.</u>	<u>Marketing No.</u>
• <u>NAKED MINI 4 Assembler User's Manual</u>	90-93500-01	63-93500-80
• <u>LSI 4/10 Computer Reference Manual</u>	91-20990-00	63-20990-91
• <u>LSI 4/30 Computer Reference Manual</u>	91-20991-00	63-20991-91
• <u>LSI 4/90 Computer Reference Manual</u>	91-20945-00	63-20945-91

SECTION 4. GENERAL DESCRIPTION

The basic PAT4 is comprised of six separate tests assembled in three parts:

Part 1	Test 1	Instruction Diagnostic (IDP4)
Part 2	Test 2	Unimplemented Instruction Test (UIT4)
	Test 3	Memory Test (MEM4)
	Test 4	Real-Time Clock Test (RTC4)
	Test 5	I/O Test (IOT4)
Part X	Test A	AutoLoad Test (ALT4)

Depending on which version of PAT4 is used, one or more of the following tests may also be included:

Test 6	Floating point instruction test (FPT4)
Test 7	Lists instruction test (LST4)
Test 8	Character/numeric instruction test (CNM4)

The separately assembled parts are linked together by the OS:LNK utility. The tape or diskette is delivered with an OS:LNK memory map which lists the starting location of each part, as it resides in memory after loading. On the supplied listing, only Part 1 shows addresses corresponding to the test in memory. Part 2 and Part X addresses are offset by a value which can be obtained from the OS:LNK map.

For example, the Part 2 assembly shows :0000 as Part 2's starting address, while the memory map shows :1000 as the starting location of Part 2 when loaded in memory. Therefore, if a Test 2 error halt (see Section 5.3.2) occurs with the P register contents :1033, the corresponding assembly listing address is found by subtracting :1000 from :1033. In this example the error halt corresponds to listing address :0033. This process can be reversed to examine the contents of a listing in memory. To examine the memory contents of listing location :0033 in this example, add :0033 to starting location :1000, and examine memory location :1033.

PAT4 includes the utility programs DEBUGM.BIN, DEBUGS.BIN, and DEBUGX.BIN which allow the user to inspect, print, and modify any memory location. Refer to the memory map for the starting location of DEBUG4. When PAT4 is delivered in diskette form, it also contains the OMEGA4 PLUS Disk Utility Program.

Appendix A contains a sample memory map and label definitions.

SECTION 5. PROGRAM OPERATION

5.1 LOADING AND START UP

PAT4 has three modes of entry: default mode at location :100, demand mode at location :101, and restart mode at location :102. Test A, ALT4, cannot be run at any of these addresses. Refer to Section 6.9 for Test A start-up procedures.

5.1.1 Default Mode -- Address :100

Loading and starting procedures for PAT4 vary depending on whether the program comes in paper tape or diskette form.

5.1.1.1 Paper Tape

AutoLoad the tape. It will load and then branch to begin execution at location :100. Tests 1 through 4 run continuously. The A register run options are set to zero.

5.1.1.2 Diskette

When PAT4 is delivered in diskette form, it must be loaded using the OMEGA4 PLUS Disk Utility Program, which resides as a file on the product diskette. AutoLoading the Disk Utility Program generates a TTY display of the form

```
OMEGA DISK UTILITY (CO):5DA0-:7DFE
```

The user should issue an E command to load and execute PAT4 via its file name. Filenames are listed in Section 2.2. The E command should take a form similar to the following:

```
?E,DF.PAT410BASIC.BIN,0,5000
```

Because this version of PAT4 must be loaded at location :0, the user must specify both the load address for PAT4 (0), and the load address for PAT4's bootstrap program (e.g., 5000). Keeping in mind the extents of the Disk Utility Program and the size of PAT4, the user should specify a load address for the bootstrap which falls between them. The size of PAT4 is indicated in the diskette's view listing; the program's bootstrap occupies approximately :300 words of memory. Note that the Disk Utility Program is loaded automatically into the high end of memory. For more information on the OMEGA4 PLUS Disk Utility Program, see Appendix C of this manual.

After loading, PAT4 branches to location :100 and begins execution. At this location, Tests 1 through 4 are run continuously. The A register run options are set to zero.

5.1.2 Demand Mode -- Address :101

The program can be started at location :101 if specific tests or run options are desired. Test 1, IDP4, is always run, regardless of test selection.

If PAT4 is not loaded, proceed with Default mode (Section 5.1.1) load instructions. If the diskette form is used, PAT4 will begin execution after the E command. Press STOP to terminate the program, then proceed as follows:

1. Enter the following information in the Console Data Register before starting the program at this location:

<u>Bit</u>	<u>Value</u>	<u>1=</u>
0	--	Not used
1	:02	Run Test 2 (UIT4)
2	:04	Run Test 3 (MEM4)
3	:08	Run Test 4 (RTC4)
4	:10	Run Test 5 (IOT4)
5	:20	Run Test 6 (FPT4) if included
6	:40	Run Test 7 (LST4) if included
7	:80	Run Test 8 (CNM4) if included
8-15	--	Not used

For example, to run Tests 3, 4, and 6, enter the value :002C into the Console Data Register.

2. Enter :101 into the P register.
3. Enter the following run options in the A register.

<u>A Register Bit</u>	<u>Value</u>	<u>Run Option</u>
0	:0001	1= Bypass relocation and low memory testing during Test 3.
	:0000	0= Relocate PAT4 and perform low memory testing during Test 3 (MEM4). (See Section 6.3.5.)
1	:0002	1= Do not halt the program if power up and power down counters are unequal. (See Section 5.5.)
	:0000	0= Halt if counters are unequal.
2-7	---	Not used, enter zero

8-14	:xx00	Enter the memory size (in increments of 1K) as a binary number. For example, if 32K of memory is used, bits 8-14 contain 0100000 (i.e., :20). If bits 8-14 contain all zeros, PAT4 automatically assumes that the memory size calculated in Test 3 is correct.
15	--	Not used, enter zero.

For example, to bypass relocation and low memory testing during Test 3 on a system with 24K words of memory, enter the value :1801 into the A register.

4. Press RUN after entering A Register and Data Register information. PAT4 executes IDP4 and selected tests continuously.

5.1.3 Restart Mode -- Address :102

Restarting PAT4 at location :102 does not reset the pass counters and powerfail counters. To restart PAT4:

1. Enter the Console Data Register test selections and A register run options (see Section 5.1.2).
2. Enter :102 into the P register.
3. Press RUN to begin execution.

5.2 RUN INDICATION

During execution, bits 0-7 of the Console Data Register (i.e., the two right-most hexadecimal digits) normally display the Test number value as specified in Section 5.1.2. Refer to the specific test description for detailed run indication descriptions.

5.3 TERMINATION

5.3.1 Normal Termination

Press the RUN/STOP switch to terminate PAT4. (During MEM4, reverse the state of the SENSE switch before pressing RUN/STOP.) The program can be subsequently restarted at location :102.

5.3.2 Error Termination

On any error condition, PAT4 halts and the Programming Console hexadecimal display contains "IDLE." To display the Console Data Register contents, press the RUN/STOP switch once. Examine the P register and refer to the program listing to determine the cause of the error.

The Console Data Register displays error status during a Test 3 (MEM4) error halt only. Refer to Section 6.3.6 for error status description.

5.4 CONSOLE INTERRUPT TESTING

Press INT during PAT4 execution to test console interrupt recognition. The Console Data Register flashes an alternating :AAAA/:5555 pattern and the OV indicator remains on for approximately three seconds. If INT is pressed when interrupts are disabled, the INT indicator remains lit until interrupts are enabled. The alternating pattern is displayed when the interrupt is serviced.

5.5 POWER FAIL/POWER UP TESTING

The contents of the A, Q, X, Y, K, L, and S registers, the Console Data Register, and the Sense Switch status are saved and the power down count (PDNCT) is incremented if a power failure occurs.

When power is restored, the power up count (PUPCT) is incremented, the contents of the volatile registers are restored, and if test 1,2, or 3 was executing, program execution is resumed at the location where the power failure occurred. If test 4 through 8 was executing, PAT4 is restored at the beginning of IDP4.

An error halt occurs if PDNCT and PUPCT are not equal unless this halt was suppressed (A register run option bit 1=1) prior to PAT4 execution. Press RUN following this error halt to resume execution at the beginning of IDP4. The three halts (see Section 6.1) are bypassed.

5.6 UNEXPECTED TRAPS

PAT4 halts immediately if unexpected memory exception, unimplemented instruction, arithmetic error, character/numeric exception, or stack exception traps occur. Press the RUN/STOP switch once to examine the Console Data Register. On an unexpected trap, bits 8-15 display :AA; bits 0-7 remain unchanged. Press RUN again to restart PAT4. IDP4 begins execution, although the three initial halts (see Section 6.1) are bypassed.

SECTION 6. TEST DESCRIPTIONS

6.1 TEST 1, INSTRUCTION DIAGNOSTIC (IDP4)

IDP4 exercises all non-I/O instructions in the NAKED MINI 4 instruction repertoire and determines whether each instruction performs as expected, although it does not check the effects of instruction execution on uninvolved registers or memory cells.

During IDP4, bits 0-7 of the Console Data Register display :01.

The testing is conducted in the following order.

6.1.1 Basic Instruction Testing

This part of the IDP4 test verifies basic instructions which are required to extensively test remaining instructions. Testing is performed using absolute direct addressing, word mode only.

Three halts occur at the beginning of this test, during the first pass of IDP4. Press RUN after each halt to verify the Console Data Register contents and press RUN again to continue. All halts after Halt 3 are error halts.

<u>HALT</u>	<u>ACTION</u>
Halt 1	Console Data Register should display :XX14.
Halt 2	Console Data Register should display :XX17.
Halt 3	Console Data Register should display a machine type indicator of the form :000X. This indicator is produced by execution of a Fetch Machine Type (FMT) instruction.

6.1.2 Single Word Memory Reference Testing

Single word memory reference instruction testing is performed after basic instruction testing. Extensive testing is performed on COPY and EXCH in all memory reference modes; other instructions are tested less extensively.

6.1.3 Register to Register Testing

The COPY instruction is tested with all register pair combinations; other instructions are tested with one register pair. The OV bit is checked following the execution of a NEG, ADD, or SUB instruction.

6.1.4 Shift/Rotate and Bit Manipulation Test

Single register shift/rotates are tested with all registers. All shift counts (1-16) are tested with the A register. Double register shifts/rotates are tested with register pairs AQ, AX, and AY.

Bit manipulation instruction SBIT is tested on register bits 0-15; all other bit manipulation instructions are tested on one bit only.

6.1.5 Immediate Literal Testing

Each instruction is tested with one register. The OV bit is checked following the execution of an ADD or SUB instruction.

6.1.6 Conditional Jump Testing

Each condition which causes or does not cause a jump is verified.

6.1.7 Multiple Word Memory Reference Testing

All memory reference modes are checked on the ADDC instruction; other instructions are tested less extensively. The OV bit is checked following execution of an ADDE, SUBE, IMSE, or DMSE instruction; the OV and CY bits are checked following execution of an ADDC or SUBC instruction. MUL and DIV are tested for the four combinations of positive and negative arguments. Arithmetic exception traps and stack exception traps (JSKE) are also tested.

6.1.8 Stack Testing

Instructions are tested on all aspects of correct operation. Stack overflow trap operation is verified for JSK and PUSH. Register effects are checked for PUSH and POP.

6.1.9 Trap Emulation Testing

STRAP and UTRAP instructions are verified.

Each pass of IDP4 tests each of the above instruction groups eight times.

6.2 TEST 2, UNIMPLEMENTED INSTRUCTION TEST (UIT4)

In this test, all unimplemented instructions are executed to verify the operation of the Unimplemented Instruction Trap.

UIT4 uses the fetch machine type (FMT) instruction to determine whether or not the optional instruction set is installed, and selects the appropriate set of unimplemented instruction codes accordingly.

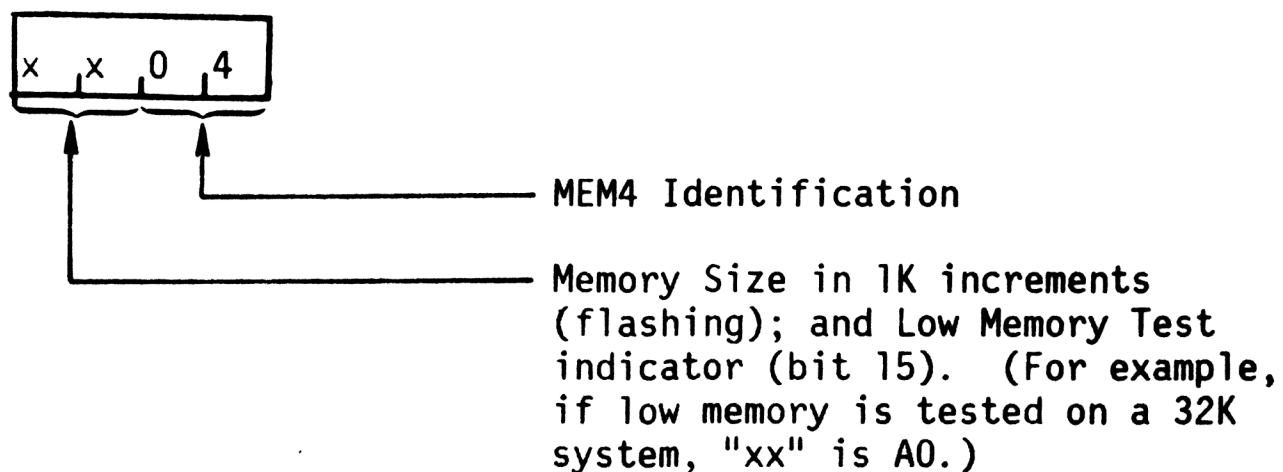
During UIT4, bits 0-7 of the Console Data Register display :02 and bits 8-15 display the upper 8 bits of the unimplemented instruction currently tested.

6.3 TEST 3, MEMORY TEST (MEM4)

The memory test verifies the integrity of the memory used in the NAKED MINI 4 configuration. The test consists of five parts:

1. High Memory Addressability Test
2. Address Test
3. Spiral Test
4. Byte Test
5. Low Memory Test

During MEM4, the Console Data Register displays:



6.3.1 High Memory Addressability Test

The highest read/write location is found. If PAT4 was entered at location :100, the calculated memory size is assumed to be correct. If PAT4 was entered at location :101 or :102, the calculated memory size is compared to bits 8-14 of the A register run options entered prior to execution (see Section 5.1.2). The test halts if a miscompare occurs. Press the RUN/STOP switch twice to continue execution. (The memory size entered in the A register is assumed to be correct).

NOTE

If A register bits 8-14 are entered as all zeroes, the calculated memory size is assumed to be correct.

The program then attempts to access location :8001 in two different ways. The first access attempts to store a register (using post-indexed addressing); the second access attempts to load a register (using single level indirect addressing with XA bit =1). If the system contains 32K or less of memory, both accesses should cause an Uninstalled Memory Trap. If the system contains more than 32K of memory, the contents of memory location :8001 are verified. Location :1 is checked for possible alteration (due to wrap-around addressing) in all memory configurations.

6.3.2 Address Test

The value :0000 is written into each location of unused memory and verified. The address of each location is then written into that location and verified.

6.3.3 Spiral Test

A rotating bit pattern (a one bit is rotated through all bit positions of a word) is written and verified for each location of unused memory.

6.3.4 Byte Test

The pattern :55AA is written into each word location of unused memory, and then verified one byte at a time.

6.3.5 Low Memory Test

The length of the PAT4 program is compared to the total amount of memory in the system. PAT4 then relocates itself to the next highest 4K boundary if sufficient memory is available; otherwise the Low Memory Test is skipped. The Address Test, Spiral Test, and Byte Test are then performed on locations :00 up to the starting location of the relocated PAT4 program.

NOTE

1. Locations :80 through :FF are excluded from testing. The Power Fail, Memory Exception, and Unimplemented Instruction Traps remain active throughout MEM4.
2. Console interrupts are disabled during the Low Memory Test. Interrupts are not serviced while PAT4 is relocated.

PAT4 relocates itself back to its original position after low memory testing is completed.

The operator can bypass the Low Memory Test by setting bit 0 of the A register run options (see Section 5.1.2).

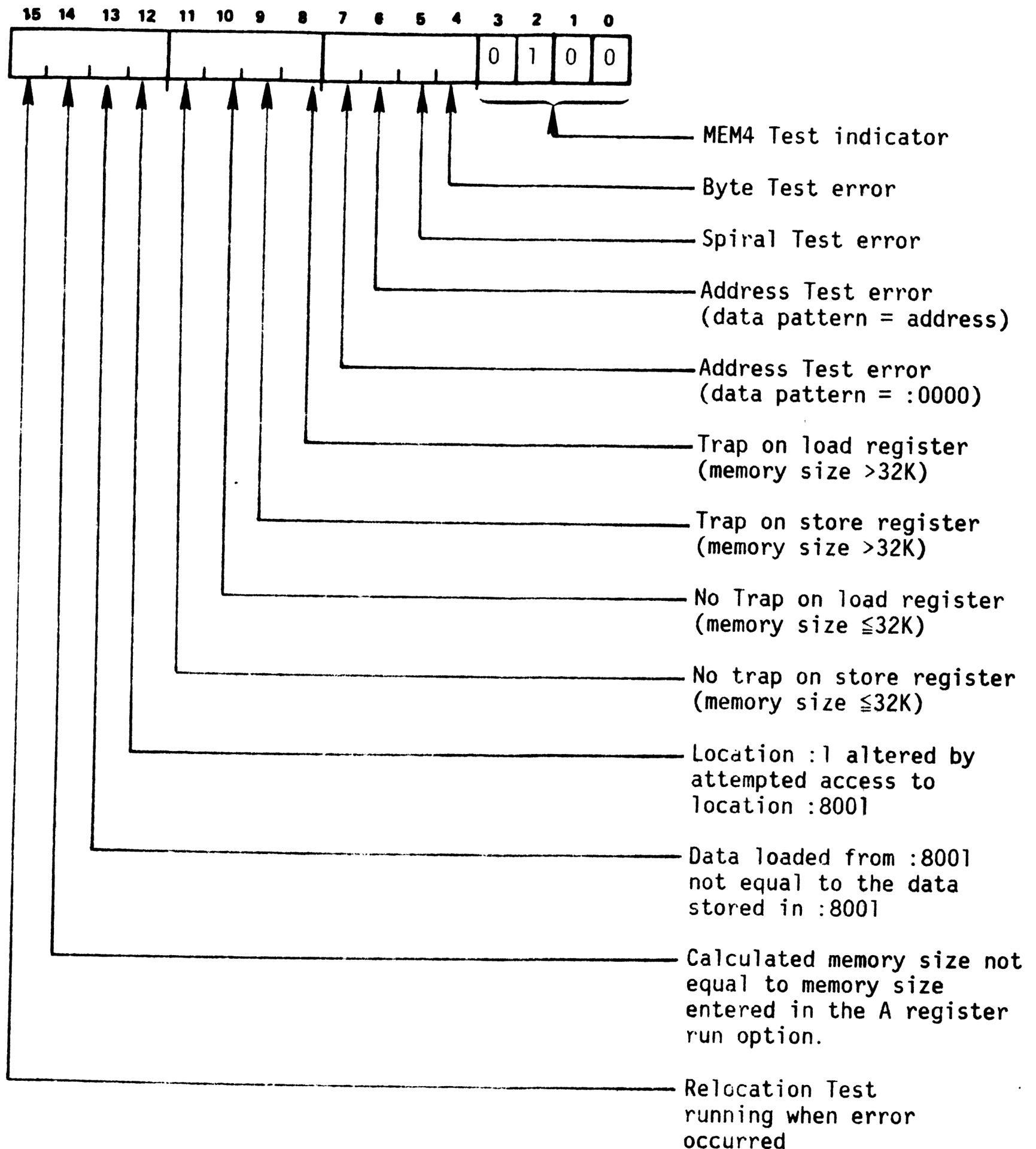
MEM4 terminates following the Low Memory Test. However, MEM4 can be terminated at any time by reversing the state of the SENSE Switch. PAT4 immediately skips to the next test in sequence.

NOTE

Do not press the RUN/STOP switch while MEM4 is executing. To halt the processor during MEM4, reverse the state of the SENSE Switch, then press RUN/STOP. Failure to follow this procedure may require reloading the PAT4 program.

6.3.6 Error Halts

Error halts during MEM4 load the following information in the Console Data Register. (A "1" in any bit position corresponds to the following status.)



The contents of the A, Q and X registers contain additional information.

Console Data Register

If any of bits 4-7 =1:
(i.e., the second hexadecimal
digit from the right is
non-zero)

A = actual data transferred
Q = expected data transferred
X = memory location where miscompare
occurred

If bits 12 or 13 =1:

A = actual location contents
Q = expected location contents

If bit 14 =1:

A = calculated memory size
Q = memory size entered in the A register
prior to execution

6.4 TEST 4, REAL-TIME CLOCK TEST (RTC4)

RTC4 verifies that the Real-Time Clock can generate timing ("tick") interrupts and checks that RTC interrupts occur when the increment location count reaches zero. Increment location :91 is preset to a two-second count. RTC4 monitors location :91 until the count reaches zero.

An error halt occurs under any of the following conditions:

1. The timing loop expires before a timing interrupt ("tick") occurs.
2. Increment location :91 is incremented by a value other than 1.
3. A RTC interrupt is not taken when location :91 is zero.
4. A RTC interrupt is taken when location :91 is not zero.
5. Increment location :91 is not -1 immediately prior to the RTC interrupt.

RTC4 also performs a Real-Time clock consistency test. The RTC is run for about .25 second and an average 2-tick software count is taken. A $\pm 6\%$ window is calculated on this software count, and used to determine the consistency of the RTC over a 4-second span.

During RTC4, bits 0-7 of the console data register display :08. During the first part of RTC4, the OV indicator blinks (about 2 seconds), and during the second part of RTC4, bits 12-15 of the console data register display a counter incrementing from 0 to F (about 4 seconds).

6.5 TEST 5, I/O TEST (IOT4)

IOT4 tests all input/output functions of the processor. In addition, it verifies all motherboard signals and features of processor operation associated with interrupts.

NOTE

An I/O Test Module is required for this test.

The individual tests in IOT4 are:

1. Device Address Function Code Exercise
2. Programmed I/O Instruction testing
3. I/O clocks testing
4. Interrupt instruction testing
5. Auto I/O instruction testing
6. Instruction non-interruptability testing
7. DMA testing
8. Power-Fail testing

During IOT4, bits 0-7 of the console data register display :10. Bits 8-15 display :0F for about .25 second, and then :FF for about .25 second. (The test lasts about .5 seconds.)

6.6 TEST 6, FLOATING POINT INSTRUCTION TEST (FPT4)

This test verifies the correct operation of the floating point class of instructions, plus FIX and FLOAT (console and control class). The test checks for correct results with a wide range of operands for each instruction tested. The addressed memory locations and the X and Y registers are tested for side effects. Arithmetic error trap operation is checked. Only relative addressing mode is tested.

During the FPT4, bits 0-7 of the console data register display :20. The test lasts approximately .5 second.

NOTE

If the floating point test is run without the floating point option installed, an unimplemented instruction trap results.

6.7 TEST 7, LIST INSTRUCTION TEST (LST4)

This test verifies the correct operation of the list instruction. SEARCH is tested with all search conditions and all list structures. INSERT and DELETE are tested with absolute and relative linked list structures. Both one and two word INSERT and DELETE instructions are tested. Only relative list descriptor block addressing and relative head pseudo node addressing are tested.

During LST4, bits 0-7 of the console data register display :40. The test lasts approximately .5 second.

NOTE

If the list instruction test is run without the lists option installed, an unimplemented instruction trap results.

6.8 TEST 8, CHARACTER/NUMERIC INSTRUCTION TEST (CNM4)

This test verifies the correct operation of the character/numeric class of instruction.

NOTE

An I/O Test Module is required for this test.

A wide range of strings are used to test each instruction. The character/numeric exception trap is tested and the following string descriptor block addressing modes are tested:

- absolute direct
- absolute direct pre-indexed
- absolute direct pre-post-indexed
- absolute indirect
- absolute indirect pre-post-indexed
- relative direct forward
- relative direct backward
- relative direct backward post-indexed
- relative indirect backward
- relative indirect backward post indexed

The following string addressing modes are tested:

absolute
absolute post indexed
relative

CNM4 requires that an I/O test module is installed in the system. If the test is run without an I/O test module, those sections which do not require it are executed. When the parts which require it are reached, two error halts occur. Press RUN/STOP to continue PAT4 execution.

NOTE

If the character/numeric test is run without the character/numeric option installed, an unimplemented instruction trap results.

6.9 TEST A, AUTOLOAD TEST (ALT4)

This test verifies the proper loading of the AutoLoad program from the AutoLoad ROM. This test cannot be entered from address :100, :101, or :102. If PAT4 is not loaded, proceed with Default mode (Section 5.1.1) load instructions. If the diskette form is used, PAT4 will begin execution after the E command. Press STOP to terminate the program, then proceed as follows:

1. Enter the starting address of ALT4 (available on the OS:LNK memory map) in the P register and press RUN to begin execution. Memory locations :00-:FF are copied to another 256-word area of memory. The processor then halts with :FFFF in the Console Data Register. Press RUN/STOP to display the Console Data Register value.
2. Set the SENSE switch to off to ensure that the AutoLoad program will be loaded but not executed.
3. Press AUTO and then RUN/STOP. The processor halts with the AutoLoad program in memory locations :00-:FF.
4. Press RUN/STOP and then enter address ALTRTN (available on the OS:LNK memory map) in the P register and press RUN/STOP again.

The AutoLoad program is then moved to the AutoLoad Test Buffer ALTBUF (address available on the OS:LNK memory map) and the original contents of :00-:FF are restored. ALT4 computes the checksum of the AutoLoad program instructions and loads it into the A register. If the checksum is correct, :FF00 is loaded in the Console Data Register; an incorrect checksum causes :00FF to be loaded. To display this value, press RUN/STOP.

If PAT4 was executed at :100, :101, or :102 prior to the AutoLoad Test, the prior test sequence can now be resumed by pressing RUN/STOP again.

NOTE

Interrupts and traps are inoperable during the AutoLoad test. If a power fail, interrupt, or trap occurs during ALT4, PAT4 must be reloaded to restore locations :00-:FF. Do not test Console interrupt recognition during ALT4.

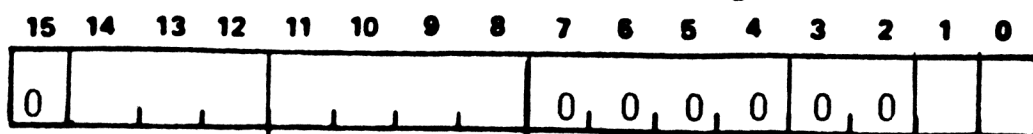
SECTION 7. OPERATION SUMMARY

7.1 DEFAULT MODE

1. Load and execute the PAT4 program at address :100. IDP4, UIT4, MEM4 and RTC4 are executed continuously. (Press RUN/STOP twice after each of the three halts during the first pass of IDP4.)
2. Press RUN/STOP to terminate execution (During MEM4, reverse the state of the SENSE switch before pressing RUN/STOP). Press RUN/STOP twice to continue execution following any error halts.

7.2 DEMAND MODE

1. Load the PAT4 program and enter address :101 in the P register.
2. Enter run options in the A register as follows:

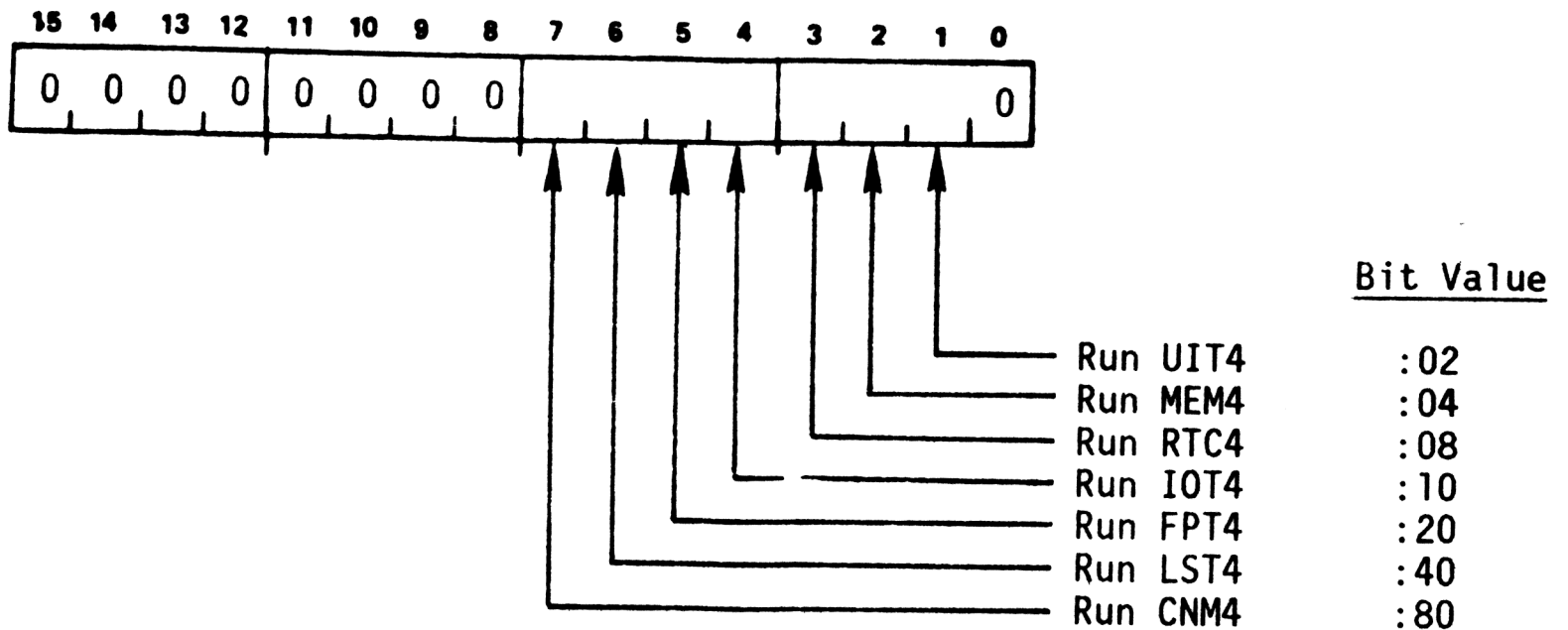


1 = Bypass MEM4 Relocation Test
0 = Run MEM4 Relocation test

1 = Do not halt if PDNCT and PUPCT unequal
0 = Halt if PDNCT and PUPCT unequal

Enter binary memory size ($\leq 64K$)

3. Set Console Data Register bits corresponding to desired tests, as follows:



4. Press RUN/STOP. IDP4 and selected tests are run continuously. (Press RUN/STOP twice after each of the three halts during the first pass of IDP4).
5. Press RUN/STOP to terminate execution (During MEM4, reverse the state of the SENSE switch before pressing RUN/STOP). Press RUN/STOP twice to continue execution following any error halts.

7.3 RESTART MODE

1. Load the PAT4 program and enter address :102 in the P register.
2. Follow steps 1 through 5 as described in DEMAND MODE. Pass counters and power fail counters remain accumulative at this address.

7.4 AUTOLOAD TEST (ALT4)

1. Load the PAT4 program, enter link map address ALT4 in the P register, and press the RUN/STOP switch.
2. When "IDLE" appears on the console display, press RUN/STOP. The Console Data Register displays :FFFF.
3. Turn SENSE off.
4. Press AUTO and then RUN/STOP.
5. When "IDLE" appears, press RUN/STOP.
6. Enter link map address ALTRTN in the P register. Press RUN/STOP,
7. When "IDLE" appears, press RUN/STOP. The Data Register displays :FF00 if the test was successful, :00FF if the test failed.
8. Press RUN/STOP if PAT4 was previously executed at address :100, :101, or :102 to resume execution of other PAT4 tests.

APPENDIX A. OS:LNK MEMORY MAP

An OS:LNK memory map is delivered with each PAT4 program tape. This map lists each label in the program which has been used in a NAM directive, along with its actual address in memory when loaded. The following page shows a sample memory map.

MEMORY MAP LABEL DEFINITIONS

COUNT1	Bits 0-15 of a 32-bit pass counter
COUNT2	Bits 16-31 of a 32-bit pass counter
PUPCT	Power up count
PDNCT	Power down count
PFERRS	Power up/down miscompare count
A01000	Starting address of IDP4 (first pass)
A02000	Starting address of IDP4 (subsequent passes)
PART2	Starting Address of PART2 (Contains UIT4, MEM4, RTC4, IOT4)
PARTX	Starting Address of PARTX (Contains MOVE, ALT4)
FLTPT1 - FLTPT9	Starting Addresses of Floating Point Segments
LISTS1 - LISTS9	Starting Addresses of Lists Segments
CNUM01 - CNUM17	Starting Addresses of Character/Numeric Segments ALT4
ALTRTN	Starting address of ALT4
ALTRTN	Second address loaded by the operator during ALT4 execution
ALTBUF	Starting address of buffer containing the AutoLoad program during ALT4
MOVE	Starting address of Move utility
DEBUG4	Starting address of the DEBUG4 utility
PATEND	Highest PAT4 location



PAGE 1 05/02/78 15:06:36 NS:LNK (R2) MEMORY MAP

CREATED FILE RR0235

MISSING

G:0	G:1	G:2	G:3	G:4	G:5	G:6	G:7	G:8	G:9	G:A	G:M
G:C	G:D	G:E	G:F	G:G	G:H	G:I	G:J	G:K	G:L	G:N	G:P
G:O	G:P	G:Q	G:R	G:S	G:T	G:U	G:V	G:W	G:X	G:Y	G:Z

PROGRAM

COUNT1 002C	COUNT2 002D	PUPCT	PDNCT 00EE	PFFRRS 00FF	A01000 0113
A0200 0120	PART2 1000	FLTP1 1800	FLTP2 1A00	FLTP3 1C00	FLTP4 1F00
FLTP5 2000	FLTP6 2200	FLTP7 2400	FLTP8 2600	FLTP9 2900	LIST1 2C00
LIST2 2F00	LIST3 3200	LIST4 3500	LIST5 3800	LIST6 3B00	LIST7 3E00
LIST8 4100	LIST9 4400	PARTX 4700	MUVF 4740	ALT4 4780	ALTM 4790
ALBUF 4800	DERUG4 4900	MDRUG4 4900	PATFND 4CFA		

MEMORY USAGE

SCRATCH-PAD PROGRAM	0000-007H
MAIN MEMORY PROGRAM	00R0-4CF9
EXEC ADDRESS	0100

SCRATCHPAD USAGE TABLE:

ADDR	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
0010	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
0020	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
0030	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
0040	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
0050	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
0060	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
0070	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P

LEGEND:

A=ABSOLUTE LITERAL
R=BYTE RELOCATABLE LITERAL
P=ABSOLUTE PROGRAM
R=WORD RELOCATABLE LITERAL
S=SRCL PROGRAM
W=WORD RELOCATABLE INDEX POINTER
X=ABSOLUTE INDEX POINTER
Y=BYTE RELOCATABLE INDEX POINTER

PROCESSED LSI 3 OBJECT

NO ERRORS

APPENDIX B. MOVE UTILITY

A general-purpose move routine is included in PAT4. Operation is as follows.

1. Enter register values:

Q register - length of code segment to be moved

X register - starting address of code segment to be moved

Y register - starting address of destination area

2. Enter the starting address of the move routine in the P register.
3. Press the RUN/STOP switch.

After the transfer has completed, the move routine verifies the transfer. If an error is encountered during the verify, the program halts with :BAD in the Console Data Register. The X register contains the read pointer, and the Y register contains the write pointer. Press RUN/STOP to display the Console Data Register value first. Then, display the X and Y register values if the Console Data Register contains :BAD. Press RUN/STOP to continue the verification. At the end of the verification, the program halts with :C000 in the Console Data Register.

APPENDIX C. OMEGA4 PLUS DISK UTILITY PROGRAM

C.1 INTRODUCTION

The Disk Utility Program is included as a file on IECD diskette; it provides commands to perform the functions listed below:

- Copy a disk file or an entire disk to another file or device
- Delete a file from a disk
- Format a disk to prepare for labeling it
- Label a disk to be used with either OS4 or OMEGA4 PLUS
- Display the current contents of the disk directory on the current list output device
- Dump (in hexadecimal notation) the contents of a disk
- Rename a file

- Load and execute a program module
- Transfer control to another address or a program
- Designate the list output device

Disks generated by OS4 or the OMEGA4 PLUS Disk Utility Program may be used under both OMEGA4 PLUS and the Real-Time Executive (RTX)-based operating system OS4. OS4-labeled disks, however, may not be used to hold programs to be executed via the OMEGA4 PLUS E command. This exception is due to the fact that the bootstrap code, which is written to a disk during the labeling operation, does not operate in the same way or provide the same facilities under OS4 as under OMEGA4. Programs to be executed via the OMEGA4 PLUS E command, therefore, either must be loaded using an OMEGA4-labeled disk or copied to one before execution is attempted. In all other respects, an OS4-labeled disk is identical to an OMEGA4-labeled disk.

C.2 PHYSICAL DEVICE USE

The OMEGA4 PLUS Disk Utility Program reads input from and writes output to several physical devices. To associate an input or an output operation with a particular device, the user normally identifies the physical device in a command line by specifying its device name. Because the device specified is determined by the device's function within the system as well as by the actual name of the device, the mnemonic value of the device name is lost in many cases. For example, CR may be used to identify a TTY paper tape punch for output, but this is not clear from the name "CR."

Device names, descriptions, and device addresses are given below. It is recommended that the device specifications included in command lines be consistent with the required device addresses, and vice versa.

- TY refers to the interactive terminal used to enter commands. List output goes to the printer or screen portion of the terminal and all other output goes to the paper tape (or cassette) portion of the terminal. All input (except command input) is read without echoing from the paper tape (or cassette) input portion of the terminal. The device address used is :F8.
- TX refers to the paper tape or cassette input and output portions of the TTY. All input is read (without echoing) from the reader and all output is directed to the punch. The device address used is :F8.
- TV refers to the TTY or CRT. All input is read from the keyboard; all output is written to the printer or screen. All input is echoed, but no line-feeds are appended to returns; these must be explicitly entered. The device address used is :F8.

- PT refers to the high-speed paper tape reader for input and the high-speed paper tape punch for output. Either or both of these devices may be replaced with RS-232 or interactive terminals emulating the reader and punch if no high-speed paper tape reader or punch is included in the system. The device addresses used are :FA for the reader and :FC for the punch. The card reader may be reassigned to the paper tape reader input address, but assignment to PT is not the same as assignment to the changed CR.
- PR is used identically to PT except that the device address of :FA is used for both the reader and the punch.
- PP is used identically to PT except that the device address of :FC is used for both the reader and the punch.
- LP refers to either a line printer or an RS-232-compatible interactive device that emulates line printer operations. If an RS-232-compatible device is used, it must not perform automatic line feeds. If LP is specified as an input device, input is taken from a high-speed paper tape reader or TTY paper tape reader. The device address used is :FE.
- CR refers to a card reader for input and to a TTY paper tape punch or TTY printer for output. The data read is in the form of 80-column records in standard ASCII-Hollerith form. The input is processed as standard Hollerith punched card codes, with all EBCDIC codes permuted to one 8-bit ASCII code. The device address normally used with the CR device name is :FO. When an LSI 4/10 processor is used, a different device address derived from patching is used.
- DFnn refers to a floppy disk mounted in drive unit nn. The unit number may be any value from 0-3, with or without a leading zero. If the value is zero, the unit number may be omitted altogether. The device address used with the floppy disk controller is :88 (or :11 if a five-bit device address).
- DFnn.filename.extension

refers to a file named filename on a floppy disk mounted in drive unit nn. The unit number may be any value from 0-3, with or without a leading zero. If the value is zero, the unit number may be omitted altogether. The file name is a string of one to 11 alphanumeric characters, the first of which is always alphabetic.

A file name extension is a four-character string (a period followed by three letters) appended to the file name to reflect the type of information the file contains. The file name extensions are:

- .ASM - Assembly language source text file
- .MAC - Assembly language Definition File
- .FTN - FORTRAN source text file
- .BAK - Source backup file
- .TXT - Text file
- .LST - Listing file
- .OBJ - Object file
- .BIN - LINK output file
- .LIB - Library file
- .JCL - Job file
- .DAT - Data file
- .HLP - Documentation file
- .Sxx - (where "x" denotes a letter) System file

The programmer can assign the file name and extension when he creates the file. While it is not mandatory that he assign file name extensions according to the conventions listed above, it is recommended that he do so. It is also recommended that the programmer avoid assigning file names which conflict with names having a defined meaning in OS4: command words, logical unit names, and the like.

File name extensions allow related files to share a common file name. For example, the programmer might create the following set of files:

- PROG.ASM (containing an assembly language source program)
- PROG.BAK (containing an assembly language source backup of PROG.ASM)
- PROG.OBJ (containing the object code produced by assembling PROG.ASM and PROG.MAC)
- PROG.LST (containing the assembly listing produced by the assembly)
- PROG.BIN (containing PROG.OBJ linked with other object modules)

OMEGA4 accepts one additional device name in special situations; its use is treated in more detail later in this appendix.

DU refers to a simulated dummy file. When read from it always returns an end-of-file (EOF) condition, and when written to it always accepts data and returns a no-error condition.

C.3 DISK UTILITY PROGRAM COMMANDS

The programmer can issue any of the following commands in response to the Disk Utility Program's command prompt, a question mark (?). These commands are summarized at the end of this appendix. If the Disk Utility Program detects an error in a user command, it generates one of the error messages described at the end of this appendix.

C.3.1 Designating the List Output Device: the P Command

Format: P { [Wwidth][Llength],device
 C[Wwidth][Llength][,device]
 D[Wwidth][Llength][,device] }

The Disk Utility Program provides commands which generate list output, such as the commands to generate view listings and dumps. Before entering one of these commands, the programmer can issue a P command to specify which device is to receive the listing.

Entering the P command with the device parameter included serves to direct list output generated by the Disk Utility Program to "device." The device parameter is one of the physical device names listed in Subsection C.2, including a floppy disk file.

When a line printer is specified as the device, the C or D option may be included in the command line. Including the C option in the P command line causes only a carriage return to be generated before a new line is printed. The C option typically is included when a Centronics line printer is used, or when any printer that generates an internal line feed in response to a carriage return is used. Including the D option causes both a carriage return and a line feed to be generated before each new line is printed. The D option typically is included when a Dataproducts line printer or any other printer requiring an explicit line feed is used. When neither option is included and the line printer is specified as a device, the system defaults to generating only a carriage return before printing a line.

If a paper tape or disk file is specified as the device, list output may be copied to a printer at a later time via the OMEGA4 PLUS Disk Utility Program's C command, discussed in Topic C.3.2.

The width is used to specify page width in characters per line; the default is 80. The length is used to specify page length in lines per page; the default is 46.

The device parameter may be omitted in the D command line and only the C or D option included, if the user previously designated the line printer as the list output device during Disk Utility Program execution. A P command formatted in this manner is used to override the prevailing printer control option.

The P specification remains in effect until:

- 1) it is overridden by a subsequent P command, or
- 2) the programmer exits the current module.

If the programmer does not issue a P command and if no list output specification currently is in effect, list output is generated to the command output device by default.

Example: ?PD,LP

.
.
.
?P,TY
.
.
.
?P,LP

In this sequence, the programmer first directs list output to the line printer and specifies that LP refers to a Dataproducts line printer. Later, he changes the list output device to the TY device. At a later point in the session he issues another P command to direct list output to the LP device. Since he has not overridden his earlier specification concerning the line printer type, LP still refers to a Dataproducts line printer.

C.3.2 Copying a File or an Entire Disk: the C Command

Format: C,[type][format],device1,device2

The C command is used to copy the contents of the device or file (either binary or source) named "device1" to another device or file named "device2," where the device specifications made are from Subsection C.2. Various types of copy operations may be specified, represented by the type parameter. The format of the text being copied also may be specified as the format parameter.

The format parameter must be one of the following:

- S Specifies that data within the file is to be treated as source text, and will result in all parity bits being set to zero.
- B Specifies that data within the file is to be treated as binary object text.
- L Specifies that the data is in the form of lines of text preceded by either an ASCII control character or a FORTRAN control character.
- A Specifies that all files on a disk will be copied, but otherwise acts as if no format specifier were entered. When the A format is included, the device parameters must represent disks.

When the user includes the S, B, or L parameter, the system is free to reblock the input, to standardize its parity, or to modify the input in other ways during the copy operation. In any case, appropriate validity checks are made on the input and check sums are output.

When the format parameter is omitted, or when A is the parameter, the system generates to device2 a byte-for-byte unformatted copy of device1. The unformatted copy is an exact image of the source code, except for differences in extent boundaries and other physical variations. Neither software CRC validity checks nor special end-of-file checks are performed when the format parameter is omitted.

The type parameters and corresponding functions are as follows:

- U Copies an entire disk in the unformatted manner described above; notes error but does not attempt error recovery. The format parameter cannot be used with the U type parameter. Device parameters used must represent floppy disk devices.
- F Copies device1 to device2. If a file with the same name as "device1" already exists on device2, the contents of that file are replaced by those of device1.
- C Copies device1 to device2. If a file with the same name as "device1" already exists on device2, no copying takes place.
- R Replaces the contents of a file named "device1" that resides on device2, with the contents of device1. If a file named "device1" does not exist on device2, no copying takes place.

If the type parameter is omitted from the command line, an F-type copy operation is performed by default. If both type and format parameters are omitted, an unformatted F-type copy operation is performed.

Example: ?C,,DF1.ANYFILE,DF.YZXX3

In this example, the programmer instructs the system to copy the contents of a file named ANYFILE from the floppy disk in unit 1 to a file named YZXX3 on the floppy disk in unit 0. Since the format specifier is omitted here (two commas are entered consecutively instead) the copy will be an exact image of ANYFILE on DF0. No error checking, parity standardization, or reformatting will be done.

If a paper tape or disk file was indicated to receive list output in a previously-executed module's P command, the C command later may be used to copy that file to the line printer.

Example: ?C,L,DF.PGMG,LP

In this example, the file PGMG, previously written to the floppy disk in unit DF, is reproduced on the line printer. Care should be taken not to use "S" as a format specifier, nor to omit the specifier entirely, as an "S" defeats page formatting, and omitting the specifier entirely results in no carriage returns being issued.

C.3.3 Deleting a File: the D Command

Format: D,file

The D command provides the user with a means of removing the file named "file" from a floppy disk. The name "file" first is removed from the floppy disk's directory; then, the space allocated to the file "target" is made available.

The parameter "file" must be an acceptable disk file name, as described in Subsection C.2. If a non-file device such as a keyboard/printer or paper tape device is specified as "file," the command is completely ignored. Likewise, use of the device parameter DU causes the command to be ignored.

C.3.4 Formatting a Disk: the F Command

Format: F,disk[,k]

The F command provides a means of reformatting a floppy disk to prepare it for labeling. Entry of the F command causes a floppy disk, identified using the specification in Subsection C.2, to be formatted. The optional parameter "k" may be specified to identify the sector-ordering sequence to be used.

Floppy disks are formatted using the standard IBM sector-ordering sequences presented in Figure C-1. If the k parameter is not specified, the disk is formatted in sequence number 1.

Sequence Number																											
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
2	1	3	5	7	9	11	13	15	17	19	21	23	25	2	4	6	8	10	12	14	16	18	20	22	24	26	
3	1	4	7	10	13	16	19	22	25	2	5	8	11	14	17	20	23	26	3	6	9	12	15	18	21	24	
4	1	5	9	13	17	21	25	2	6	10	14	18	22	26	3	7	11	15	19	23	4	8	12	16	20	24	
5	1	6	11	16	21	26	2	7	12	27	22	3	8	13	18	23	4	9	14	19	24	5	10	15	20	25	
6	1	7	13	19	25	2	8	14	20	26	3	9	15	21	4	10	16	22	5	11	17	23	6	12	18	24	
7	1	8	15	22	2	9	16	23	3	10	17	24	4	11	18	25	5	12	19	26	6	13	20	7	14	21	
8	1	9	17	25	2	10	18	26	3	11	19	4	12	20	5	13	21	6	14	22	7	15	23	8	16	24	
9	1	10	19	2	11	20	3	12	21	4	13	22	5	14	23	6	15	24	7	16	25	8	17	26	9	18	
10	1	11	21	2	12	22	3	13	23	4	14	24	5	15	25	6	16	26	7	17	8	18	9	19	10	20	
11	1	12	23	2	13	24	3	14	25	4	15	26	5	16	6	17	7	18	8	19	9	20	10	21	11	22	
12	1	13	25	2	14	26	3	15	4	16	5	17	6	18	7	19	8	20	9	21	10	22	11	23	12	24	
13	1	14	2	15	3	16	4	17	5	18	6	19	7	20	8	21	9	22	10	23	1	24	12	25	13	26	
14	1	15	2	16	3	17	4	18	5	19	6	20	7	21	8	22	9	23	10	24	11	25	12	26	13	14	
15	1	16	2	17	3	18	4	19	5	20	6	21	7	22	8	23	9	24	10	25	11	26	12	13	14	15	
16	1	17	2	18	3	19	4	20	5	21	6	22	7	23	8	24	9	25	10	26	11	12	13	14	15	16	
17	1	18	2	19	3	20	4	21	5	22	6	23	7	24	8	25	9	26	10	11	12	13	14	15	16	17	
18	1	19	2	20	3	21	4	22	5	23	6	24	7	25	8	26	9	10	11	12	13	14	15	16	17	18	
19	1	20	2	21	3	22	4	23	5	24	6	25	7	26	8	9	10	11	12	13	14	15	16	17	18	19	
20	1	21	2	22	3	23	4	24	5	25	6	26	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
21	1	22	2	23	3	24	4	25	5	26	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
22	1	23	2	24	3	25	4	26	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
23	1	24	2	25	3	26	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
24	1	25	2	26	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
25	1	26	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	

Figure C-1. Standard IBM Sector-Ordering Sequences for Floppy Disks

NAKED MINI 4 PROCESSOR ACCEPTANCE TEST (PAT4)

90-93802-00

```

AD = :0000
0000  4F83 0004 0001 0000 0006 0009 0200 0000
0008  0006 0006 0006 1000 0200 0F68 0040 3000
0010  8271 8FF8 2200 2020 5A76 3100 AF75 866F
0018  2901 0908 0300 2200 030A 0600 9F7D 826A
0020  19FF 1101 0F01 7263 2081 0000 9FF9 9FFC
0028  0FF5 0125 0015 0120 0000 0000 0000 001A
0030  0001 0000 0000 0000 0188 0000 0007 0000
0038  0000 0000 0002 0100 0100 0000 0000 0000
0040  5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A
0048  5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A
0050  5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A
0058  5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A
0060  5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A
0068  5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A
0070  5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A
0078  5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A 5A5A
0080  4F53 3420 2020 0000 001A 0001 0040 0001
0088  0006 00FA 0000 4F53 3735 3033 3037 3137
0090  3030 3532 4F55 3730 3034 3238 3135 3234
0098  3239 2020 3732 3033 323E 3135 3234 3239
00A0  0000 0000 0000 0000 0000 0000 0000 0000
00A8  0000 0000 0000 0000 0000 0000 0000 0000
00B0  0000 0000 0000 0000 0000 0000 0000 0000
00B8  002F 0F40 0547 0120 5559 494C 2056 4F4C
00C0  2041 4F41 4C1F 472F 4009 8749 5401 4C2F
00C8  414F 414C 4F47 2044 4901 470F 4F53 5449
00D0  4320 2020 2020 2020 2020 2020 2020 2020
00D8  2020 2020 2020 2020 2020 2020 2020 2020
00E0  2020 2020 2020 2020 2020 2020 2020 2020
00E8  2020 2020 2020 2020 2020 2020 2020 2020
00F0  2020 2020 2020 2020 2020 2020 2020 2020
00F8  2020 2020 2020 2020 2020 2020 2020 2020

```

Figure C-2. 8-Column Dump Listing

C.3.8 Renaming a File: the R Command

Format: R,file1,file2

The R command is used to change the name of file1 to file2, where file1 is a valid floppy disk file identifier, as described in Subsection C.2. The parameter file2 must be a valid file name, i.e., a file identifier minus the disk device specification. If a file already exists with the name "file2" on that floppy disk, the operation aborts and no renaming is done. If file1 does exist and the renaming does not generate any file name conflicts, that fact is noted in a RENAMED message from the Disk Utility Program.

Example: ?R,DF1.OLDNAME,NEWNAME
 RENAMED
 ?

Entering this command changes the name of the file OLDNAME on the DF1 device to NEWNAME.

C.3.9 Transferring Control to Another Address or Program: the J Command

Format: J,address

The J command is used to transfer program control to a specified hexadecimal address, generally the starting address of any other program resident in memory. When control is transferred to PAT4 or another program in memory, user program or other memory address, the system begins to execute that program.

C.3.10 Invoking Another Program Module: the E Command

Format: E,file[,address1[,address2]]

The E command is used to terminate an executing program, then to load and execute another program. Entering the E command causes the floppy disk program "file" (identified using the specification in Subsection A.2 to be loaded and executed. Before attempting to execute this command, OMEGA4 PLUS closes all open devices other than command input and command output files, thereby ensuring the availability of buffers and control blocks for the transfer of control.

The optional address1 parameter is entered when the user wants the module to be loaded at a particular address. Otherwise, the system loads the module as high in memory as possible.

The user enters the optional address2 parameter when he wants to specify that the module's bootstrap program be loaded at address2 and that the module itself be loaded at address1. Entry of this parameter is essential if the module to be loaded will occupy any part of memory currently occupied by the resident module's bootstrap program. For example, this parameter should be used when a diagnostic program with a required execution address of :0 is being loaded. The bootstrap program occupies approximately :300 words of memory; therefore, to execute the module in low memory it is recommended that the bootstrap program be loaded in high memory.

After the E command is entered with one of the PAT4 files specified, PAT4 identifies itself and generates a command prompt.

Example: ?E,DF.PAT410BASIC.BIN,0,5000
 ?
 _

In the above sequence, the programmer instructs the system to load the PAT4 file's bootstrap program at location :5000, to load PAT4 at location :0, and to execute it.

C.4 OMEGA4 PLUS DISK UTILITY PROGRAM COMMAND SUMMARY

The commands listed below may be entered under OMEGA4 PLUS Disk Utility Program control.

C,[type][format],device1,device2

Copies source or binary code from device1 to device2 with the following allowable type options enabled:

- U - Performs the copy operation on an entire disk without regard to format.
- F - Performs the copy operation, replacing a file on device2 named "device1" with the contents of device1.
- C - Performs the copy operation, but does not replace any existing file on device2 named "device1."
- R - Performs the copy operation but requires that a file named "device1" exist on device2 to be replaced.

D,file

Removes name of file from disk directory; physically removes file contents.

E,file [,address1[,address2]]

Loads and executes a specified file at address1, optionally loading the file's bootstrap program at address2.

F,disk[,k]

Formats a disk with an optional sector-ordering sequence k.

J,address

Transfers control to the specified address.

L,disk[(count)],label

Performs a quick analysis of a disk and labels it with the specified label.

P { [Wwidth][Llength],device
C[Wwidth][Llength][,device]
D[Wwidth][Llength][,device] }

Designates the list output device.

R,file1,file2

Changes the name of a file in a disk directory.

V[N],disk

Displays a listing of files in a disk directory.

X $\begin{matrix} W \\ L \end{matrix}$,disk,au1[,au2] Displays a listing of files in a disk directory.

C.5 OMEGA4 PLUS DISK UTILITY PROGRAM MESSAGE SUMMARY

Various messages are generated by the OMEGA4 PLUS Disk Utility Program as a result of both normal processing and error detection. Those messages resulting from normal processing require operator intervention, as indicated beside the appropriate messages below. The meanings of those messages generated in response to error conditions are listed below also.

Message

Meaning

:nnnn BAD SECTORS

nnnn (in hexadecimal) floppy disk sectors generated CRC errors when tested for usability during a label operation (L command). A list of the bad sectors will be displayed with the message on the CO device.

** DE 02 device

A needed device is not ready, no device is connected to a needed logical unit, or no directory is present on a disk.

<u>Message</u>	<u>Meaning</u>
** DE 03	A hardware status error has occurred on a device being used.
** DE 04	The device directory or F-list is full.
** DE 05	A needed device is not labeled.
** DE 07	A referenced file was not found on the requested device.
** DE 13	An invalid logical unit was specified (system error).
** DE 14	All FCBs are in use (system error).
** DE 15	Insufficient buffer space is available for requested operation. (System error)
DELETED	The file specified was deleted.
** PE 01 ON CYL :nn	An error occurred in reading a disk during an unformatted a disk during an unformatted copy operation (C command).
** PE 02	An error occurred in writing to a disk during an unformatted copy operation (C command).
** PE 03	A label operation failed for lack of sufficient good AUs on the disk to store the system control block.
RENAMED	The file specified was renamed.
** SE	A syntax error was detected in a user command.

