

MOS RAM DIAGNOSTIC

93926-90



Engineering Reference Number
90-93926-00A2

NAKED MINI. Division

HERTFORD HOUSE
DENHAM WAY, MAPLE CROSS
RICKMANSWORTH, HERTFORDSHIRE WD3

Telephone Rickmansworth 71211

Telex S

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

REVISION HISTORY

<u>Revision</u>	<u>Issue Date</u>	<u>Comments</u>
A0		Original issue
A1	October 1977	Incorporates LSI-3/05, NAKED MINI® 4
A2	July 1979	Updates manual to reflect software compatibility with OMEGA4 PLUS loader; stylistic modifications

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 REQUIRED HARDWARE	1
	2.2 REQUIRED SOFTWARE	2
SECTION 3	REFERENCED DOCUMENTS	2
SECTION 4	GENERAL DESCRIPTION	2
SECTION 5	PROGRAM OPERATION	3
	5.1 PROGRAM LOADING	3
	5.1.1 Papertape	4
	5.1.2 Diskette	4
	5.2 TTY QUERIES AND RESPONSES	5
	5.3 DIAGNOSTIC RUN INDICATORS	6
	5.3.1 Sense Switch	6
	5.3.2 Console Data Register	6
	5.4 DIAGNOSTIC TERMINATION AND RESTART	6
	5.5 POWER FAIL HANDLING	6
	5.6 MEMORY PARITY ERROR HANDLING	7
SECTION 6	ERROR HANDLING	7
SECTION 7	DIAGNOSTIC TEST DESCRIPTIONS	7
	7.1 UNIQUE ADDRESS TEST	7
	7.2 PING-PONG TEST	8
SECTION 8	RUN SUMMARY	9

APPENDIX A	OMEGA4 PLUS DISK UTILITY PROGRAM	9
A.1	INTRODUCTION	9
A.2	PHYSICAL DEVICE USE	10
A.3	DISK UTILITY PROGRAM COMMANDS	12
A.3.1	Designating the List Output Device: the P Command	13
A.3.2	Copying a File or an Entire Disk: the C Command	14
A.3.3	Deleting a File: the D Command	16
A.3.4	Formatting a Disk: the F Command	16
A.3.5	Labeling a Disk: the L Command	18
A.3.6	Displaying a Disk's Files: the V Command	18
A.3.7	Dumping the Absolute AUs of a Disk: the X Command	19
A.3.8	Renaming a File: the R Command	21
A.3.9	Transferring Control to Another Address or Program: the J Command	21
A.3.10	Invoking Another Program Module: the E Command	21
A.4	OMEGA4 PLUS DISK UTILITY PROGRAM COMMAND SUMMARY	22
A.5	OMEGA4 PLUS DISK UTILITY PROGRAM MESSAGE SUMMARY	23

FIGURES

Figure

7-1	Ping-Pong Test	8
A-1	Standard IBM Sector-Ordering Sequences for Floppy Disks	17
A-2	8-Column Dump Listing	20

SECTION 1. INTRODUCTION

The MOS RAM Diagnostic program is designed to test Computer Automation's metal oxide semiconductor (MOS) random access memory (RAM). The diagnostic consists of four main parts:

- Executive
- Unique Addressing Test
- Ping-Pong Test
- Error Handling

The primary objectives of the diagnostic are:

- Pattern sensitivity testing
- Addressing logic testing
- Interaction of MOS RAM row addressing
- Interaction between bits
- Pattern sequence dependency

Because the MOS RAMs are used with memory banking, the diagnostic testing is designed to check banking logic also.

The MOS RAM Diagnostic runs on any Alpha LSI or NAKED MINI® 4 Family computer.

SECTION 2. ENVIRONMENT

The minimum hardware and software requirements to run this program are given below.

2.1 REQUIRED HARDWARE

The following hardware is required to run this program on any of the specified processors:

- LSI-2 Series, LSI-3/05 Series or NAKED MINI 4 Family processor with Programmer's Console
- Minimum 8K words of memory (16K for NAKED MINI 4)
- ASR33 TTY (or CRT)
- Paper tape reader or floppy disk drive for program loading

2.2 REQUIRED SOFTWARE

The following software is required:

Paper tape

- LSI-2 Series, LSI-3/05 Series or NAKED MINI 4 Family binary loader

Engineering No.

- Program tape
 - LSI-2 00-93926-30
 - LSI-3/05 00-93926-31
 - NAKED MINI 4 00-93926-42

Diskette

- LSI-2, LSI-3/05 00-93926-01
- NAKED MINI 4 00-93926-05

SECTION 3. REFERENCED DOCUMENTS

Refer to the following documents for supplementary information:

	<u>Engineering No.</u>	<u>Marketing No.</u>
• LSI-2 Series Minicomputer Handbook	91-20400-00	63-20400-91
• LSI-3/05 Series Millicomputer Handbook	91-10005-00	63-10005-91
• LSI 4/10 Computer Reference Manual	91-20990-00	63-20990-91
• LSI 4/30 Computer Reference Manual	91-20991-00	63-20991-91
• LSI 4/90 Computer Reference Manual	91-20945-00	63-20945-91
• Memory Banking Controller User's Manual	91-20403-00	63-20403-91
• Directoried Load/Dump (DLD) User's Manual	90-93012-00	63-93012-90

SECTION 4. GENERAL DESCRIPTION

The MOS RAM diagnostic test consists of several subroutines under the control of the Executive. The diagnostic test is small enough to reside within 4K words of memory and is loadable into any memory bank of 4K or more words. It communicates with the user via TTY or CRT.

The Executive portion of the diagnostic communicates with the user and monitors the subroutines. The Executive calculates the starting address, evaluates any memory banking requirements, and controls the read/write functions, the number of test passes, and the overall execution of the diagnostic.

The main diagnostic functions are performed by the Unique Address test and the Ping-Pong test subroutines. Both tests can exercise memory banking logic when required. In case of an error, a call is made to the Error Handling routine which can give either a full report of each error or just the number of errors per pass. The Unique Address test verifies the RAM row addressing logic and possible interaction between two locations. The Ping-Pong test is a general-purpose semiconductor memory test program that tests all bits in a 4K array as well as addressing, interaction between bits, and pattern and sequence dependency.

The memory banking test control logic resides in the Executive part of the diagnostic program. Memory banking is a method of enabling and disabling blocks of memory so that the total installed memory capacity of the system can exceed the addressing limit of the processor. The integrity of data stored in a disabled block is not affected by the memory banking logic. (For details, refer to the Memory Banking Controller User's Manual.)

The Executive part of the diagnostic program switches to desired alternate memory by using output/select and present commands and passes the address and other parameters to the Unique Address and Ping-Pong test subroutines. The diagnostic program can also test non-banked memory by bypassing the logic that selects the modules for testing.

SECTION 5. PROGRAM OPERATION

5.1 PROGRAM LOADING

The program may be loaded into any memory module of at least 4K.

During each pass of the diagnostic, the Unique Address test is performed first for all of the memory specified for testing, followed by the Ping-Pong test for each 4K module. The home primary (and its alternates) for the program is tested after all of the others. In an LSI-2 system, the program is relocated to the next primary, the Ping-Pong test (only) is performed in the home area, and the program is returned to its home primary. (If the program is loaded in resident memory, no relocation is necessary.)

The program is not relocated during testing in an LSI-3/05 or NMD-4 system, so it must be loaded into a resident memory below or above the memory under test. Consequently, in a NAKED MINI 4 system, the program can occupy memory from :0-:FFF and the test range can be from :1000-:FFFF maximum, or the memory under test can occupy lower memory and the program can be loaded in increments of 4K anywhere in the high memory up to 64K (:FFFF). In the LSI-3/05 system, the program can be loaded anywhere in memory, provided the total memory does not exceed 32K.

5.1.1. Papertape

If the program is supplied on paper tape, it may be loaded using any binary loader, including AutoLoad. The diagnostic loads initially at REL :100. After loading, the queries will be displayed.

5.1.2 Diskette

Procedures for loading the MOS RAM Diagnostic vary depending on whether it is used with an LSI-2, LSI-3/05 or NAKED MINI 4 system.

5.1.2.1 Using the LSI-2, LSI-3/05

To begin execution of the MOS RAM Diagnostic from an LSI-2 or LSI-3/05 diskette, AutoLoad the MOS RAM Diagnostic diskette. The DLD Software System will come up. In a response to the prompt, enter a DLD command to load and execute the program via its filename. The Diagnostic will begin execution at location :100.

To relocate to other places in Memory, refer to the Directoried Load/Dump (DLD) User's Manual for further instructions on relocating the program to other places in memory.

5.1.2.2 Using the NAKED MINI 4 Diskette

When a diskette is used to load and execute the MOS RAM Diagnostic on a NAKED MINI 4 processor, 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

OMEGA4 DISK UTILITY (CO):5DA0-:7DFE

The user should issue an E command to load and execute the MOS RAM Diagnostic via its file name, MOSRAMDIAG.BIN. The E command should take a form similar to the following:

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

Because this version of the MOS RAM Diagnostic must be loaded at location :0, the user must specify both the load address for the Diagnostic (0), and the load address for the Diagnostic bootstrap program (e.g., 5000). Keeping in mind the extents of the Disk Utility Program and the size of the MOS RAM Diagnostic, the user should specify a load address for the bootstrap which falls between them. The size of the Diagnostic 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. Refer to Appendix A for more information on the OMEGA4 Disk Utility Program.

After loading, the diagnostic will begin execution at location :100 and the queries will be displayed.

5.2 TTY QUERIES AND RESPONSES

Test parameters are entered via the TTY (or CRT) in response to queries from the diagnostic program. In order to clarify which is query and which is response, the query from the diagnostic is underlined in this document; the response from the user is not. No spaces are permitted between parameters. Each reply must be terminated by a carriage return. A back arrow (←) is used to signify erroneous input or for the user to cancel the input. The query will be repeated. An up arrow (↑) can be used to restart the queries from the beginning.

*FULL REPORT? { Y
N

Y = Full report including failed location actual data, received data.
N = Number of errors per pass, only (no printout if no errors).

*ADDRESS RANGE? LLLL,hhhh

Memory test range from low address (LLLL) to high address (hhhh). High address must be a multiple of xFFF.

*BANKED? { Y
N

Y = Memory is banked; must provide parameter for the following query.
N = Memory is not banked; end of query.

*Pn?SP,SA1,SA2(#IB),SA3...

Size of primary module n, where:
SP = Primary module size
SA1 = Alternate 1 size
SA2 = Alternate 2 size
#IB = Number of interbanked modules;
alternate and primary module sizes are
always the same.

Examples:

*P0?8,8,8(2),8

Primary 0 has 8K words and 3 alternates;
Alternate 2 is interbanked with 2 modules

*P1?16,16,16,32(3)

Primary 1 has 16K words and 3 alternates;
Alternate 3 is interbanked with 3 modules.

*P2?#

No more primaries; end of query.

5.3 DIAGNOSTIC RUN INDICATORS

5.3.1 Sense Switch

The Sense switch controls the number of passes. If set to 1, the program stops after one complete pass; if set to 0, the test keeps repeating until the Sense switch is reset to 1.

5.3.2 Console Data Register

Bit 15 of the Console Data Register indicates which test is being run (bit 15 = 0, Unique Address test; = 1, Ping-Pong test). The remaining 14 bits display the current pass number being run.

5.4 DIAGNOSTIC TERMINATION AND RESTART

In order to stop the diagnostic program, set the Sense switch to 1. The program will stop after completing the current pass. Press RUN to continue execution of the next pass. If a new query is desired, start the program from the beginning by loading the P Register with the start address.

5.5 POWER FAIL HANDLING

When a power failure occurs, the diagnostic halts. On power-up, the program displays a power-fail message and automatically restarts from the beginning.

5.6 MEMORY PARITY ERROR HANDLING

The parity error interrupt vector preserves the location that caused the parity error. The interrupt vector is located at:

- For LSI-2 :12
- For LSI-3/05 :88 (uninstalled memory trap)
- For NAKED MINI4 :88 (uninstalled memory trap)

In case of a full error report, the parity error message and location are displayed. If the full report option is not selected, a count of parity errors is kept and is displayed once every pass. The program does not come to a halt.

SECTION 6. ERROR HANDLING

The Error Handling routine can give either a full report of each error or the total number of errors for each pass, depending on the response to the query:

*FULL REPORT?

If the response is Y (yes), a full report is given for each error:

ERROR LOCATION :XXXX, DATA RECEIVED :XXXX, ACTUAL DATA :XXXX

If the response is N (no), or the return key is hit, the above information is suppressed and only the total number of errors are given after each pass. However, there will be no error output unless an error occurs.

SECTION 7. DIAGNOSTIC TEST DESCRIPTIONS

7.1 UNIQUE ADDRESS TEST

The Unique Address test verifies the RAM row addressing logic and possible interaction between two locations. The test writes a unique pattern in each memory location and then reads and verifies each address for correct data. The Unique Address test is not performed in the module where the diagnostic program resides. If test memory exceeds 64K words, the unique pattern repeats itself. Before starting the Unique Address test, memory is initialized to zero (i.e., zeros are written and read back to ensure a proper initialization).

7.2 PING-PONG TEST

The Ping-Pong test is a general-purpose semiconductor memory test of all bits in a 4K array, addressing, interaction between bits and a pattern, and sequence dependency. The test is always performed in modules of 4K words. The test writes a unique pattern (generated by an algorithm) in each memory location of the 4K module. Then a read/compare-to-pattern is performed in ping-pong fashion starting at location 1, then jumping to the end location of the module. With each jump the start location is incremented by one or the end location is decremented by one so that the test ends in the middle of the module.

An example is given below. The number in the upper right-hand corner of each square represents the test sequence. The data contained in each square is given at the bottom. Each square represents a word of memory. Note that the test sequence ping-pongs from the upper left to the lower right, etc.

1 :55AA	3 :A	5	7
9	11	13	15
16	14	12	10
8	6	4 :AA	2 :AA55

Figure 7-1. Ping-Pong Test

SECTION 8. RUN SUMMARY

The MOS RAM Diagnostic goes through the following steps, under the control of the Executive:

1. Request error option and save response
2. Request address range and save response
3. Request banking information
4. If banked, build a data table for parameters
5. Extract data for desired primary and alternate modules
6. JST to Unique Address test with required parameters
7. Perform Unique Address test on each (primary) module (and alternate, if banked)
8. JST to Ping-Pong test with required parameters
9. Perform Ping-Pong test on 4K modules (primary and alternates, if banked)
10. Verify that all memory specified has been tested.
11. Go to 5 for next pass if Sense switch is set to 0, or halt if set to 1.

APPENDIX A. OMEGA4 PLUS DISK UTILITY PROGRAM

A.1 INTRODUCTION

The Disk Utility Program is included as a file on the MOS RAM Diagnostic diskette, allowing the user to create floppy disks in Standard File Manager (SFM) format, and to maintain such disks. The Disk Utility Program 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.

A.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 :F0. 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 extension 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.

A.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.

A.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 A.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 A.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.

A.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 A.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.

A.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 A.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.

A.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 A.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 A-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	17	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	25
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 A-1. Standard IBM Sector-Ordering Sequences for Floppy Disks

A.3.5 Labeling a Disk: the L Command

Format: L,disk[(count)],label

The Standard File Manager (SFM) requires that all file-type devices it accesses be labeled in SFM format. The L command can be used by the OMEGA4 PLUS programmer to label his disks.

When the L command is entered, the system performs a quick surface analysis of the disk (identified using the appropriate device specification in Subsection A.2, searching for bad sectors. Using this information, the SFM is able to avoid accessing faulty sectors later.

The system then writes the designated "label" to the floppy disk. It also places an F-list on the floppy disk with space for the number of files specified in the optional decimal "count" or for 32 files if count is not specified. A directory, tables of usable and bad sectors, and the bootstrap code are also written to the floppy disk.

The label entered in the command is a character string of up to 14 characters. If the programmer enters less than 14 characters, the program pads the string with blanks to produce the 14-character string. The label must begin with a letter and may contain only alphanumeric characters, embedded colons, and trailing blanks, if the disk is to be fully compatible with OS4.

Example: L,DF3(20),OTHER:COPY

In the example presented above, the programmer instructs the system to perform a surface analysis of the standard floppy disk located in drive unit 3, and to label it "OTHER:COPY." The command also specifies that space be allocated for 32 (hexadecimal 20) files in the F-list.

A.3.6 Displaying a Disk's Files: the V Command

Format: V[N],disk

The V command generates to the list output device a listing of all files in the directory of a disk. Each file's F-number and other pertinent F-list information also is included in the listing.

The device parameter included must be a device identifier for a floppy disk, as described in Subsection A.2. When the optional N parameter is not included in the command line, the current page of list output is ejected, and the file listing begins at the top of the next page. Entering the N parameter causes the listing to be printed directly under other list output, with no page ejected.

Example: ?V,DF
 ?
 _

This command requests a listing of the files contained on the DF device. Since the N option is not included, the listing is output after a page is ejected.

A.3.7 Dumping the Absolute AUs of a Disk: the X Command

Format: X $\begin{bmatrix} L \\ W \end{bmatrix}$,device,au1[,au2]

The X command generates to the current list output device a listing of the contents of a specified allocation unit (AU) or a range of AUs. An AU is a unit of data equivalent to 510 bytes or 4 sectors on a floppy disk.

The device parameter included must be a device identifier for a floppy disk, as described in Subsection A.2. The au1 parameter, when entered without the au2 parameter, specifies the number of the AU to be dumped. When the au2 parameter also is entered, the range of AUs from au1 through au2 is dumped. The L parameter is entered when an AU listing 8 columns wide is desired. The W parameter is entered when wide line printer paper is being used; it causes the contents of 16 words of storage to be output per line instead of eight. Use of the W parameter requires the printer page to be 87 columns wide and the contents of word 6 of memory to be greater than :57. When neither parameter is included, an 8-column wide listing is produced by default.

The data in the listing is ordered so that bytes within a sector follow one another, that sectors on one surface of a disk follow one another, and that the other surfaces of the disk accessible without head movements, follow. Two AUs are printed on each page, identified by AU number. The contents of each AU are identified by a relative word number on the left (in hexadecimal), and are printed in 8 or 16 columns of 4 hexadecimal digits each. Figure A-2 illustrates a dump listing 8 columns wide.

Example: ?P,LP
 ?X,DF1,26,28
 ?
 _

This command sequence assigns list output to the line printer device, then it requests a dump of AUs 26 through 28 from the DF1 unit.

```

AU = : 0000
0000  9E8D 0004 0000 0000 0000 0009 0200 0000
0008  0000 0000 0000 1000 0200 9E0B 0940 3000
0010  827C 86F8 2900 8080 AA76 31BD AE75 A66F
0018  2901 0988 030A 2200 030A 0600 9E7D 826A
0020  18FF 1141 0E0D A263 9C81 0000 9EF8 9EBB
0028  9EF4 0192 00B5 0129 0000 0000 0040 001A
0030  0001 004D 0004 0004 0088 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 0080 001A 0001 004D 00C1
0088  0004 00FA 0000 4F53 3739 3031 3234 3036
0090  3039 3039 4F53 3739 3031 3234 3036 3039
0098  3039 2020 3739 3031 3234 3036 3039 3039
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  000E 4F4D 4547 4134 3A50 4C55 5320 2020
00C0  2020 2020 2020 2020 2020 2020 2020 2020
00C8  2020 2020 2020 2020 2020 2020 2020 2020
00D0  2020 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 A-2. 8-Column Dump Listing

A.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 A.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.

A.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 or any other program resident in memory. When control is transferred to another program in memory, user program or other memory address, the system begins to execute that program.

A.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 the MOS RAM Diagnostic as the specified file, the Diagnostic identifies itself and generates a command prompt.

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

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

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

A.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 C0 device.

** DE 02

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 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.