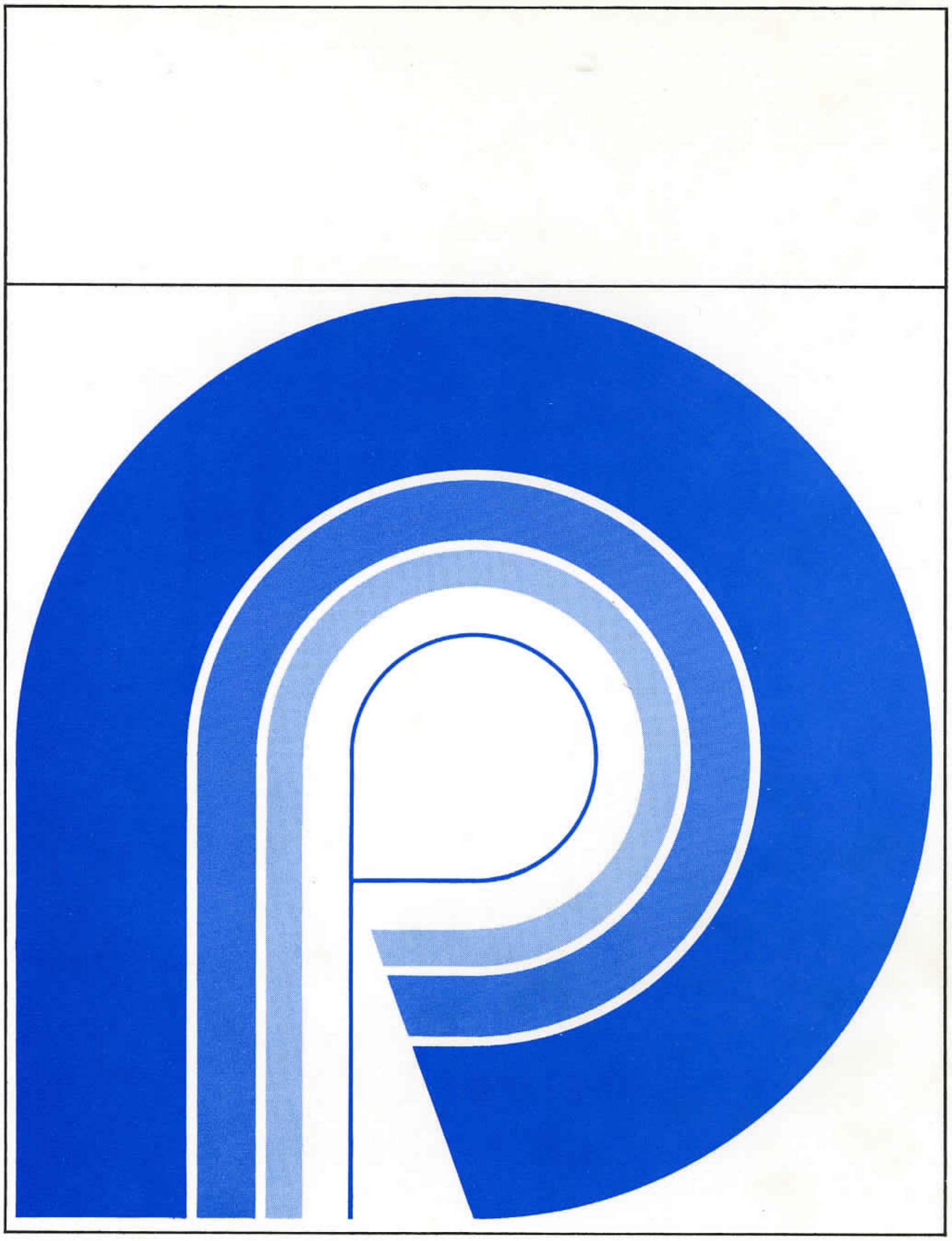


PCC

PERTEC
COMPUTER
CORPORATION
MICROSYSTEMS DIVISION

FDOS-III

OPERATOR'S MANUAL

505-345-9084

BoB williams

Product Support

icom® MICROPERIPHERALS™

Products of PCC Pertec Computer Corporation

Microsystems Division

September, 1977

NOTICE

Information contained within this document may not be reproduced, distributed or disclosed in full or in part by any person without prior approval of PERTEC MICROSYSTEMS DIVISION.

TABLE OF CONTENTS

Section Page

TABLE OF CONTENTS

Section Page

TABLE OF CONTENTS

SECTION I

PRELIMINARY OPERATION

1-1 LOADING FDOS-III

To load FDOS-III, follow the microcomputer's recommended start-up procedure for the resident debug or monitor, if available. Insert a system diskette into drive Ø, allow the drive to come up to speed and then GOTO the starting PROM address as detailed below.

Configuration		PROM Resident Starting Address (in hex)	Comments	Mini-monitor (See appendix B)
System	iCOM Dash Number			
MDS	-53	E8ØØH		No
SBC 8Ø/1Ø SBC 8Ø/2Ø	-56	E8ØØH E8ØØH		No No
Altair	-57	CØØØH CØØØH	2SIO Ports 1ØH,11H	Yes
Altair S-1ØØ Bus	-58	CØØØH	Initialize I/O vectors, see 1-2	Yes
Poly	-59	CØØØH	With 4.Ø monitor	Yes
Sol	-6Ø	B8ØØH	With Solos monitor	Yes

The resident routine will load the FDOS-III executive into RAM. FDOS-III is loaded and waits for an input command when the prompt character ! appears on the console.

On the iCOM -57, -59, and -60 system I/O vector tables are loaded, after the Executive, into the interface on-board RAM. These vector tables configure the I/O to the appropriate system.

On the -58, the vector table must be initialized, refer to 1-2.

1-2 I/O VECTOR INITIALIZATION FOR -58 SYSTEMS

The system has an Altair^{tm}S-100 type bus and an I/O configuration of:

Port Ø used for control
Port 1 used for data
Input Ready Bit = Ø (zero = ready)
Output Ready Bit = 1 (zero = ready)

Initialize the vectors of the interface RAM by executing at C3E7H. This loads console vectors and gains access to the Minimonitor.

Then enter: GCØØØ(CR)

If a different I/O configuration is used, additional subroutines are needed.

Perform the following:

Insert system diskette in drive Ø
Execute at ØCØØØH
When CPU cycles at C2XX, halt computer
Enter CI (console in) and CO (console out) vectors to point to the user I/O subroutines according to the following table.

RAM Location	USER's CONTENTS	SUB-ROUTINE	DESCRIPTION
ØØ55H	ØC3H	JMP	Subroutine to return one character from console key board via the A-register carry bit reset.
ØØ56H	CI Address (Lo Byte)	CO	
ØØ57H	CI Address (Hi Byte)		
ØØ58H	ØC3H	JMP	Subroutine accepts a character from the C-register and outputs it to the console.
ØØ59H	CO Address (Lo Byte)	CO	
ØØ5AH	CO Address (Hi Byte)		
C4ØØH	ØC3H	JMP	Subroutine to return one character from console key board via the A-register, carry bit reset.
C4Ø1H	CI Address (Lo Byte)	C1	
C4Ø2H	CI Address (Hi Byte)		
C4Ø3H	ØC3H	JMP	Subroutine accepts a character from the C-register and outputs it to the console.
C4Ø4H	CO Address (Lo Byte)	CO	
C4Ø5H	CO Address (Hi Byte)		
Ø176H	CO Address (Lo Byte)		Vector CO restoration
Ø177H	CO Address (Hi Byte)	CO	

Above subroutines will establish jump instructions in FDOS-III.
Now, GOTO Ø18ØH. This will restart FDOS-III and its prompt character ! appears on the console.
Perform a SYSGN function as given in 5-23.

SECTION II

SYSTEM ORGANIZATION

2-1 SOFTWARE MODULES

FDOS-III consists of the following modules:

- Resident Module
- Executive
- Text Editor
- Relocating Assembler
- Linker
- Library Manager
- Memory-To-Disk Program
- Copy Program
- System I/O Generation

2-2 RESIDENT MODULE

The Resident Module is contained in PROM memory and is usually located on the interface board. The Resident Module performs disk and write operations. Also, the Resident Module contains the disk Input/Output handler and the bootstrap loader.

2-3 EXECUTIVE

Executive is transferred from disk into the microcomputer's RAM memory when program control is transferred to the bootstrap loader contained in the Resident Module. The Executive is in RAM and waiting for a FDOS-III directive, when ! (exclamation point) appears on the output console device. The Executive performs command line interpretation, file management, and operational functions.

2-4 TEXT EDITOR

The Text Editor takes data from a disk file, places it in RAM, performs the editing function, and stores the edited data back onto a disk file. The Text Editor is transferred from the disk memory file, EDIT, into RAM when the editor command is executed. Upon completion of the edit operations, the Executive is reloaded in RAM.

2-5 RELOCATING ASSEMBLER

Source program input is derived from a disk file and assembled object output is stored into a file on disk. The Relocating Assembler is transferred from the disk memory file ASMB into RAM when the assemble command is executed. At completion of the assembly operations, the Executive is reloaded into RAM.

2-6 LINKER

The Linker command is derived from the specified command file on diskette, binary relocatable modules are obtained from file on diskette, and output executable object code is stored on a disk file. The Linker is loaded from the file LINK to memory when the FDOS-III link command is executed. The Executive is automatically reloaded into RAM and executed when linking process is completed.

2-7 LIBRARY MANAGER

The Library Manager is invoked with LIB command. The LIB command causes the Library Manager to be loaded and executed. Upon termination, Executive is automatically reloaded and executed.

2-8 COPY

Copy allows data residing on the Drive Ø diskette to be duplicated onto the diskette in Drive 1. When Copy is completed, Executive regains control provided the system's diskette is in drive Ø.

2-9 MEMORY-TO-DISK

Memory-To-Disk program is transferred from diskette to memory and executed. Termination causes the Executive to be reloaded and executed, or control given to the system monitor.

2-10 SYSTEM I/O GENERATION

FDOS-III System Generation program is transferred from diskette to memory and executed. Termination causes the system data to be written to the system diskette, and Executive loaded and executed.

2-11 DISK LAYOUT

Except for the Resident Module, all programs have been stored on the diskette enclosed with the FDOS-III Software Package. Disk storage space is divided into distinct regions, four on a systems diskette and two on a user diskette.

2-12 SYSTEM DISKETTE

A system diskette is divided into four regions: file directory, system I/O data, system, and user file areas. The diskette enclosed with the FDOS-III Software Package is a preloaded FDOS-III diskette. On a system diskette, track Ø is reserved for the file directory, tracks 1 thru 3 are reserved for the storage of the system Executive, and the balance of the disk storage area is available as user file area.

NOTE

The Text Editor, Linker, Library Manager, System Generator Copier, Memory-to-Disk and Relocating Assembler should reside on a system diskette within the user file area, as they are on the supplied system diskette.

The system diskette contains the following programs:

Title	Definition	Format
ASMB	Relocating Assembler	Hex-ASCII Object
COPY	Copy Program	Hex-ASCII Object
EDIT	Editor	Hex-ASCII Object
EXEC	Backup Copy of FDOS Executive	Hex-ASCII Object
DIAGO	Disk Diagnostic Object	Hex-ASCII Object
DIAGS	Disk Diagnostic Source	Source
DKHNB	Disk Handler Routine	Relocatable Binary Object
LIB	Library Handler	Hex-ASCII Object
LINK	Linker	Hex-ASCII Object
MTDK	Memory-to-Disk	Hex-ASCII Object
MTDKS	Memory-to-Disk	Source
RDBFL	Binary Object File Reader	Hex-ASCII Object
SYSGN	System Generation Program	Hex-ASCII Object
TESTS	Disk Handler Test Program	Source
TEST1	Disk Handler Test File	Text
CMNDF	Link Command File for Disk Handler Test Program	

2-13 USER DISKETTE

A User Diskette is divided into file directory and user file areas. Track Ø is reserved for the file directory and the balance of the disk is available as user file area.

2-14 DISK FILES

A file is any collection of information. For example, it may contain program object, program source, or user generated information.

2-15 LOCATION

All disk files are contained in the user file region of a diskette. As disk files are created, disk space is reserved for that file; the next file immediately follows.

A disk file and its entry in the file directory are contained on a diskette. When a file is deleted, its space is made available to the succeeding file. The same is done to the file directory, the space of the deleted file entry is taken by the next entry. This technique of filling the deleted space is referred to as disk packing.

2-16 FILE NAME

Each file in the user area of a diskette is accessible by a filename. This filename is stored, along with other file information, in the file directory area of the diskette. The filename is a string of one to five ASCII characteristics.

Examples of valid filenames:

JACK
JOE3
X
#SAM
BLOB5

2-17 DRIVE SPECIFIER

To call-up a disk file, a disk specifier is used. The drive specifier refers to the required drive. If the specifier is omitted, disk drive Ø is actuated.

For example:

JOE3:1
#SAM:3
X:Ø
JACK:2

2-18 DIRECTION LOCATION

Each diskette contains a file directory located on sectors 4 thru 26 of track Ø. The first three sectors are for system generation data. Each sector contains eleven file control blocks (FCB) and each FCB is eleven bytes long. Thus a file directory has room to accommodate up to 253 unique files per diskette.

Track Ø

Sector	Contents
1	System I/O Data
2	User Object Code
3	User Object Code
4	File Control Block (FCB) 1 thru 11
5	FCB's 12 thru 22
.	
.	
.	
26	FCB's 243 thru 253

2-19 CONTENTS DIRECTORY

File information of a diskette is kept in the file directory. The file directory consists of FCB's (file control blocks) each having eleven bytes.

The FCB layout is:

Byte	File Elements
1-5	Name padded with spaces (code 2Ø hex)
6	Attributes
7	Starting track address
8	Starting sector address
9-10	Length in sectors, most significant byte first
11	(reserved for Batch Mode Control Counter)

Since all file names on the diskette are contained in a single directory, each file name must be unique. An attempt to add a file name to the directory when the same file name already exists causes an error indication.

File attributes are characteristics of files than can be set and changed by the user. The FDOS-III attributes are:

ØØ - user file, no restrictions
Ø1 - permanent file, cannot be deleted

A file may have a length of from one sector up to a maximum of 1,975 sectors (252,8000 bytes).

2-20 SYSTEM DEVICE

The system device is always assumed to be a disk drive Ø. The diskette contained in this drive should always be a system diskette (see 2-12). Disk drive units 1, 2, and 3 may contain a system or a user diskette.

The system device is used as the bootstrap device. FDOS-III when it brings Executive, Text Editor, or other program modules from disk memory into RAM memory, assumes that the system diskette is contained disk drive Ø.

The system device is also considered to be the default directory device. Whenever a device suffix is omitted from FDOS-III command or from a file name, disk drive unit Ø is assumed.

2-21 SYSTEM I/O DATA AREA

The system I/O Data is always assumed to be on track Ø, sectors 1 thru 3, of the drive Ø diskette. This data is used by the Executive to specialize FDOS-III software for compatibility to the various hardware configurations.

The INIT and XGEN commands cause the I/O data area to be cleared (set to FFFF or FF hex).

SECTION III

OPERATION

3-1 STARTING FDOS-III

To start FDOS-III, follow the instructions for loading up operation as given in 1-1. When an exclamation mark (!) is printed on the console device, FDOS-III is awaiting command directives. A command directive is available that will return user control to the microcomputer's debug or monitor program (see section V).

3-2 COMMAND LINE

The FDOS-III command line is a command directive, followed by operands. The command directive must be separated from the first operand by a comma. Also, operands must be separated from one another by a comma.

For example: !ASMB,AL,BOB,3(CR)

NOTE

The FDOS-III command line must be terminated by a carriage return (CR). FDOS-III does not attempt to interpret or execute any command directive until the command line is terminated.

Prior to terminating a command line, characters in the command line may be deleted from the command device by pressing the RUBOUT key on the console device. Each time the RUBOUT key is pressed, the last character existing in the command line is deleted and echoes out to the console device, as verification that the character was deleted.

Example: (Rubout Key Depressed)

!ASMQQB,XL,LXAL,BOB,993(CR)

Is the Same As

!ASMB,AL,BOB3(CR)

To abort program, press the ESCAPE key (ESC). The command line is ignored and the FDOS-III prompt reappears.

Prior to terminating the command line with a return, reiteration of line may be caused by entering CTL-R. Entry of command line is then resumed from the last character entered.

Control characters (ØØH thru IFH), other than RETURN (ØDH), CTL-R (12H), and ESC (1BH), will be interpreted as invalid command inputs and will cause the FORMAT ERROR message to appear.

The command test may be any combination of upper or lower case. All numeric data is to be decimal unless followed immediately with an upper case H, denoting hex.

Example: 256 and 1ØØH will be interpreted as the same value.

3-3 ERROR MESSAGES

When a command is issued that contains an error, an error message will appear. The error messages are:

FORMAT ERROR - Command line format was incorrect and command execution could not proceed.

NO SUCH FILE - File name does not exist in the file directory of the specified diskette.

DUPL NAME - Attempt was made to enter a file name already existing in the file directory.

NO ROOM - More file disk space was requested than was available on the diskette, or the file directory entries was in excess of 254.

MEDIA ERROR - A copy of this media should be made to recover all but the inaccessible regions.

DRIVE NOT READY - An attempt was made to access a drive which was not engaged or which contained an unformatted diskette. Under most circumstances this message will be repeated once a second until the drive comes ready or until approximately 10 seconds have elapsed.

There are four error messages that originate from the Resident Module. The error message is a single digit, or a question mark, followed by a return to the microcomputer's debug or monitor program.

They are:

? - Checksum error incurred while loading an object file from disk.

1 - Unable to ready from diskette.

2 - Attempt to write more information to a file than space available.

3 - Referenced disk drive unit not ready.

3-4 OPERATOR INTERRUPTION

Operator termination of lengthy FDOS-III functions may be accomplished by entering CTL-C (Ø3 hex) from the input console device. Upon termination, control is passed to the console for normal FDOS-III command input.

Temporary interruption of any function may be accomplished by entering CTL-C from the console. The interrupted process will resume upon the second CTL-C entry.

NOTE

The entry of any data or command during the interruption will cause unpredictable results and may result in loss of data files.

SECTION IV

RESIDENT MODULE

4-1 DISK INPUT/OUTPUT

The Resident Module enables development of programs utilizing the floppy disk drive control. The module contains disk read (RI) and disk write (WRT) routines that provide byte oriented input and output capabilities.

To use the disk input and output routines, RI and WRT, first set up pointers to the area on disk that is to be accessed. This opens a disk file. Once a disk file has been opened, RI and WRT may be called any number of times in the same fashion as the console input and output routines in the microcomputer's debug or monitor program.

The driver handles all maintenance of the file pointers, once the file has been opened. Only one input file and one output file may be opened at any given time.

The following RAM memory locations are used by the RI and WRT routines. Refer to the Appendix for the memory address locations.

Locations	Description	
ISIZE	Input file	- size in sectors (2 bytes)
ITRK	Input file	- beginning track address
ISCTR	Input file	- beginning unit & sector address
ICNTR	Controller	- read buffer counter
OSIZE	Output file	- size in sectors (2 bytes)
OTRK	Output file	- beginning track address
OSCTR	Output file	- beginning unit and sector address
OCNTR	Controller	- write buffer counter

4-2 DISK INPUT

To open an input file, store the appropriate input file information into locations ISIZE, ITRK, ISCTR, AND ICNTR. Then, each call to RI will return the next byte of data from the disk into the same register that the microcomputer's debug or monitor program would normally return a console input data byte. If no additional data exists (input file size ISIZE has reached Ø) the carry bit is returned as 1; otherwise, the carry bit is returned as a Ø.

The contents of ISIZE should be set to one more than the number of sectors to be read before RI is returned to an end-of-file indication (carry bit set). If a unique end-of-file is to be performed, the file size may be set to some arbitrarily large number (FFFF, for example). The contents if ITRK should be set to the track number (ØØ-4C) from which input data is to be read.

The contents of ISCTR should be set to contain the drive unit number (ØØ-11) in bits 6 & 7, and the sector number (ØØ-19 hex) in bits Ø thru 5.

The contents of ICNTR should be set to ØØ. Each call to RI will bring in the next sequential date byte from the disk. As a sector (128 bytes) of data is read, RI increments the disk address (ITRK and ISCTR) and decrements the input size (ISIZE). Any sector containing a DD mark is ignored, but it is computed in the input size.

4-3 DISK OUTPUT

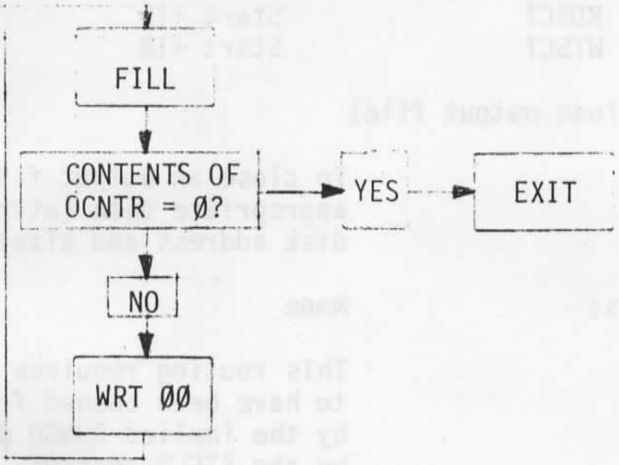
To open an output file, store the appropriate output file information into locations OSIZE, OTRK, OSCTR, and OCNTR. Each call to WRT will output, to disk, the byte contained in the same register that the microcomputer's debug or monitor program normally outputs a console data byte.

The contents of OSIZE should be set to the number of sectors allowed to be written before WRT terminates, causing error message 3 to appear on the console (see 3-3). If a maximum file size monitoring is to be performed, the file size may be set to some arbitrarily large number (for example FFFF).

The contents of OTRK should be set to the track number (ØØ-4C) where the output data is to begin writing. The contents of OSCTR should be set to the drive number (ØØ-11 in bits 6 & 7, and the sector Ø1-1A) where the output data begins its writing operation.

The contents of OCNTR should be set to ØØ. Each call to WRT will output one byte to disk. After 128 bytes have been sent to the disk, OTRK and OSCTR are incremented, and OSIZE is decremented. WRT verifies each sector it has written and if it is unable to write a sector after five attempts it writes a DD mark to that sector and advances to the next contiguous disk address and attempts disk write again. OSIZE is decremented for each sector written with a DD mark.

After all the data has been entered onto the disk, there may be data still remaining in the controller's write buffer. To insure that all data has been written onto the media, a pad character (e.g. ØØ) should be outputted until the write buffer reaches 128 bytes and WRT writes it to the disk. A flow chart of such a fill routine is:



4-4 DISK SECTORING

RI and WRT use logical physical techniques for addressing disk. The diskettes are continguous from 1-26 (Ø1-1A). After accessing physical sector 1, an entire revolution of the disk must occur if physical sector 2 cannot be immediately accessed. To avoid these rotational delays, RI and WRT translate the requested sector address (logical sector) into another sector address (physical sector) which is then used by RI and WRT. For example: if sector 2 is requested, physical sector 1Ø (ØA hex) occupies the accessed disk area.

4-5 EXECUTIVE ROUTINES

Six FDOS-III Executive routines can be used while FDOS-III Executive is in RAM. Vectors to these routines reside at START plus the specified value. The value for start is as follows:

Microprocessor	Start Address
MCS	2ØH
MDS	2ØH
ALTAIR/IMSAI	4ØH
Poly88	2Ø4ØH
SBC-8Ø/1Ø	4Ø4ØH
SBC-8Ø/2Ø	4Ø4ØH
SOL	4ØH

Routine	Vector Location
UPDAT	Start +3
OPENR	Start +6
OPENW	Start +9
STFL2	Start +12
RDSCT	Start +15
WTSCT	Start +18

4-6 UPDAT (close output file)

Function: To close an output file putting the appropriate data (attribute, starting disk address and size) in the directory.

Parameters: None

Comments: This routing requires the output file to have been opened from FDOS-III by the implied RUNGO directive or by the STFL2 routine.

4-7 OPENR (open existing file for input)

Function: A file is opened for reading by the RI or RIX routine in the resident module.

Parameters: FILENAME - must be 5 character ASCII string stored in location FIELD with associated unit number as a hex value stored in location DRIVE.

Comments: The A register indicates status

Ø - successful

Not Ø - failure

The location of field is start +7B HEX.
The location of drive is start +8Ø HEX.

4-8 OPENW (open existing file for output)

Function: A file is opened for writing by the WRT routine in the Resident Module.

Parameters: See 4-7, PENR.

Comments: See 4-7, OPENR.

4-9 OPENX (open new file for output)

Function: Opens a new file located at the first unused disk address for writing, using WRT routine of the Resident Module.

Parameters: See 4-7, OPENR.

Comments: If the attempted was successful, control will be returned to the calling routine. If unsuccessful, control is passed to the FDOS-III command input routine after the appropriate error message has been displayed on the console.

OPENX, besides opening a new output file, also functions as a device to temporarily store input file pointers. This allows the pointers for a previously opened file temporarily saved while a program object file is opened and loaded to memory or to facilitate multiple passes on a given input file. Restoration is accomplished by calling routine RESTR in the Resident Module.

4-10 RDSCT (read sector)

Function: Reads a sector of data from the diskette in the specified drive to a specified location in memory.

Parameters: A - register, Bits Ø thru 5 contain sector number, bits 6 & 7 contain drive number.

B - register, track number (hex)
H & L - registers, buffer address

Comments: On return, the contents of the A=register indicates the results of the attempted read operation.
Ø = successful - NOT Ø = unsuccessful

A drive not ready condition or the occurance of a CRC error will result in control being returned to the FDOS-III command processor after the appropriate message is displayed on the console device.

4-11 WTSCT (write a sector)

Function: Writes a sector of data to the diskette in the specified drive from the specified location in memory.

Parameters: See 4-10, RDSCT.

Comments: See 4-10, RDSCT.

4-12 DISK HANDLER ROUTINES

Disk handler routines are in relocatable binary format and are on file DKHN of the system diskette. These routines may be integrated into a user program by the link command. The code must be given in relocatable format.

NOTE

Because the data in these routines is routed through the hardware buffers in the controller (one input and one output), care should be taken when attempting to read or write more than one input file and one output file at a time. The suggested procedure is:

. Establish separate internal buffers, RAM areas of 128 bytes for each file opened, and save areas for each files pointers:

ITRK,ISCTR,ISIZE,ICNTR,OCNTR,OTRK,OSCTR, or OSIZE

The disk address pointers must be saved after opening and accessing a file. Each access must be 128 calls to RI or WRT ensuring that the buffer is emptied. The user then utilizes the data in a files buffer rather than calling RI or WRT.

4-13 UPDAT (update)

Purpose: Closes current output file and makes entry in the directory.

Parameters: None

Comments: File must have been opened by RUNGO (implied) command or a call to OPENX

The A register indicates status

Ø = Successful

Not Ø = Failure

4-14 OPENR (open existing file for reading)

Purpose: Opens an existing file that will be read by the user generated code.

Parameters: FILENAME - a five character ASCII string. Stored in location, FIELD prior to calling OPENR.

Comments: The A register indicates status

Ø = successful

Not Ø = failure, Generally because specified file is non-existant.

Calls to location RI (see 4-2) also may be used to read the opened file. The data is entered in the A register when diskread (RI) routine is completed.

NOTE

Only one input file may be opened at a time. An exception would be if special provisions are made in the user code. This would include saving and maintaining the input disk parameters (refer to 4-1). It would also include saving any remaining data of the buffers, during the second read operation of the input file.

4-15 OPENW (open existing file for writing)

Purpose: Opens an existing file, allowing data to be written to it.

NOTE

Data in output file will be lost, unless special actions are taken to save it.

Parameters: Filename, a five character ASCII string. Stored in location FIELD prior to calling OPENW.

Comments: The A register indicates status

Ø = successful

Not Ø = failure, Generally because specified file is non-existant.

4-16 OPENX (open new file for writing)

Purpose: Opens a new output file for writing

Parameters: Filename, a five character ASCII string. Stored in location FILED prior to calling OPENR.

Comments: File will be placed on the first available disk space.

The A register indicates status
Ø = successful
Not Ø = failure

NOTE

Care should be exercised when attempting to read or write more than one file.

Establish separate internal buffers (RAM areas 128 bytes) for each file.

4-17 RDSCT (read=A=Sector)

Purpose: Reads a specified sector of a diskette.

Parameters: B - register, track number (hex)
A - register, sector (hex) - Bits 0-5
Drive - Bits 6-7
H and L - register, buffer address

Comments: A - register indicates status
Ø = Successful
Not Ø = Failure

If a drive is not ready or a CRC error occurs, control returns to FDOS-III command after displaying an error message.

4-18 WTSCT

Purpose: Writes 128 bytes to specified sector of a diskette.

Parameters:

B - register, track number (hex)
A - register, sector (hex) - Bits 0-5
Drive - Bits 6-7
H and L register, buffer address

Comments: In this routine when using the DKHN command two internal labels are required, ERFLG and ERRTN.

ERFLG - one byte area that stores error code prior to error message.

NOTE

ERFLG must be first initialized to Ø before calling any disk handler routine.

ERRTN - two byte area that contains error condition routine address. This routine is given control by a jump command when one of these errors appear.

ERFLG value	PROBLEM
1	Media error
2	Drive not ready
3	Duplicate filename
4	Insufficient disk space

SECTION V

COMMANDS

5-1 FDOS-III COMMANDS

When an exclamation (!) appears on the console output device, FDOS-III is waiting for a directive. These commands are listed in alphabetical order in this section. Also a summary list of the commands is included for quick reference.

All console numeric data is displayed in decimal. Input numeric command data may be decimal or hex. A command to a drive not in a ready state will: cause DISK NOT READ to be displayed on the console device, have the console alarm ring, and retry to access after a one second delay. These action will continue until the drive becomes ready and resumes normal operation. or until the not-ready message has repeated ten times. Then, FDOS-III returns to the command input mode.

5-2 ALLOC

Format: ALLOC,filesize,filename(CR)

Purpose: Creates a designated file name in the directory and allocates file space.

Comments: The filesize is specified in decimal with a minimum size of 1 sector.

Example: ALLOC,31,JACK(CR)

Creates a new file directory entry with the file name JACK, that has attributes of ØØ, and allocated disk space of 31 sectors.

5-3 ASMB

Format: ASMB,sourcefilename,objectfilename.passoption(CR)

Purpose: Assembles contents of the source file. Directs the assembled object output to the object output file, and directs assembled listing to a list device or to a disk file.

Comments: All three operands must be specified. In no object or listing file is to be created, any dummy file name (e.g.X, Y or Z) may be entered in this operand field since no file directory entry will be created.

5-4 BATCH

Format: BATCH(CR)

Purpose: Executes the directives in the file name Batch residing on the systems diskette in Drive Ø.

Comments: Batch Command requirements:

. Resident on drive Ø diskette

. Contains valid directives

Contents of file Batch is treated as if it is entered by console and each directive is executed in order.

```
Example:      BATCH(CR)       BATCH FILE CONTENT

                              RENAM, EXEC, EXECX
                              MERGE, EXEC, EXEC:1
                              DELET:1, EXEC
```

File EXEC is renamed EXECX, and File EXEC on Drive 1 is merged to new file EXEC on Drive Ø. File EXEC is deleted from Drive 1 and control is returned to the console.

Batch messages may be inserted in the Batch File before or after any command in the following format:

"X.......X"(CR) Where X.....X is a string of ASCII characters of any length.

The message text may contain any ASCII character, except the double quotation mark (22H). The text must be enclosed by double quotation marks (22H). These terminating quotation marks should be followed by a carriage return.

When the message is encountered, the console alarm will be sounded, the message displayed, and processing delayed until any character is entered from the console device.

To terminate BATCH, enter CTL-B (Ø2H) from the console. The current directive will be completed, the batch mode terminated and control returned to the console.

5-5 CHGAT

Format: `CHGAT,filename,newattributes(CR)`

Purpose: Changes present attributes of designated file to those specified in the new attributes operand.

Comments: See 2-19.

Examples:

```
CHGAT,MAIN,1
```

Set the attributes of file Main to Ø1, thus setting it as a permanent, non-deletable, file.

```
CHGAT,MAIN,Ø(CR)
CHGAT,MAIN(CR)
```

Set the attributes of file Main to ØØ, thus placing no restrictions on its use or access.

5-6 COPY

Format: `COPY(CR)`

Purpose: Copies contents of diskette in drive Ø onto diskette in drive 1.

Comments: This is a one-for-one image copy; therefore, the contents of either diskette need not be of FDOS-III format.

If any sector of the source diskette is determined bad after 5 read tries, a message will appear on the console giving the operator the option of continuing or aborting the copy process.

NOTE

COPY is a separate program and must reside on a diskette in the specified drive.

Example: COPY:1

5-7 DELET & PACK

Format: DELPK:unitnumber,filename1,filename2,...,filenameN(CR)

Purpose: To delete the designated, non-permanent, files from the diskette, in the specified drive unit, and then to repack the contents of that diskett's user file area and file directory area, thus making the disk space available for additional files.

Comments: The file names need not be in any specific order.

The unit number referes to the drive unit in which the diskette, with the specified files to be deleted, is loaded. The unit number may be Ø, 1, 2, or 3. If the unit number is omitted, Ø is assumed.

Examples: DELPK :2,JOE1,JOE7,AL,SAM,JACK(CR)

Deletes the specified files from the diskette loaded into drive unit 2.

DELPK :Ø,JOE1,JOE7,AL,SAM,JACK(CR)
DELPK ,JOE1,JOE7,AL,SAM,JACK(CR)

Deletes and packs the specified files from the diskette loaded into drive unit Ø.

5-8 DELET

Format: DELET:unitnumber,filename1,filename2,...,filenameN(CR)

Purpose: To delete the designated non-permanent files from the directory of the diskette in the specified drive.

Comments: Until the DELPK or PACK directive is issued the deleted files will appear in the directory listing but will be inaccessible to FDOS-III. The deleted file may be re-activated by using the CHGAT command changing the attribute to Ø or 1.

NOTE

Interruption of the pack function when using CTL-C is not recommended as results are indeterminate.

5-9 DUMP

Format: DUMP,filename,B(CR)

Purpose: To dump the contents of the specified file to the designated punch device, and data is assumed to be non-hex-ASCII.

Comments: Leader and trailer (blank) paper tape is produced when applicable. If optional qualifier B is ommitted, data isassumed to be ASCII and the first occurrence of ASCII CTL-Z (Hex 1A) will be interpreted as EOF terminating the operation.

Example: DUMP,MAIN(CR)

Transfers the contents of file MAIN to the punch output device.

5-10 EDIT

Format: EDIT,inputfilename,newoutputfilename(CR)

Purpose: To enable editing of the contents of the input file, using the FDOS-III Text Editor. Edited data is stored into the new output file.

Comments: Data to be edited is brought from the disk input file into the text editor's RAM buffer by using the editor's A command. Edited data is transferred from the text editor's RAM buffer to the disk output file by using the editor's P Command. The edit operation is terminated, the file directory updated, and control returned to FDOS-III when the editor's E command is executed.

Example: EDIT,BOB1,BOB2(CR)

Establishes a new file BOB2 which will receive the data edited from the contents of the existing file BOB1.

Example: EDIT,,BOB3(CR)

Established a new file BOB3 which will receive data only through the console via the editor insert function.

5-11 EXIT

Format: EXIT(CR)

Purpose: Returns control back to the microcomputer's debug or monitor program.

5-12 HOME

Format: HOME,unitnumber(CR)

Purpose: To position the disk head, on the specified drive unit, to track Ø.

Comments: The unit number may be Ø, 1, 2, or 3. If the unit number is omitted, Ø is assumed.

Examples: HOME,2(CR)

Returns the disk head, on drive unit 2, to track Ø.

HOME(CR)
HOME,(CR)
HOME,Ø(CR)

Returns the disk head, on drive unit Ø, to track Ø.

5-13 INIT

Format: INIT,unitnumber(CR)

Purpose: Initializes the file directory area on the specified drive diskette.

Comments: The unit number may be 1, 2, 3, or 99, where 99 specifies drive unit Ø.

All existing files are cleared from the specified file directory, and all system I/O data is set to FFFF hex or FF hex (not specified). This command must be used to prepare any non-FDOS-III or FDOS-II diskette for use.

NOTE

INIT creates a user diskette and should generally not be used for a system diskette.

Examples: INIT,1(CR)

Initializes file area of diskette in drive unit 1.

INIT,99(CR)

Initializes file area of diskette in drive unit Ø.

5-14 LIST

Format: LIST,unitnumber,listdevice,MODE

Purpose: Prints out contents of a specified diskette file directory on the specified device (c=console, l=line printer, default is console). Lists the filename, attributes, file's starting tract and sector, and the file's size in sectors. LIST also indicates the number of free sectors remaining on the diskette, and the volume name.

Comments: The unit number may be Ø, 1, 2, or 3. If the unit number is omitted, Ø is assumed.

Examples: LIST,1(CR)

Lists the file directory of the diskette in drive 1.

LIST(CR)
LIST,(CR)
LIST,Ø(CR)

Lists the file directory of the diskette in drive unit Ø.

LIST,,,X Where X is any ASCII character other than space.

Lists the first 11 entries of the diskette file directory in drive Ø.

Subcommand N causes the next 11 entries to be listed.

Subcommand P causes the preceding 11 entries to be listed.

The list function is terminated by CR or by FDOS-III when the end of directory is encountered.

LIST,1,L,X

Causes directory of diskette in drive 1 to be listed on the print device 11 entries at a time.

5-15 LOAD

Format: LOAD,newfilename,B(CR)

Purpose: To create the specified file entry and to transfer the contents of the reader input device into that file.

If Parameter "B" (Binary) is not specified, the LOAD function will be terminated by the first CTL-Z (Hex 1A) Read.

If Parameter "B" is specified, the LOAD function will be terminated when a call to the paper tape reader driver returns with the carry bit set. This condition is interpreted by FDOS-III as EOF.

5-16 MERGE

Format: MERGE,newfilename,filename1,filename2,.....,filenameN(CR)

Purpose: Creates a new file whose contents is the concatenation of the contents of the specified hex-ASCII files, in the order in which they appear in the command. Data is not altered in the transfer process.

Comments: The existing files are unaffected.

Examples: MERGE,MAIN,SUB1,SUB2,SUB3(CR)

Creates the new file MAIN with the contents of files SUB1, SUB2, and SUB3, in that order.

MERGE,MAINC,MAIN(CR)

Copies the contents of file MAIN into a new file MAINC.

5-17 MERGB Same as MERGE except file data is assumed to be binary.

NOTE

MERGE assumes a CTL-Z to denote EOF and upon encountering will terminate reading of that file.

MERGB ignores CTL-Z thus transferring the entire contents of the file including the filler nulls in the last sector of each file.

5-18 PACK

Format PACK:unitnumber(CR)

Purpose: Eliminates deleted filenames from the directory of the specified drive and packs the remaining files on the diskette thus making more file space available to the user.

NOTE

See DELET and PACK regarding process interruption.

5-19 PAUSE

Format: PAUSE(CR)

Purpose: Used primarily as a command in the Batch File to halt sequence of executions when operator attention is required.

Comments: Bell rings once.

Example: PAUSE(CR)

Bell rings and computer waits for any character to be entered from console.

5-20 PRINT

Format: PRINT,filename,linesperframe,beginninglinenumber, listdevice(CR)

Purpose: Prints the contents of the specified file to the designated list device.

Lines-per-frame defaults to 9999+.
Beginning-line-number defaults to Ø.
Listdevice defaults to lineprinter.
Listdevice = C causes output to be directed to the console device.

See View command for explanation of keys N, P, F, B.

5-21 RENAM

Format: RENAM,oldfilename,newfilename(CR)

Purpose: Modifies specified file directory entry by replacing its existing file name with a new file name.

Comments: Only the file name of the file directory entry is affected.

Example: RENAM,MAIN5,MAIN(CR)

Renames the file MAIN5 with the name MAIN.

5-22 RUN

Format: RUN,objectfilename,offsetbias(CR)

Purpose: To load the contents of the object file into RAM memory for execution. The data is loaded into memory at locations which are the sum of the memory address specified in the object file plus the offset bias.

Comments: The offset bias address is specified in decimal or hex. If omitted, the offset bias is equal to Ø.

Following the loading of the object file, control will return to the microcomputer's debug or monitor program, if no auto-start address exists in the object file, or to the specified auto-start address if it exists.

Examples: RUN,MAIN(CR)
RUN,MAIN,(CR)
RUN,MAIN,Ø(CR)

Loads the contents of the object file MAIN into RAM memory with an offset bias of Ø.

RUN,MAIN,6144Ø(CR)

Loads the contents of the object file MAIN into RAM memory with an offset bias of FØØØ hex.

RUN,MAIN,1ØØH
Loads the contents of the file MAIN into RAM with an offset of 1ØØ HEX.

5-23 Rungo (Implied)

Format: Hexobjectfilename,inputfilename,outputfilename,N(CR)

FDOS-III assumes any illegal command to be the name of a Hex-ASCII program file, and will search the directory of the specified diskette for that filename. If not found the No Such File error message will be displayed and control returned to the FDOS-III command line processor.

Purpose: To load the contents of the object file into RAM memory for execution. In addition, inputfilename and outputfilename are opened and the number N is converted to Hex and is placed in location PASS. After loading, program control is transferred to memory location ASMB.

Comments: Any or all of the last three fields may be omitted. If an omitted field is followed by a supplied field, the correct number of commas must exist in the command line.

N may be any decimal number from Ø to 255.

By default, inputfile parameters are indeterminate, outputfile parameters are track=76 sector=1 size=1, and N parameter is Ø.

To update the directory following outputs to outputfilename, the user program must perform a JUMP to location UPDAT in the FDOS PROM driver.

Examples: MAIN(CR)

Loads the contents of the object file MAIN into RAM memory and transfer program control to memory location ASMB.

ICE8Ø,LOADF,SAVEF(CR)

Opens the input file LOADF, creates and opens the output file SAFEF, loads the contents of file ICE8Ø into RAM memory, and transfers program control to memory location ASMB.

TRY,,,7(CR)

Sets memory location PASS to 7, then loads the contents of file TRY into RAM memory and transfers program control to memory location ASMB.

NOTE

To "rewind" the input file, the user's program should perform a CALL RESTR, where RESTR is in the FDOS-III Resident (see Appendix).

Following completion of output to the output file, the user should terminate with a JMP UPDAT, where UPDAT is in the FDOS-III Resident (see Appendix). This JMP loads the FDOS-III Exec into RAM and updates the output file's directory entry.

5-23 SYSGN

Format: SYSGN(CR)

Purpose: Permits alteration of initialization data, thus allowing flexibility in the I/Ø routines. When in SYSGN mode, data is accepted from the console and is stored on the systems area of the diskette in drive Ø (system diskette). This is generally done to set up I/O vectors; and/or enter volume name.

Comments: When FDOS-III Executive is loaded and executed the system area of drive Ø diskette (system diskette) will be examined. The data supplied in the SYSGN function will be used to specialize FDOS-III.

NOTE

If the data in the systems area is FFFFH or FFH the system specialization will not be activated.

SYSGN procedure

* Enter: SYSGN(CR)

On Console

ICOM SYSTEM I/O GENERATION(N/R/F)

* Enter: N - New data

R - Revise existing data

F - FDOS-III return

* Each title will appear sequentially after a (CR) is entered.

To alter data enter new data followed by a (CR). If data is not to be changed, enter a (CR).

To terminate, enter: ESC (escape)

Title	Format	Example
VOLUME NAME	Up to 21 ASCII character	ALTAIR2S10
CONSOLE INPUT VECTOR	Hex address	ØØ11
CONSOLE OUTPUT VECTOR	Hex address	ØØ10
READER VECTOR	Hex address	ØØ11
PRINTER VECTOR	Hex address	ØØ10
PUNCH VECTOR	Hex address	ØØ10
MONITOR RE-ENTRY VECTOR	Hex address	C3E4
I/O INITIALIZATION VECTOR	Hex address	ØØØØ
HIGH MEMORY ADDR	Hex address	7FFF
CONSOLE STATUS PORT	Hex value	1Ø
CONSOLE DATA PORT	Hex value	11
INPUT DATA AVAIL MASK	Hex value	Ø1

Title	Format	Example
INPUT DATA AVAIL STATE (HI=Ø1)	Hex value	Ø1
LINE PRINTER WIDTH	Hex value	4F
OBJECT CODE LOAD ADDR	Hex address	ØØØØ
NO. OBJECT CODE BYTES	Hex value	1B

BYTE NO. ØØ =
BYTE NO. Ø1 =
ETC.

Object code, up to 256 bytes may be entered.

NOTE

The number of object code bytes are counted while the code is being entered. Each byte of code must be altered or defaulted (by a CR) to ensure the counter is properly updated.

5-25 VIEW

Format: VIEW,filename,linesperframe,firstline,listdevice

Purpose: To display, onto the console device, the contents of the specified file one frame at a time. The number of lines per displayed frame, if not specified, is 2Ø by default. The first line displayed is line Ø, if not specified otherwise.

Comments: Lines per frame and/or first line number may be omitted; and if so, are assumed to be 2Ø and Ø respectively. All numbers are in decimal.

Listdevice is the output device. L - Line Printer. Default is console.

When in the VIEW command, the following four keys may be used:

nN Causes the next frame to be displayed

nP Causes the previous frame to be displayed.

When n is any decimal value from 1 to 65535 and is the value added (or subtracted) to current line number to determine next line to be displayed.

F Causes the first frame to be displayed (i.e. that frame whose first line is "first line").

B Causes the beginning frame to be displayed (i.e. that frame whose first line is Ø).

CR (Carriage Return) Returns to FDOS-III Executive.

5-26 XGEN

Format: XGEN,filename(CR)

Purpose: To generate the system region of a system diskette (See 2-12) in drive unit Ø from the copy of the FDOS-III Executive which is loaded into the reader input device if filename is omitted.

Comments: This command is used primarily to generate new system diskettes as new versions of the FDOS-III Executive become available or when no system diskette exists.

All I/O data is set to FFFFH or FFH (non-specified) and must be reset using the SYSGN command after using the XGEN command.

If no system diskette exists, one can be generated as follows:

1. Load the copy of the FDOS-III Executive into RAM memory and execute it at memory location start.
2. Insert a new diskette into drive unit Ø.
3. Place a copy of the FDOS-III Executive into the reader input device and enter:

```
XGEN(CR)
```

or place a diskette with a copy of the FDOS-III Executive in Drive 1 and type:

```
XGEN,EXEC:1(CR)
```

APPENDIX A

COMMAND GLOSSARY

ALLOC,filename

creates the designated filename in the directory and allocates disk space equal to size.

ASMB,sourcefilename,destinationfilename,P

assembles the contents of the source file and directs the object to the destination file. P is the pass number which determines whether the assembly should produce a listing only, object only, or both.

BATCH

causes the execution of FDOS-III directives contained in the file named BATCH in Drive Ø.

CHGAT,filename,newattributes

changes the present attributes of the designated file to those specified in the new attributes filed.

COPY

copies the contents of the diskette in drive Ø onto the diskette in drive 1.

DELPK:u,filename1,filename2,.......,filenamen

deletes the designated files from the diskette in drive unit u, and then repacks the contents of that diskette, making the disk space available for additional files.

DELET:u,filename1,filename2,.......,filenamen

deletes the designated files from the diskette in drive unit u.

DUMP,filename,B

dumps the contents of the file to the punch output storage device.

EDIT,inputfilename,outputfilename

enables editing of the input file's contents. Edited data is stored into the output file.

EXIT

returns to the microcomputer system monitor.

HOME,u

positions the disk head on drive unit u to track Ø.

INIT,u

initializes the file directory on the diskette in drive unit u. Clears any existing user files on that diskette.

LIST,u,d,m

lists the contents of the file directory on the diskette in drive unit u. Lists the filenames, attributes, and file sizes in sectors, on device d (CRT or LIST)

LOAD,destinationfilename,B,filesize

loads the contents of the reader device into the specified file on diskette.

MERGE,newfilename,filename1,filename2,.......,filenamen

creates a new file which is a concatenation of filenames 1-n, in that order.

PACK:u

packs the contents of the diskette in drive unit u eliminating all deleted files from the directory and the file space.

PRINT,filename,linesperframe,beginninglinenumber,listdevice

prints the contents of the file on the list output device.

RENAM,oldfilename,newfilename

renames the old file with the new filename.

RUN,filename,offsetbias

loads the contents of the file into RAM for execution.

* Rungo hexobjectfilename,inputfilename,outputfilename,n

sets up the specified input, output, and n parameters, if given; loads the contents of the hex object file into RAM for execution; and then does a branch to location.

SYSGN permits alteration of initialization data.

VIEW,filename,linesperframe,firstline,listdevice

displays the contents of the specified file one frame at a time.

XGEN,filename

enables system generation of other iCOM FDOS versions as might become available in the future.

* Implied

APPENDIX B

MINIMONITOR

B-1 MINIMONITOR

NOTE

> The Minimonitor is only available for microcomputers employing an Altair^{tm} S-100 bus structure.

The minimonitor is called when C3E4 is executed. Its prompt character is a greater than sign >.

The minimonitor accepts three commands:

- Goto
- Memory display/alter
- Memory examine

B-2 GOTO

The goto function directs the program to a desired execution address.

Format: GXXX(CR)

where XXXX is the execution address.

Data of the current memory location can be altered by entering two hex characters. The next location of memory will then be displayed.

B-3 MEMORY DISPLAY/ALTER

Memory display/alter allows a RAM location to be displayed

Format: MXXX(CR)

where XXXX is displayed location.

To change memory contents:

- Enter two hex characters for each byte.
- The next memory contents are displayed.

  To simply view the memory contents, press the space bar to display the next memory location.
- To terminate function; press carriage return (CR).

B-4 MEMORY TEST

```
Format:     TXXXX,YYY(CR)

where XXXX low RAM address
      YYYY high RAM address.
```

Memory failure will appear on the console. The items displayed are: the address of the failure, the data written, and the data read.

```
Format:     XXXX = YY ZZ

where XXXX - Addressing failed location
        YY - Data written
        ZZ - Data read.
```

APPENDIX C

DIAGNOSTIC LISTING

```
;PROM RESIDENT DIAGNOSTIC
;
;
;FDOS III DIAGNOSTIC FOR ALTAIR/IMSAI BASED SYSTEMS
;
; START AT LOCATION 100H

;LOAD A SCRATCH DISKETTE INTO THE DRIVE UNIT TO
;    BE TESTED

;TYPE THE DESIRED TEST TO BE PERFORMED

;CONTINUOUS TESTS MAY BE MANUALLY ABORTED

;BY PRESSING "CTL-C"


;U=UNIT NUMBER  0(OR NOTHING), 1, 2, OR 3
;T=TRACK
;S=SECTOR

;A      -CLEAR DRIVE ELECTRONICS
;BU,T   -SEEK TO TRACK
;DU,S   -READ TO BUFFER FROM PRESENT TRACK
;FU,S   -WRITE FROM BUFFER TO PRESENT TRACK
;GU,S   -RD/WRT (BFR) CONTINUOUS ON PRESENT TRACK
;HU     -TRK0 TO TRK76 LOOP
;I      -UNIT SELECT TEST
;JU     -SEEK TEST ONCE(2 MIN)
;KU     -SEEK TEST CONTINUOUS
;LU     -SEEK TEST READ ONLY
;MU     -DD MARK TEST ONCE
;N      -RETURN TO MONITOR

;LIST OF ERRORS

;01 - CRC ERROR ON READ 5 TIMES - 01(TRK)(UNIT/SCTR)
;02 - CRC ERROR ON WRITE 5 TIMES - 02(TRK)(UNIT/SCTR)
;03 - RD/WRT DATA ERROR - (REC'D)(EXP'D)(BYTE#)
;04 - UNIT SELECT ERROR - (REC'D)(EXP'D)
;05 - SEEK ERROR - (REC'D)(EXP'D)(TRK)(SCTR)
;06 - DD MARK ERROR - (SCTR)
;07 - DD MARK ERROR ON RD/WRT


;BUFFER = 1000H - 107FH
```

```
                    ;
                    ;
C000                PROM      EQU       0C000H
                    ;*****
                    ;
                    ;RESIDENT 8080 ALTAIR/IMSAI/POLY 88 FDOS III  VERSION 1.0
                    ;
                    ;*****
                    ;
                    ;
                    ;ENTRY ADDRESSES-
                    ;         POWER UP = C3E7 HEX
                    ;         RE-ENTRY = C3E4 HEX
                    ;
                    ;
00C1                DATAO     EQU       0C1H
00C0                DATAI     EQU       0C0H
00C0                CNTRL     EQU       0C0H
0000                CCTRL     EQU       0         ;CONSOLE CONTROL PORT
0001                CDATA     EQU       1         ;CONSOLE DATA PORT
0001                CRRDY     EQU       1         ;CONSOLE DATA READY
0080                CTRDY     EQU       80H       ;CONSOLE XMIT READY
                    ;
                    ;
C400                SCTCH     EQU       0C400H    ;SCRATCH RAM
C400                VCTRS     EQU       SCTCH     ;I/O VECTORS
C430                BASE      EQU       SCTCH+30H
C47F                STACK     EQU       SCTCH+7FH
                    ;
                    ;
C430                PASS      EQU       BASE
C431                OFILE     EQU       BASE+1
C432                OUNIT     EQU       BASE+2
C433                IUNIT     EQU       BASE+3
C434                ISIZE     EQU       BASE+4
C436                ITRK      EQU       BASE+6
C437                ISCTR     EQU       BASE+7
C438                ICNTR     EQU       BASE+8
C439                OSIZE     EQU       BASE+9
C43B                OTRK      EQU       BASE+11
C43C                OSCTR     EQU       BASE+12
C43D                OCNTR     EQU       BASE+13
C42F                TITRK     EQU       BASE-1
C43E                TISZE     EQU       BASE+14

C418                ASMB      EQU       VCTRS+24
C41B                START     EQU       VCTRS+27
C41E                UPDTX     EQU       VCTRS+30
```

```
              ;
              ;
C000                  ORG     PROM
              ;
              ;
              ;
              ;ENTRY POINT WHEN Q IS TYPED
              ;LOADS FDOS AND BRANCHES TO FDOS S.A.

C000 C315C0           JMP     FDOS

C003 C300C4   CI:     JMP     VCTRS   ;KEYBOARD INPUT VECTOR

C006 C303C4   CO:     JMP     VCTRS+3 ;CONSOLE OUTPUT VECTOR

C009 C306C4   RDRIN:  JMP     VCTRS+6 ;READER INPUT VECTOR

C00C C309C4   LO:     JMP     VCTRS+9 ;LIST OUTPUT VECTOR

C00F C30CC4   PO:     JMP     VCTRS+12        ;PUNCH OUTPUT VECTOR

C012 C30FC4   MNTR:   JMP     VCTRS+15        ;SYSTEM MONITOR VECTOR

C015 317FC4   FDOS:   LXI     SP,STACK
C018 CD5EC0           CALL    FDOS1
C01B C31BC4           JMP     START
              ;
C01E C354C0   RSTV:   JMP     RESET
C021 C3E0C1   XUSV:   JMP     XUS
C024 C3EFC1   XXUSV:  JMP     XXUS
C027 C3F7C1   SEEKV:  JMP     SEEK+1
C02A C303C2   RFLGV:  JMP     RFLAG
C02D C305C2   LOOPV:  JMP     LOOP
C030 C37AC0   RSTRV:  JMP     RESTR
              ;
              ;
              ;
              ;
              ;
C033 C309C1   RIV:    JMP     RI
              ;
C036 C394C1   WRTV:   JMP     WRT
              ;
              ;
C039 C338C2   PASSV:  JMP     IPASS   ;ASMB INTERPASS FNC

C03C CD93C0   ASSEM:  CALL    REDX
C03F CD7AC0           CALL    RESTR
C042 C318C4           JMP     ASMB
```

```
                ;
C045 317FC4     UPDAT:   LXI     SP,STACK
C048 CD5EC0              CALL    FDOS1
C04B C31EC4              JMP     UPDTX
                ;
                ;
C04E CD93C0     PROG:    CALL    REDX
C051 C312C0              JMP     MNTR
                ;
                ;
C054 3E81       RESET:   MVI     A,81H
C056 CD05C2              CALL    LOOP
C059 3E0D                MVI     A,0DH
C05B C305C2              JMP     LOOP
                ;
C05E CD54C0     FDOS1:   CALL    RESET
C061 210000              LXI     H,0       ;SET BIAS=0
C064 E5                  PUSH    H
C065 216900              LXI     H,105
C068 2234C4              SHLD    ISIZE
C06B 2136C4              LXI     H,ITRK
C06E 3601                MVI     M,1       ;TRACK=1
C070 2C                  INR     L
C071 3600                MVI     M,0       ;SECTOR=0
C073 2C                  INR     L         ;READ BFR EMPTY
C074 3600                MVI     M,0
C076 CD93C0              CALL    REDX
C079 C9                  RET               ;GO TO FDOS
                ;
                ;
                ;
C07A 2A3EC4     RESTR:   LHLD    TISZE     ;RESTORE IFILE POINTERS
C07D 2234C4              SHLD    ISIZE
C080 3A2FC4              LDA     TITRK
C083 3236C4              STA     ITRK
C086 3A33C4              LDA     IUNIT
C089 0F                  RRC
C08A 0F                  RRC
C08B 3237C4              STA     ISCTR
C08E 97                  SUB     A
C08F 3238C4              STA     ICNTR
C092 C9                  RET
                ;
                ;
                ;
                ;SUBROUTINE TO READ A HEX FILE INTO MEMORY
                ;STARTS WITH ROUTINE REDO, USES ALL REGISTERS
                ;
                ;
C093 E1         REDX:   POP     H       ;SWAP BIAS & RETURN
C094 E3                 XTHL
C095 E5                 PUSH    H
C096 E1         REDO:   POP     H       ;GET BIAS
C097 E5                 PUSH    H
C098 CD00C1             CALL    RIX     ;GET CHAR INTO A
C09B 063A               MVI     B,':'
C09D 90                 SUB     B
C09E C296C0             JNZ     REDO
C0A1 57                 MOV     D,A
C0A2 CDD7C0             CALL    BYTE
C0A5 CAC8C0             JZ      RED2
C0A8 5F                 MOV     E,A
C0A9 CDD7C0             CALL    BYTE
C0AC F5                 PUSH    PSW
C0AD CDD7C0             CALL    BYTE
C0B0 C1                 POP     B
C0B1 4F                 MOV     C,A
C0B2 09                 DAD     B
C0B3 CDD7C0             CALL    BYTE
C0B6 CDD7C0     RED1:   CALL    BYTE
C0B9 77                 MOV     M,A
C0BA 23                 INX     H
C0BB 1D                 DCR     E
C0BC C2B6C0             JNZ     RED1
C0BF CDD7C0             CALL    BYTE
C0C2 C2D1C2             JNZ     LER
C0C5 C396C0             JMP     REDO
C0C8 CDD7C0     RED2:   CALL    BYTE
C0CB 67                 MOV     H,A
C0CC CDD7C0             CALL    BYTE
C0CF 6F                 MOV     L,A
C0D0 B4                 ORA     H
C0D1 CAD5C0             JZ      RED3
C0D4 E9                 PCHL
C0D5 E1         RED3:   POP     H
C0D6 C9                 RET
                ;
                ;
                ;
C0D7 CD00C1     BYTE:   CALL    RIX
C0DA CDEEC0             CALL    NBL
C0DD 07                 RLC
C0DE 07                 RLC
C0DF 07                 RLC
C0E0 07                 RLC
C0E1 4F                 MOV     C,A
C0E2 CD00C1             CALL    RIX
```

```
C0E5 CDEEC0               CALL    NBL
C0E8 B1                   ORA     C
C0E9 4F                   MOV     C,A
C0EA 82                   ADD     D
C0EB 57                   MOV     D,A
C0EC 79                   MOV     A,C
C0ED C9                   RET
                ;
                ;
                ;SUBROUTINE TO CONVERT TWO HEX CHARACTERS
                ;    TO ONE BYTE
C0EE D630       NBL:      SUI     '0'
C0F0 D8                   RC
C0F1 C6E9                 ADI     0E9H
C0F3 D8                   RC
C0F4 C606                 ADI     6
C0F6 F2FCC0               JP      NIO
C0F9 C607                 ADI     7
C0FB D8                   RC
C0FC C60A       NIO:      ADI     10
C0FE B7                   ORA     A
C0FF C9                   RET
                ;
                ;
                ;
                ;
                ;
                ;SUBROUTNE TO READ A BYTE FROM DISK
                ;PLACES CHAR IN A-REG.  ENTRY AT RI READS 8 BITS,
                ;ENTRY AT RIX READS 7 BITS.
                ;
                ;
C100 CD09C1     RIX:      CALL    RI
C103 DA12C0               JC      MNTR
C106 E67F                 ANI     7FH
C108 C9                   RET
                ;
                ;
C109 C5         RI:       PUSH    B           ;SAVE REG D-L
C10A E5                   PUSH    H
C10B 2138C4               LXI     H,ICNTR
C10E 7E                   MOV     A,M
C10F A7                   ANA     A           ;CNT=0?
C110 C26EC1               JNZ     RI10        ;NO
C113 2E37       RI5:      MVI     L,ISCTR AND 0FFH          ;YES-INCR D.A.
```

```
C115 CD82C1             CALL    INCDA
C118 2A34C4             LHLD    ISIZE
C11B 2B                 DCX     H
C11C 2234C4             SHLD    ISIZE
C11F 7D                 MOV     A,L
C120 A7                 ANA     A
C121 C236C1             JNZ     RI3
C124 7C                 MOV     A,H
C125 A7                 ANA     A
C126 C236C1             JNZ     RI3
C129 23                 INX     H
C12A 2234C4             SHLD    ISIZE
C12D 2138C4             LXI     H,ICNTR
C130 3600               MVI     M,0
C132 37                 STC             ;SET EOF
C133 E1         RI2:    POP     H       ;RESTORE D-L
C134 C1                 POP     B
C135 C9                 RET
                ;
C136 2137C4     RI3:    LXI     H,ISCTR ;XMIT UNIT/SECTOR
C139 CDE0C1             CALL    XUS
C13C CD28C2             CALL    CHK     ;MAKE SURE A DISK
C13F 2C                 INR     L       ;SET CNTR=128
C140 3680               MVI     M,128
C142 0E05               MVI     C,5     ;SET TRY CNT=5
C144 2E36               MVI     L,ITRK AND OFFH ;SEEK TRACK
C146 CDF6C1             CALL    SEEK
C149 3E03       RI6:    MVI     A,3     ;READ DATA
C14B CD05C2             CALL    LOOP
C14E DBC0               IN      DATAI   ;DD MARK?
C150 E680               ANI     80H
C152 CA5BC1             JZ      RI4     ;NO
C155 CD03C2             CALL    RFLAG
C158 C313C1             JMP     RI5
C15B DBC0       RI4:    IN      DATAI   ;CRC ERROR?
C15D E608               ANI     8H
C15F CA6EC1             JZ      RI10    ;NO
C162 CD03C2             CALL    RFLAG
C165 0D                 DCR     C       ;DECR CNTR
C166 C249C1             JNZ     RI6
C169 3E01               MVI     A,1
C16B C32FC2             JMP     CHK1
                ;
C16E 3E40       RI10:   MVI     A,40H   ;READ BYTE INTO A
C170 D3C0               OUT     CNTRL
C172 DBC0               IN      DATAI
C174 4F                 MOV     C,A
C175 3E41               MVI     A,41H   ;STROBE BUFFER
C177 CD05C2             CALL    LOOP
C17A 2E38               MVI     L,ICNTR AND OFFH        ;DECR   READ COUNTER
```

```
C17C 35                 DCR     M
C17D 79                MOV     A,C
C17E B7                 ORA     A
C17F C333C1             JMP     RI2
                ;
                ;
                ;
                ;ROUTINE TO INCREMENT DISK ADDRESS
                ;
C182 34         INCDA:  INR     M
C183 7E                 MOV     A,M
C184 E61F               ANI     1FH
C186 FE1B               CPI     27
C188 CA8DC1             JZ      INCDB
C18B 2D                 DCR     L
C18C C9                 RET
C18D 7E         INCDB:  MOV     A,M
C18E E6C1               ANI     0C1H
C190 77                 MOV     M,A
C191 2D                 DCR     L
C192 34                 INR     M
C193 C9                 RET
                ;
                ;
                ;
                ;SUBROUTINE TO WRITE A BYTE TO DISK
                ;EXPECTS CHAR TO BE IN C-REG
                ;
C194 79         WRT:    MOV     A,C
C195 E5                 PUSH    H
C196 D3C1               OUT     DATAO   ;OUTPUT HAR
C198 3E31               MVI     A,31H
C19A CD05C2             CALL    LOOP
C19D 213DC4             LXI     H,OCNTR
C1A0 34                 INR     M       ;INCREMENT BFR CNT
C1A1 7E                 MOV     A,M
C1A2 FE80               CPI     128     ;=128?
C1A4 C2DEC1             JNZ     WRT4    ;NO
C1A7 3600               MVI     M,0     ;CLEAR COUNT
C1A9 213CC4     WRT1:   LXI     H,OSCTR ;XMIT UNIT/SECTOR
C1AC CDE0C1             CALL    XUS
C1AF CD28C2             CALL    CHK     ;MAKE SURE A DISK
C1B2 0E05               MVI     C,5     ;SET TRY CNT=5
C1B4 2D                 DCR     L       ;SEEK TRACK
C1B5 CDF6C1             CALL    SEEK
C1B8 3E05       WRT2:   MVI     A,5     ;WRITE DATA
C1BA CD05C2             CALL    LOOP
C1BD 3E07               MVI     A,7     ;READ FOR CRC
C1BF CD05C2             CALL    LOOP
C1C2 DBC0               IN      DATAI   ;CRC ERROR?
```

```
C1C4 E608                ANI     8H
C1C6 CADBC1              JZ      WRT3     ;NO
C1C9 CD03C2              CALL    RFLAG
C1CC 0D                  DCR     C        ;DECR TRY CNT
C1CD C2B8C1              JNZ     WRT2     ;TRY AGAIN
C1D0 3E0F                MVI     A,0FH    ;WRITE AS DD
C1D2 CD05C2              CALL    LOOP
C1D5 CD11C2              CALL    WRTN     ;INCREMENT DA & CHK SIZE
C1D8 C3A9C1              JMP     WRT1
C1DB CD11C2     WRT3:    CALL    WRTN     ;INCREMENT DA & CHK SIZE
C1DE E1         WRT4:    POP     H        ;RESTORE D-L
C1DF C9                  RET
                ;
                ;
                ;
                ;SUBROUTINE TO TRANSMIT UNIT/SECTOR BYTE
                ;
C1E0 7E         XUS:     MOV     A,M
C1E1 E61F                ANI     1FH      ;EXTRACT LOG SECTOR
C1E3 E5                  PUSH    H
C1E4 215EC2              LXI     H,TBL-1  ;GET TABLE PNTR
C1E7 85                  ADD     L        ;MAKE SECTOR PNTR
C1E8 6F                  MOV     L,A
C1E9 4E                  MOV     C,M      ;GET PHYS SECTOR
C1EA E1                  POP     H
C1EB 7E                  MOV     A,M
C1EC E6C0                ANI     0C0H
C1EE B1                  ORA     C        ;MERGE UNIT & PHYS SCTR
C1EF D3C1       XXUS:    OUT     DATAO
C1F1 3E21                MVI     A,21H
C1F3 C305C2              JMP     LOOP
                ;
                ;
                ;
                ;SUBROUTINE TO SEEK TRACK IN A
                ;
C1F6 7E         SEEK:    MOV     A,M
C1F7 D3C1                OUT     DATAO
C1F9 3E11                MVI     A,11H
C1FB CD05C2              CALL    LOOP
C1FE 3E09                MVI     A,09
C200 C305C2              JMP     LOOP
                ;
                ;
                ;
                ;SUBROUTINE TO RESET FLAG
                ;
C203 3E0B       RFLAG:   MVI     A,0BH
                ;SUBROUTINE TO ISSUE CMD & LOOP ON BUSY
                ;
```

```
C205 D3C0     LOOP:    OUT     CNTRL
C207 97                SUB     A
C208 D3C0              OUT     CNTRL
C20A DBC0     LOOP1:   IN      DATAI
C20C 1F                RAR
C20D DA0AC2            JC      LOOP1
C210 C9                RET
              ;
              ;
              ;
              ;SUBROUTINE TO INCR DISK ADDR & CHK OFILE SIZE
              ;
C211 2E3C     WRTN:    MVI     L,OSCTR AND OFFH
C213 CD82C1            CALL    INCDA
C216 2A39C4   WRTN2:   LHLD    OSIZE
C219 2B                DCX     H
C21A 2239C4            SHLD    OSIZE
C21D 7D                MOV     A,L
C21E A7                ANA     A
C21F C0                RNZ
C220 7C                MOV     A,H
C221 A7                ANA     A
C222 C0                RNZ
C223 3E02              MVI     A,2
C225 C32FC2            JMP     CHK1
              ;
              ;
              ;
              ;SUBROUTINE TO CHECK IF A DISK, ELSE ERR03
              ;
C228 DBC0     CHK:     IN      DATAI
C22A E620              ANI     20H
C22C C8                RZ
C22D 3E03              MVI     A,3
              ;ROUTINE TO PRINT ERR(E)
C22F F630     CHK1:    ORI     30H      ;CONVERT TO ASCII
C231 4F                MOV     C,A
C232 CD06C0            CALL    CO
C235 C312C0            JMP     MNTR
              ;
              ;
              ;
              ;INTERPASS FUNCTIONS
              ;IF BIT 0 OF (PASS) IS EQUAL TO 1, THEN BIT 0 OF
              ;   (PASS) IS SET TO 0 AND 31H, ASCII 1, IS RETURNED IN
              ;   A-REG.  IF BIT 0 OF (PASS) IS EQUAL TO 0, THEN (PASS)
              ;   IS SET TO 00 AND 30H, ASCII 0, PLUS (PASS) SHIFTED
              ;   RIGHT 1 BIT POSITION IS RETURNED IN A-REG.  IF (PASS)
              ;   IS EQUAL TO 00, JMP UPDAT OCCURS.
```

```
C238 3A30C4   IPASS:  LDA     PASS
C23B 1F               RAR
C23C D24BC2           JNC     PASS2
C23F 3A30C4           LDA     PASS
C242 3D               DCR     A
C243 3230C4           STA     PASS
C246 3E01             MVI     A,1
C248 C35CC2           JMP     PASS3
C24B A7       PASS2:  ANA     A
C24C CA45C0           JZ      UPDAT
C24F CD7AC0           CALL    RESTR
C252 3A30C4           LDA     PASS
C255 1F               RAR
C256 F5               PUSH    PSW
C257 97               SUB     A
C258 3230C4           STA     PASS
C25B F1               POP     PSW
C25C C630     PASS3:  ADI     30H
C25E C9               RET

              ;PHYSICAL SECTOR TABLE.  IS IN ORDER
              ;    OF LOGICAL SECTOR NUMBER.
              ;
C25F 01       TBL:    DB      1
C260 0A               DB      0AH
C261 13               DB      13H
C262 02               DB      2
C263 0B               DB      0BH
C264 14               DB      14H
C265 03               DB      3
C266 0C               DB      0CH
C267 15               DB      15H
C268 04               DB      4
C269 0D               DB      0DH
C26A 16               DB      16H
C26B 05               DB      5
C26C 0E               DB      0EH
C26D 17               DB      17H
C26E 06               DB      6
C26F 0F               DB      0FH
C270 18               DB      18H
C271 07               DB      7
C272 10               DB      10H
C273 19               DB      19H
C274 08               DB      8
C275 11               DB      11H
C276 1A               DB      1AH
C277 09               DB      9
C278 12               DB      12H
```

```
C279 00                 NOP
C27A 00                 NOP
C27B 00                 NOP
              ;
              ;
              ;
              ;
              ;         CONSOLE INPUT ROUTINE
              ;
C27C DB00     CIX:      IN      CCTRL              IN    1ØH
C27E E601               ANI     CRRDY              ANI   Ø1H
C280 C27CC2             JNZ     CIX                JZ    C27C
C283 DB01               IN      CDATA              IN    11H
C285 E67F               ANI     7FH
C287 C9                 RET
              ;
              ;
              ;
C288 0E0D     CRLF:     MVI     C,0DH
C28A CD06C0             CALL    CO
C28D 0E0A               MVI     C,0AH
C28F C306C0             JMP     CO
              ;
              ;         CONSOLE OUTPUT ROUTINE
              ;
C292 DB00     COX:      IN      CCTRL              IN    1ØH
C294 E680               ANI     CTRDY              ANI   Ø2H
C296 C292C2             JNZ     COX                JZ    C292
C299 79                 MOV     A,C
C29A D301               OUT     CDATA         ←    OUT   11H
C29C C9                 RET
              ;
              ;
C29D 010BC3   INIT:     LXI     B,0C30BH
C2A0 2100C4             LXI     H,VCTRS
C2A3 11EAC3             LXI     D,VECTR
C2A6 70       INIT1:    MOV     M,B
C2A7 23                 INX     H
C2A8 1A                 LDAX    D
C2A9 77                 MOV     M,A
C2AA 23                 INX     H
C2AB 13                 INX     D
C2AC 1A                 LDAX    D
C2AD 77                 MOV     M,A
C2AE 23                 INX     H
C2AF 13                 INX     D
C2B0 0D                 DCR     C
C2B1 C2A6C2             JNZ     INIT1
C2B4 317FC4   MNTRX:    LXI     SP,STACK
C2B7 CD88C2             CALL    CRLF
```

```
C2BA 0E3E                MVI     C,3EH
C2BC CD06C0              CALL    CO
C2BF CDD9C2              CALL    CECHO
C2C2 FE54                CPI     'T'
C2C4 CA80C3              JZ      TSTM
C2C7 FE4D                CPI     'M'
C2C9 CA58C3              JZ      MEM
C2CC FE47                CPI     'G'
C2CE CAE1C2              JZ      GO
C2D1 0E3F       LER:     MVI     C,'?'
C2D3 CD06C0              CALL    CO
C2D6 C3B4C2              JMP     MNTRX
                ;
                ;
C2D9 CD03C0     CECHO:   CALL    CI
C2DC 4F                  MOV     C,A
C2DD CD06C0              CALL    CO
C2E0 C9                  RET
                ;
                ;
C2E1 CDE8C2     GO:      CALL    PARAM
C2E4 CD88C2              CALL    CRLF
C2E7 E9                  PCHL
                ;
                ;
C2E8 210000     PARAM:   LXI     H,0
C2EB CDD9C2     PARM1:   CALL    CECHO
C2EE FE0D                CPI     0DH
C2F0 C8                  RZ
C2F1 FE2C                CPI     ','
C2F3 C8                  RZ
C2F4 29                  DAD     H
C2F5 29                  DAD     H
C2F6 29                  DAD     H
C2F7 29                  DAD     H
C2F8 DAD1C2              JC      LER
C2FB CDEEC0              CALL    NBL
C2FE DAD1C2              JC      LER
C301 B5                  ORA     L
C302 6F                  MOV     L,A
C303 C3EBC2              JMP     PARM1
                ;
                ;
C306 CDD9C2     BYTEC:   CALL    CECHO
C309 CDEEC0     BYTC1:   CALL    NBL
C30C 07                  RLC
C30D 07                  RLC
C30E 07                  RLC
C30F 07                  RLC
C310 F5                  PUSH    PSW
```

```
C311 CDD9C2                 CALL    CECHO
C314 CDEEC0                 CALL    NBL
C317 C1                     POP     B
C318 B0                     ORA     B
C319 C9                     RET
                ;
                ;
C31A F5         BYTEO:      PUSH    PSW
C31B CD2AC3                 CALL    BYTO1
C31E 4F                     MOV     C,A
C31F CD06C0                 CALL    CO
C322 F1                     POP     PSW
C323 CD2EC3                 CALL    BYTO2
C326 4F                     MOV     C,A
C327 C306C0                 JMP     CO
                ;
                ;
C32A 0F         BYTO1:      RRC
C32B 0F                     RRC
C32C 0F                     RRC
C32D 0F                     RRC
C32E E60F       BYTO2:      ANI     0FH
C330 FE0A                   CPI     0AH
C332 FA37C3                 JM      BYTO3
C335 C607                   ADI     7
C337 C630       BYTO3:      ADI     30H
C339 C9                     RET
                ;
                ;
C33A CD88C2     HLCO:       CALL    CRLF
C33D 7C                     MOV     A,H
C33E CD1AC3                 CALL    BYTEO
C341 7D                     MOV     A,L
C342 CD1AC3                 CALL    BYTEO
C345 C9                     RET
                ;
                ;
C346 CD3AC3     DSPYM:      CALL    HLCO
C349 0E3D                   MVI     C,'='
C34B CD06C0                 CALL    CO
C34E 7E                     MOV     A,M
C34F CD1AC3                 CALL    BYTEO
C352 0E20                   MVI     C,20H
C354 CD06C0                 CALL    CO
C357 C9                     RET
                ;
                ;
C358 CDE8C2     MEM:        CALL    PARAM
C35B CD46C3     MEM1:       CALL    DSPYM
C35E CDD9C2                 CALL    CECHO
```

```
C361 FE0D               CPI     0DH
C363 CAB4C2             JZ      MNTRX
C366 FE20               CPI     20H
C368 CA6FC3             JZ      MEM9
C36B CD09C3             CALL    BYTC1
C36E 77                 MOV     M,A

C36F 23         MEM9:   INX     H
C370 C35BC3             JMP     MEM1
                ;
                ;
C373 DB00       KBINT:  IN      CCTRL
C375 E601               ANI     CRRDY
C377 C0                 RNZ
C378 DB01               IN      CDATA
C37A FE03               CPI     3
C37C CA12C0             JZ      MNTR
C37F C9                 RET
                ;
                ;
C421            HIGH    EQU     SCTCH+21H

C380 CDE8C2     TSTM:   CALL    PARAM
C383 E5                 PUSH    H
C384 EB                 XCHG
C385 CDE8C2             CALL    PARAM
C388 2221C4             SHLD    HIGH
C38B EB                 XCHG
C38C CD88C2             CALL    CRLF
C38F 3600       TSTM2:  MVI     M,0
C391 7B                 MOV     A,E
C392 BD                 CMP     L
C393 C29BC3             JNZ     TSTM3
C396 7A                 MOV     A,D
C397 BC                 CMP     H
C398 CA9FC3             JZ      TSTM4
C39B 23         TSTM3:  INX     H
C39C C38FC3             JMP     TSTM2

C39F 1E01       TSTM4:  MVI     E,1
C3A1 E1         TSTM7:  POP     H
C3A2 E5                 PUSH    H
C3A3 34         TSTM1:  INR     M
C3A4 7B                 MOV     A,E
C3A5 BE                 CMP     M
C3A6 C4C2C3             CNZ     TSTM6
C3A9 3A21C4             LDA     HIGH
C3AC BD                 CMP     L
C3AD C2BEC3             JNZ     TSTM5
C3B0 3A22C4             LDA     HIGH+1
```

```
C3B3 BC                    CMP     H
C3B4 C2BEC3                JNZ     TSTM5
C3B7 1C                    INR     E
C3B8 CD73C3                CALL    KBINT
C3BB C3A1C3                JMP     TSTM7

C3BE 23           TSTM5:   INX     H
C3BF C3A3C3                JMP     TSTM1

C3C2 CD46C3       TSTM6:   CALL    DSPYM
C3C5 7B                    MOV     A,E
C3C6 CD1AC3                CALL    BYTEO
C3C9 C388C2                JMP     CRLF

                  ;
                  ;
                  ;
C3CC C303C0       RDIX:    JMP     CI
                  ;
C3CF C306C0       LOX:     JMP     CO
                  ;
C3D2 C306C0       POX:     JMP     CO
                  ;
C3E4                       ORG     PROM+3E4H
                  ;
                  ;        I/O VECTOR TABLE
                  ;
                  ;***** MONITOR RE-ENTRY ADDRESS *********
C3E4 C3B4C2                JMP MNTRX
                  ;
                  ;
                  ;****** MONITOR STARTING ADDRESS *********
C3E7 C39DC2                JMP     INIT     ;************

                  VECTR:   DW      CIX      ;CONSOLE IN VECTOR
C3EC 92C2                  DW      COX      ;CONSOLE OUT VECTOR
C3EE CCC3                  DW      RDIX     ;PAPER TAPE READER VECTOR
C3F0 CFC3                  DW      LOX      ;LINE PRINTER VECTOR
C3F2 D2C3                  DW      POX      ;PUNCH VECTOR
C3F4 B4C2                  DW      MNTRX    ;MONITOR VECTOR
C3F6 09C1                  DW      RI       ;DISK READ VECTOR
C3F8 94C1                  DW      WRT      ;DISK WRITE VECTOR
C3FA 4000                  DW      40H      ;ASSEM/EDIT VECTOR;
C3FC 4000                  DW      40H      ;EXECUTIVE VECTOR
C3FE 4300                  DW      43H      ;UPDAT VECTOR

                  ;
                           END
```

Now COLD LOADER

```
ASMB   C418        ASSEM  C03C        BASE   C430        BYTC1  C309
BYTE   C0D7        BYTEC  C306        BYTEO  C31A        BYTO1  C32A
BYTO2  C32E        BYTO3  C337        CCTRL  0000        CDATA  0001
CECHO  C2D9        CHK    C228        CHK1   C22F        CI     C003
CIX    C27C        CNTRL  00C0        CO     C006        COX    C292
CRLF   C288        CRRDY  0001        CTRDY  0080        DATAI  00C0
DATAO  00C1        DSPYM  C346        FDOS   C015        FDOS1  C05E
GO     C2E1        HIGH   C421        HLCO   C33A        ICNTR  C438
INCDA  C182        INCDB  C18D        INIT   C29D        INIT1  C2A6
IPASS  C238        ISCTR  C437        ISIZE  C434        ITRK   C436
IUNIT  C433        KBINT  C373        LER    C2D1        LO     C00C
LOOP   C205        LOOP1  C20A        LOOPV  C02D        LOX    C3CF
MEM    C358        MEM1   C35B        MEM9   C36F        MNTR   C012
MNTRX  C2B4        NBL    C0EE        NIO    C0FC        OCNTR  C43D
OFILE  C431        OSCTR  C43C        OSIZE  C439        OTRK   C43B
OUNIT  C432        PARAM  C2E8        PARM1  C2EB        PASS   C430
PASS2  C24B        PASS3  C25C        PASSV  C039        PO     C00F
POX    C3D2        PROG   C04E        PROM   C000        RDIX   C3CC
RDRIN  C009        REDO   C096        RED1   C0B6        RED2   C0C8
RED3   C0D5        REDX   C093        RESET  C054        RESTR  C07A
RFLAG  C203        RFLGV  C02A        RI     C109        RI10   C16E
RI2    C133        RI3    C136        RI4    C15B        RI5    C113
RI6    C149        RIV    C033        RIX    C100        RSTRV  C030
RSTV   C01E        SCTCH  C400        SEEK   C1F6        SEEKV  C027
STACK  C47F        START  C41B        TBL    C25F        TISZE  C43E
TITRK  C42F        TSTM   C380        TSTM1  C3A3        TSTM2  C38F
TSTM3  C39B        TSTM4  C39F        TSTM5  C3BE        TSTM6  C3C2
TSTM7  C3A1        UPDAT  C045        UPDTX  C41E        VCTRS  C400
VECTR  C3EA        WRT    C194        WRT1   C1A9        WRT2   C1B8
WRT3   C1DB        WRT4   C1DE        WRTN   C211        WRTN2  C216
WRTV   C036        XUS    C1E0        XUSV   C021        XXUS   C1EF
XXUSV  C024
```

APPENDIX D

RESIDENT LISTING

```
;FDOS III DIAGNOSTIC FOR ALTAIR/IMSAI BASED SYSTEMS
;
; START AT LOCATION 100H

;LOAD A SCRATCH DISKETTE INTO THE DRIVE UNIT TO
;   BE TESTED

;TYPE THE DESIRED TEST TO BE PERFORMED

;CONTINUOUS TESTS MAY BE MANUALLY ABORTED

;BY PRESSING "CTL-C"


;U=UNIT NUMBER  0(OR NOTHING), 1, 2, OR 3
;T=TRACK
;S=SECTOR

;A      -CLEAR DRIVE ELECTRONICS
;BU,T   -SEEK TO TRACK
;DU,S   -READ TO BUFFER FROM PRESENT TRACK
;FU,S   -WRITE FROM BUFFER TO PRESENT TRACK
;GU,S   -RD/WRT (BFR) CONTINUOUS ON PRESENT TRACK
;HU     -TRK0 TO TRK76 LOOP
;I      -UNIT SELECT TEST
;JU     -SEEK TEST ONCE(2 MIN)
;KU     -SEEK TEST CONTINUOUS
;LU     -SEEK TEST READ ONLY
;MU     -DD MARK TEST ONCE
;N      -RETURN TO MONITOR

;LIST OF ERRORS

;01 - CRC ERROR ON READ 5 TIMES - 01(TRK)(UNIT/SCTR)
;02 - CRC ERROR ON WRITE 5 TIMES - 02(TRK)(UNIT/SCTR)
;03 - RD/WRT DATA ERROR - (REC'D)(EXP'D)(BYTE#)
;04 - UNIT SELECT ERROR - (REC'D)(EXP'D)
;05 - SEEK ERROR - (REC'D)(EXP'D)(TRK)(SCTR)
;06 - DD MARK ERROR - (SCTR)
;07 - DD MARK ERROR ON RD/WRT


;BUFFER = 1000H - 107FH
```

```
                ;
                ;
C000            PROM    EQU     0C000H
                ;*****
                ;
                ;RESIDENT 8080 ALTAIR/IMSAI/POLY 88 FDOS III  VERSION 1.0
                ;
                ;*****
                ;
                ;
                ;ENTRY ADDRESSES-
                ;       POWER UP = C3E7 HEX
                ;       RE-ENTRY = C3E4 HEX
                ;
                ;
                ;
00C1            DATAO   EQU     0C1H
00C0            DATAI   EQU     0C0H
00C0            CNTRL   EQU     0C0H
0000            CCTRL   EQU     0          ;CONSOLE CONTROL PORT
0001            CDATA   EQU     1          ;CONSOLE DATA PORT
0001            CRRDY   EQU     1          ;CONSOLE DATA READY
0080            CTRDY   EQU     80H        ;CONSOLE XMIT READY
                ;
                ;
C400            SCTCH   EQU     0C400H     ;SCRATCH RAM
C400            VCTRS   EQU     SCTCH      ;I/O VECTORS
C430            BASE    EQU     SCTCH+30H
C47F            STACK   EQU     SCTCH+7FH
                ;
                ;
C430            PASS    EQU     BASE
C431            OFILE   EQU     BASE+1
C432            OUNIT   EQU     BASE+2
C433            IUNIT   EQU     BASE+3
C434            ISIZE   EQU     BASE+4
C436            ITRK    EQU     BASE+6
C437            ISCTR   EQU     BASE+7
C438            ICNTR   EQU     BASE+8
C439            OSIZE   EQU     BASE+9
C43B            OTRK    EQU     BASE+11
C43C            OSCTR   EQU     BASE+12
C43D            OCNTR   EQU     BASE+13
C42F            TITRK   EQU     BASE-1
C43E            TISZE   EQU     BASE+14

C418            ASMB    EQU     VCTRS+24
C41B            START   EQU     VCTRS+27
C41E            UPDTX   EQU     VCTRS+30
```

```
                ;
                ;
C000                      ORG     PROM
                ;
                ;
                ;
                ;ENTRY POINT WHEN Q IS TYPED
                ;LOADS FDOS AND BRANCHES TO FDOS S.A.

C000 C315C0               JMP     FDOS

C003 C300C4     CI:       JMP     VCTRS    ;KEYBOARD INPUT VECTOR

C006 C303C4     CO:       JMP     VCTRS+3 ;CONSOLE OUTPUT VECTOR

C009 C306C4     RDRIN:    JMP     VCTRS+6 ;READER INPUT VECTOR

C00C C309C4     LO:       JMP     VCTRS+9 ;LIST OUTPUT VECTOR

C00F C30CC4     PO:       JMP     VCTRS+12        ;PUNCH OUTPUT VECTOR

C012 C30FC4     MNTR:     JMP     VCTRS+15        ;SYSTEM MONITOR VECTOR

C015 317FC4     FDOS:     LXI     SP,STACK
C018 CD5EC0               CALL    FDOS1
C01B C31BC4               JMP     START
                ;
C01E C354C0     RSTV:     JMP     RESET
C021 C3E0C1     XUSV:     JMP     XUS
C024 C3EFC1     XXUSV:    JMP     XXUS
C027 C3F7C1     SEEKV:    JMP     SEEK+1
C02A C303C2     RFLGV:    JMP     RFLAG
C02D C305C2     LOOPV:    JMP     LOOP
C030 C37AC0     RSTRV:    JMP     RESTR
                ;
                ;
                ;
                ;
                ;
C033 C309C1     RIV:      JMP     RI
                ;
C036 C394C1     WRTV:     JMP     WRT
                ;
                ;
C039 C338C2     PASSV:    JMP     IPASS       ;ASMB INTERPASS FNC

C03C CD93C0     ASSEM:    CALL    REDX
C03F CD7AC0               CALL    RESTR
C042 C318C4               JMP     ASMB
```

```
                ;
C045 317FC4     UPDAT:  LXI     SP,STACK
C048 CD5EC0             CALL    FDOS1
C04B C31EC4             JMP     UPDTX
                ;
                ;
C04E CD93C0     PROG:   CALL    REDX
C051 C312C0             JMP     MNTR
                ;
                ;
C054 3E81       RESET:  MVI     A,81H
C056 CD05C2             CALL    LOOP
C059 3E0D               MVI     A,0DH
C05B C305C2             JMP     LOOP
                ;
C05E CD54C0     FDOS1:  CALL    RESET
C061 210000             LXI     H,0         ;SET BIAS=0
C064 E5                 PUSH    H
C065 216900             LXI     H,105
C068 2234C4             SHLD    ISIZE
C06B 2136C4             LXI     H,ITRK
C06E 3601               MVI     M,1         ;TRACK=1
C070 2C                 INR     L
C071 3600               MVI     M,0         ;SECTOR=0
C073 2C                 INR     L           ;READ BFR EMPTY
C074 3600               MVI     M,0
C076 CD93C0             CALL    REDX
C079 C9                 RET                 ;GO TO FDOS
                ;
                ;
                ;
C07A 2A3EC4     RESTR:  LHLD    TISZE       ;RESTORE IFILE POINTERS
C07D 2234C4             SHLD    ISIZE
C080 3A2FC4             LDA     TITRK
C083 3236C4             STA     ITRK
C086 3A33C4             LDA     IUNIT
C089 0F                 RRC
C08A 0F                 RRC
C08B 3237C4             STA     ISCTR
C08E 97                 SUB     A
C08F 3238C4             STA     ICNTR
C092 C9                 RET
                ;
                ;
                ;
                ;SUBROUTINE TO READ A HEX FILE INTO MEMORY
                ;STARTS WITH ROUTINE REDO, USES ALL REGISTERS
```

```
                  ;
                  ;
C093 E1           REDX:    POP      H           ;SWAP BIAS & RETURN
C094 E3                    XTHL
C095 E5                    PUSH     H
C096 E1           REDO:    POP      H           ;GET BIAS
C097 E5                    PUSH     H
C098 CD00C1                CALL     RIX         ;GET CHAR INTO A
C09B 063A                  MVI      B,':'
C09D 90                    SUB      B
C09E C296C0                JNZ      REDO
C0A1 57                    MOV      D,A
C0A2 CDD7C0                CALL     BYTE
C0A5 CAC8C0                JZ       RED2
C0A8 5F                    MOV      E,A
C0A9 CDD7C0                CALL     BYTE
C0AC F5                    PUSH     PSW
C0AD CDD7C0                CALL     BYTE
C0B0 C1                    POP      B
C0B1 4F                    MOV      C,A
C0B2 09                    DAD      B
C0B3 CDD7C0                CALL     BYTE
C0B6 CDD7C0       RED1:    CALL     BYTE
C0B9 77                    MOV      M,A
C0BA 23                    INX      H
C0BB 1D                    DCR      E
C0BC C2B6C0                JNZ      RED1
C0BF CDD7C0                CALL     BYTE
C0C2 C2D1C2                JNZ      LER
C0C5 C396C0                JMP      REDO
C0C8 CDD7C0       RED2:    CALL     BYTE
C0CB 67                    MOV      H,A
C0CC CDD7C0                CALL     BYTE
C0CF 6F                    MOV      L,A
C0D0 B4                    ORA      H
C0D1 CAD5C0                JZ       RED3
C0D4 E9                    PCHL
C0D5 E1           RED3:    POP      H
C0D6 C9                    RET
                  ;
                  ;
                  ;
C0D7 CD00C1       BYTE:    CALL     RIX
C0DA CDEEC0                CALL     NBL
C0DD 07                    RLC
C0DE 07                    RLC
C0DF 07                    RLC
C0E0 07                    RLC
C0E1 4F                    MOV      C,A
C0E2 CD00C1                CALL     RIX
```

```
C0E5 CDEEC0             CALL    NBL
C0E8 B1                 ORA     C
C0E9 4F                 MOV     C,A
C0EA 82                 ADD     D
C0EB 57                 MOV     D,A
C0EC 79                 MOV     A,C
C0ED C9                 RET
                ;
                ;
                ;SUBROUTINE TO CONVERT TWO HEX CHARACTERS
                ;    TO ONE BYTE
C0EE D630       NBL:    SUI     '0'
C0F0 D8                 RC
C0F1 C6E9               ADI     0E9H
C0F3 D8                 RC
C0F4 C606               ADI     6
C0F6 F2FCC0             JP      NIO
C0F9 C607               ADI     7
C0FB D8                 RC
C0FC C60A       NIO:    ADI     10
C0FE B7                 ORA     A
C0FF C9                 RET
                ;
                ;
                ;
                ;
                ;
                ;SUBROUTNE TO READ A BYTE FROM DISK
                ;PLACES CHAR IN A-REG. ENTRY AT RI READS 8 BITS,
                ;ENTRY AT RIX READS 7 BITS.
                ;
                ;
C100 CD09C1     RIX:    CALL    RI
C103 DA12C0             JC      MNTR
C106 E67F               ANI     7FH
C108 C9                 RET
                ;
                ;
C109 C5         RI:     PUSH    B       ;SAVE REG D-L
C10A E5                 PUSH    H
C10B 2138C4             LXI     H,ICNTR
C10E 7E                 MOV     A,M
C10F A7                 ANA     A       ;CNT=0?
C110 C26EC1             JNZ     RI10    ;NO
C113 2E37       RI5:    MVI     L,ISCTR AND 0FFH        ;YES-INCR D.A.
```

```
C115 CD82C1             CALL    INCDA
C118 2A34C4             LHLD    ISIZE
C11B 2B                 DCX     H
C11C 2234C4             SHLD    ISIZE
C11F 7D                 MOV     A,L
C120 A7                 ANA     A
C121 C236C1             JNZ     RI3
C124 7C                 MOV     A,H
C125 A7                 ANA     A
C126 C236C1             JNZ     RI3
C129 23                 INX     H
C12A 2234C4             SHLD    ISIZE
C12D 2138C4             LXI     H,ICNTR
C130 3600               MVI     M,0
C132 37                 STC                ;SET EOF
C133 E1         RI2:    POP     H          ;RESTORE D-L
C134 C1                 POP     B
C135 C9                 RET
                ;
C136 2137C4     RI3:    LXI     H,ISCTR ;XMIT UNIT/SECTOR
C139 CDE0C1             CALL    XUS
C13C CD28C2             CALL    CHK        ;MAKE SURE A DISK
C13F 2C                 INR     L          ;SET CNTR=128
C140 3680               MVI     M,128
C142 0E05               MVI     C,5        ;SET TRY CNT=5
C144 2E36               MVI     L,ITRK AND OFFH ;SEEK TRACK
C146 CDF6C1             CALL    SEEK
C149 3E03       RI6:    MVI     A,3        ;READ DATA
C14B CD05C2             CALL    LOOP
C14E DBC0               IN      DATAI      ;DD MARK?
C150 E680               ANI     80H
C152 CA5BC1             JZ      RI4        ;NO
C155 CD03C2             CALL    RFLAG
C158 C313C1             JMP     RI5
C15B DBC0       RI4:    IN      DATAI      ;CRC ERROR?
C15D E608               ANI     8H
C15F CA6EC1             JZ      RI10       ;NO
C162 CD03C2             CALL    RFLAG
C165 0D                 DCR     C          ;DECR CNTR
C166 C249C1             JNZ     RI6
C169 3E01               MVI     A,1
C16B C32FC2             JMP     CHK1
                ;
C16E 3E40       RI10:   MVI     A,40H      ;READ BYTE INTO A
C170 D3C0               OUT     CNTRL
C172 DBC0               IN      DATAI
C174 4F                 MOV     C,A
C175 3E41               MVI     A,41H      ;STROBE BUFFER
C177 CD05C2             CALL    LOOP
C17A 2E38               MVI     L,ICNTR AND OFFH          ;DECR   READ COUNTER
```

```
C17C 35             DCR      M
C17D 79             MOV      A,C
C17E B7             ORA      A
C17F C333C1         JMP      RI2
               ;
               ;
               ;
               ;ROUTINE TO INCREMENT DISK ADDRESS
               ;
C182 34        INCDA:   INR      M
C183 7E             MOV      A,M
C184 E61F           ANI      1FH
C186 FE1B           CPI      27
C188 CA8DC1         JZ       INCDB
C18B 2D             DCR      L
C18C C9             RET
C18D 7E        INCDB:   MOV      A,M
C18E E6C1           ANI      0C1H
C190 77             MOV      M,A
C191 2D             DCR      L
C192 34             INR      M
C193 C9             RET
               ;
               ;
               ;SUBROUTINE TO WRITE A BYTE TO DISK
               ;EXPECTS CHAR TO BE IN C-REG
               ;
C194 79        WRT:     MOV      A,C
C195 E5             PUSH     H
C196 D3C1           OUT      DATAO    ;OUTPUT HAR
C198 3E31           MVI      A,31H
C19A CD05C2         CALL     LOOP
C19D 213DC4         LXI      H,OCNTR
C1A0 34             INR      M        ;INCREMENT BFR CNT
C1A1 7E             MOV      A,M
C1A2 FE80           CPI      128      ;=128?
C1A4 C2DEC1         JNZ      WRT4     ;NO
C1A7 3600           MVI      M,0      ;CLEAR COUNT
C1A9 213CC4    WRT1:    LXI      H,OSCTR  ;XMIT UNIT/SECTOR
C1AC CDE0C1         CALL     XUS
C1AF CD28C2         CALL     CHK      ;MAKE SURE A DISK
C1B2 0E05           MVI      C,5      ;SET TRY CNT=5
C1B4 2D             DCR      L        ;SEEK TRACK
C1B5 CDF6C1         CALL     SEEK
C1B8 3E05      WRT2:    MVI      A,5      ;WRITE DATA
C1BA CD05C2         CALL     LOOP
C1BD 3E07           MVI      A,7      ;READ FOR CRC
C1BF CD05C2         CALL     LOOP
C1C2 DBC0           IN       DATAI    ;CRC ERROR?
```

```
C1C4 E608               ANI     8H
C1C6 CADBC1             JZ      WRT3    ;NO
C1C9 CD03C2             CALL    RFLAG
C1CC 0D                 DCR     C       ;DECR TRY CNT
C1CD C2B8C1             JNZ     WRT2    ;TRY AGAIN
C1D0 3E0F               MVI     A,0FH   ;WRITE AS DD
C1D2 CD05C2             CALL    LOOP
C1D5 CD11C2             CALL    WRTN    ;INCREMENT DA & CHK SIZE
C1D8 C3A9C1             JMP     WRT1
C1DB CD11C2     WRT3:   CALL    WRTN    ;INCREMENT DA & CHK SIZE
C1DE E1         WRT4:   POP     H       ;RESTORE D-L
C1DF C9                 RET
                ;
                ;
                ;
                ;SUBROUTINE TO TRANSMIT UNIT/SECTOR BYTE
                ;
C1E0 7E         XUS:    MOV     A,M
C1E1 E61F               ANI     1FH     ;EXTRACT LOG SECTOR
C1E3 E5                 PUSH    H
C1E4 215EC2             LXI     H,TBL-1 ;GET TABLE PNTR
C1E7 85                 ADD     L       ;MAKE SECTOR PNTR
C1E8 6F                 MOV     L,A
C1E9 4E                 MOV     C,M     ;GET PHYS SECTOR
C1EA E1                 POP     H
C1EB 7E                 MOV     A,M
C1EC E6C0               ANI     0C0H
C1EE B1                 ORA     C       ;MERGE UNIT & PHYS SCTR
C1EF D3C1       XXUS:   OUT     DATAO
C1F1 3E21               MVI     A,21H
C1F3 C305C2             JMP     LOOP
                ;
                ;
                ;
                ;SUBROUTINE TO SEEK TRACK IN A
                ;
C1F6 7E         SEEK:   MOV     A,M
C1F7 D3C1               OUT     DATAO
C1F9 3E11               MVI     A,11H
C1FB CD05C2             CALL    LOOP
C1FE 3E09               MVI     A,09
C200 C305C2             JMP     LOOP
                ;
                ;
                ;SUBROUTINE TO RESET FLAG
                ;
C203 3E0B       RFLAG:  MVI     A,0BH
                ;SUBROUTINE TO ISSUE CMD & LOOP ON BUSY
                ;
```

```
C205 D3C0      LOOP:    OUT     CNTRL
C207 97                 SUB     A
C208 D3C0               OUT     CNTRL
C20A DBC0      LOOP1:   IN      DATAI
C20C 1F                 RAR
C20D DA0AC2             JC      LOOP1
C210 C9                 RET
               ;
               ;
               ;
               ;SUBROUTINE TO INCR DISK ADDR & CHK OFILE SIZE
               ;
C211 2E3C      WRTN:    MVI     L,OSCTR AND OFFH
C213 CD82C1             CALL    INCDA
C216 2A39C4    WRTN2:   LHLD    OSIZE
C219 2B                 DCX     H
C21A 2239C4             SHLD    OSIZE
C21D 7D                 MOV     A,L
C21E A7                 ANA     A
C21F C0                 RNZ
C220 7C                 MOV     A,H
C221 A7                 ANA     A
C222 C0                 RNZ
C223 3E02               MVI     A,2
C225 C32FC2             JMP     CHK1
               ;
               ;
               ;
               ;SUBROUTINE TO CHECK IF A DISK, ELSE ERR03
               ;
C228 DBC0      CHK:     IN      DATAI
C22A E620               ANI     20H
C22C C8                 RZ
C22D 3E03               MVI     A,3
               ;ROUTINE TO PRINT ERR(E)
C22F F630      CHK1:    ORI     30H         ;CONVERT TO ASCII
C231 4F                 MOV     C,A
C232 CD06C0             CALL    CO
C235 C312C0             JMP     MNTR
               ;
               ;
               ;
               ;INTERPASS FUNCTIONS
               ;IF BIT 0 OF (PASS) IS EQUAL TO 1, THEN BIT 0 OF
               ;    (PASS) IS SET TO 0 AND 31H, ASCII 1, IS RETURNED IN
               ;    A-REG.  IF BIT 0 OF (PASS) IS EQUAL TO 0, THEN (PASS)
               ;    IS SET TO 00 AND 30H, ASCII 0, PLUS (PASS) SHIFTED
               ;    RIGHT 1 BIT POSITION IS RETURNED IN A-REG.  IF (PASS)
               ;    IS EQUAL TO 00, JMP UPDAT OCCURS.
```

```
C238 3A30C4   IPASS:   LDA     PASS
C23B 1F                RAR
C23C D24BC2            JNC     PASS2
C23F 3A30C4            LDA     PASS
C242 3D                DCR     A
C243 3230C4            STA     PASS
C246 3E01              MVI     A,1
C248 C35CC2            JMP     PASS3
C24B A7       PASS2:   ANA     A
C24C CA45C0            JZ      UPDAT
C24F CD7AC0            CALL    RESTR
C252 3A30C4            LDA     PASS
C255 1F                RAR
C256 F5                PUSH    PSW
C257 97                SUB     A
C258 3230C4            STA     PASS
C25B F1                POP     PSW
C25C C630     PASS3:   ADI     30H
C25E C9                RET


              ;PHYSICAL SECTOR TABLE. IS IN ORDER
              ;    OF LOGICAL SECTOR NUMBER.
              ;
C25F 01       TBL:     DB      1
C260 0A                DB      0AH
C261 13                DB      13H
C262 02                DB      2
C263 0B                DB      0BH
C264 14                DB      14H
C265 03                DB      3
C266 0C                DB      0CH
C267 15                DB      15H
C268 04                DB      4
C269 0D                DB      0DH
C26A 16                DB      16H
C26B 05                DB      5
C26C 0E                DB      0EH
C26D 17                DB      17H
C26E 06                DB      6
C26F 0F                DB      0FH
C270 18                DB      18H
C271 07                DB      7
C272 10                DB      10H
C273 19                DB      19H
C274 08                DB      8
C275 11                DB      11H
C276 1A                DB      1AH
C277 09                DB      9
C278 12                DB      12H
```

```
C279 00                  NOP
C27A 00                  NOP
C27B 00                  NOP
                ;
                ;
                ;
                ;
                ;        CONSOLE INPUT ROUTINE
                ;
C27C DB00       CIX:     IN       CCTRL
C27E E601                ANI      CRRDY
C280 C27CC2              JNZ      CIX
C283 DB01                IN       CDATA
C285 E67F                ANI      7FH
C287 C9                  RET
                ;
                ;
                ;
C288 0E0D       CRLF:    MVI      C,0DH
C28A CD06C0              CALL     CO
C28D 0E0A                MVI      C,0AH
C28F C306C0              JMP      CO
                ;
                ;        CONSOLE OUTPUT ROUTINE
                ;
C292 DB00       COX:     IN       CCTRL
C294 E680                ANI      CTRDY
C296 C292C2              JNZ      COX
C299 79                  MOV      A,C
C29A D301                OUT      CDATA
C29C C9                  RET
                ;
                ;
C29D 010BC3     INIT:    LXI      B,0C30BH
C2A0 2100C4              LXI      H,VCTRS
C2A3 11EAC3              LXI      D,VECTR
C2A6 70         INIT1:   MOV      M,B
C2A7 23                  INX      H
C2A8 1A                  LDAX     D
C2A9 77                  MOV      M,A
C2AA 23                  INX      H
C2AB 13                  INX      D
C2AC 1A                  LDAX     D
C2AD 77                  MOV      M,A
C2AE 23                  INX      H
C2AF 13                  INX      D
C2B0 0D                  DCR      C
C2B1 C2A6C2              JNZ      INIT1
C2B4 317FC4     MNTRX:   LXI      SP,STACK
C2B7 CD88C2              CALL     CRLF
```

```
C2BA 0E3E                MVI     C,3EH
C2BC CD06C0              CALL    CO
C2BF CDD9C2              CALL    CECHO
C2C2 FE54                CPI     'T'
C2C4 CA80C3              JZ      TSTM
C2C7 FE4D                CPI     'M'
C2C9 CA58C3              JZ      MEM
C2CC FE47                CPI     'G'
C2CE CAE1C2              JZ      GO
C2D1 0E3F       LER:     MVI     C,'?'
C2D3 CD06C0              CALL    CO
C2D6 C3B4C2              JMP     MNTRX
                ;
                ;
C2D9 CD03C0     CECHO:   CALL    CI
C2DC 4F                  MOV     C,A
C2DD CD06C0              CALL    CO
C2E0 C9                  RET
                ;
                ;
C2E1 CDE8C2     GO:      CALL    PARAM
C2E4 CD88C2              CALL    CRLF
C2E7 E9                  PCHL
                ;
                ;
C2E8 210000     PARAM:   LXI     H,0
C2EB CDD9C2     PARM1:   CALL    CECHO
C2EE FE0D                CPI     0DH
C2F0 C8                  RZ
C2F1 FE2C                CPI     ','
C2F3 C8                  RZ
C2F4 29                  DAD     H
C2F5 29                  DAD     H
C2F6 29                  DAD     H
C2F7 29                  DAD     H
C2F8 DAD1C2              JC      LER
C2FB CDEEC0              CALL    NBL
C2FE DAD1C2              JC      LER
C301 B5                  ORA     L
C302 6F                  MOV     L,A
C303 C3EBC2              JMP     PARM1
                ;
                ;
C306 CDD9C2     BYTEC:   CALL    CECHO
C309 CDEEC0     BYTC1:   CALL    NBL
C30C 07                  RLC
C30D 07                  RLC
C30E 07                  RLC
C30F 07                  RLC
C310 F5                  PUSH    PSW
```

```
C311 CDD9C2            CALL    CECHO
C314 CDEEC0            CALL    NBL
C317 C1                POP     B
C318 B0                ORA     B
C319 C9                RET
                  ;
                  ;
C31A F5           BYTEO:  PUSH    PSW
C31B CD2AC3            CALL    BYTO1
C31E 4F                MOV     C,A
C31F CD06C0            CALL    CO
C322 F1                POP     PSW
C323 CD2EC3            CALL    BYTO2
C326 4F                MOV     C,A
C327 C306C0            JMP     CO
                  ;
                  ;
C32A 0F           BYTO1:  RRC
C32B 0F                RRC
C32C 0F                RRC
C32D 0F                RRC
C32E E60F         BYTO2:  ANI     0FH
C330 FE0A              CPI     0AH
C332 FA37C3            JM      BYTO3
C335 C607              ADI     7
C337 C630         BYTO3:  ADI     30H
C339 C9                RET
                  ;
                  ;
C33A CD88C2       HLCO:   CALL    CRLF
C33D 7C                MOV     A,H
C33E CD1AC3            CALL    BYTEO
C341 7D                MOV     A,L
C342 CD1AC3            CALL    BYTEO
C345 C9                RET
                  ;
                  ;
C346 CD3AC3       DSPYM:  CALL    HLCO
C349 0E3D              MVI     C,'='
C34B CD06C0            CALL    CO
C34E 7E                MOV     A,M
C34F CD1AC3            CALL    BYTEO
C352 0E20              MVI     C,20H
C354 CD06C0            CALL    CO
C357 C9                RET
                  ;
                  ;
C358 CDE8C2       MEM:    CALL    PARAM
C35B CD46C3       MEM1:   CALL    DSPYM
C35E CDD9C2            CALL    CECHO
```

```
C361 FE0D              CPI      0DH
C363 CAB4C2            JZ       MNTRX
C366 FE20              CPI      20H
C368 CA6FC3            JZ       MEM9
C36B CD09C3            CALL     BYTC1
C36E 77                MOV      M,A

C36F 23       MEM9:    INX      H
C370 C35BC3            JMP      MEM1
              ;
              ;
C373 DB00     KBINT:   IN       CCTRL
C375 E601              ANI      CRRDY
C377 C0                RNZ
C378 DB01              IN       CDATA
C37A FE03              CPI      3
C37C CA12C0            JZ       MNTR
C37F C9                RET
              ;
              ;
C421          HIGH     EQU      SCTCH+21H

C380 CDE8C2   TSTM:    CALL     PARAM
C383 E5                PUSH     H
C384 EB                XCHG
C385 CDE8C2            CALL     PARAM
C388 2221C4            SHLD     HIGH
C38B EB                XCHG
C38C CD88C2            CALL     CRLF
C38F 3600     TSTM2:   MVI      M,0
C391 7B                MOV      A,E
C392 BD                CMP      L
C393 C29BC3            JNZ      TSTM3
C396 7A                MOV      A,D
C397 BC                CMP      H
C398 CA9FC3            JZ       TSTM4
C39B 23       TSTM3:   INX      H
C39C C38FC3            JMP      TSTM2

C39F 1E01     TSTM4:   MVI      E,1
C3A1 E1       TSTM7:   POP      H
C3A2 E5                PUSH     H
C3A3 34       TSTM1:   INR      M
C3A4 7B                MOV      A,E
C3A5 BE                CMP      M
C3A6 C4C2C3            CNZ      TSTM6
C3A9 3A21C4            LDA      HIGH
C3AC BD                CMP      L
C3AD C2BEC3            JNZ      TSTM5
C3B0 3A22C4            LDA      HIGH+1
```

```
C3B3 BC                 CMP    H
C3B4 C2BEC3             JNZ    TSTM5
C3B7 1C                 INR    E
C3B8 CD73C3             CALL   KBINT
C3BB C3A1C3             JMP    TSTM7

C3BE 23         TSTM5:  INX    H
C3BF C3A3C3             JMP    TSTM1

C3C2 CD46C3     TSTM6:  CALL   DSPYM
C3C5 7B                 MOV    A,E
C3C6 CD1AC3             CALL   BYTEO
C3C9 C388C2             JMP    CRLF
                ;
                ;
                ;
C3CC C303C0     RDIX:   JMP    CI
                ;
C3CF C306C0     LOX:    JMP    CO
                ;
C3D2 C306C0     POX:    JMP    CO
                ;
C3E4                    ORG    PROM+3E4H
                ;
                ;       I/O VECTOR TABLE
                ;
                ;***** MONITOR RE-ENTRY ADDRESS *********
C3E4 C3B4C2             JMP MNTRX
                ;
                ;
                ;****** MONITOR STARTING ADDRESS *********
C3E7 C39DC2             JMP    INIT    ;************
                VECTR:  DW     CIX     ;CONSOLE IN VECTOR
C3EC 92C2               DW     COX     ;CONSOLE OUT VECTOR
C3EE CCC3               DW     RDIX    ;PAPER TAPE READER VECTOR
C3F0 CFC3               DW     LOX     ;LINE PRINTER VECTOR
C3F2 D2C3               DW     POX     ;PUNCH VECTOR
C3F4 B4C2               DW     MNTRX   ;MONITOR VECTOR
C3F6 09C1               DW     RI      ;DISK READ VECTOR
C3F8 94C1               DW     WRT     ;DISK WRITE VECTOR
C3FA 4000               DW     40H     ;ASSEM/EDIT VECTOR;
C3FC 4000               DW     40H     ;EXECUTIVE VECTOR
C3FE 4300               DW     43H     ;UPDAT VECTOR
                ;
                        END
```

```
ASMB   C418      ASSEM  C03C      BASE   C430      BYTC1  C309
BYTE   C0D7      BYTEC  C306      BYTEO  C31A      BYTO1  C32A
BYTO2  C32E      BYTO3  C337      CCTRL  0000      CDATA  0001
CECHO  C2D9      CHK    C228      CHK1   C22F      CI     C003
CIX    C27C      CNTRL  00C0      CO     C006      COX    C292
CRLF   C288      CRRDY  0001      CTRDY  0080      DATAI  00C0
DATAO  00C1      DSPYM  C346      FDOS   C015      FDOS1  C05E
GO     C2E1      HIGH   C421      HLCO   C33A      ICNTR  C438
INCDA  C182      INCDB  C18D      INIT   C29D      INIT1  C2A6
IPASS  C238      ISCTR  C437      ISIZE  C434      ITRK   C436
IUNIT  C433      KBINT  C373      LER    C2D1      LO     C00C
LOOP   C205      LOOP1  C20A      LOOPV  C02D      LOX    C3CF
MEM    C358      MEM1   C35B      MEM9   C36F      MNTR   C012
MNTRX  C2B4      NBL    C0EE      NIO    C0FC      OCNTR  C43D
OFILE  C431      OSCTR  C43C      OSIZE  C439      OTRK   C43B
OUNIT  C432      PARAM  C2E8      PARM1  C2EB      PASS   C430
PASS2  C24B      PASS3  C25C      PASSV  C039      PO     C00F
POX    C3D2      PROG   C04E      PROM   C000      RDIX   C3CC
RDRIN  C009      REDO   C096      RED1   C0B6      RED2   C0C8
RED3   C0D5      REDX   C093      RESET  C054      RESTR  C07A
RFLAG  C203      RFLGV  C02A      RI     C109      RI10   C16E
RI2    C133      RI3    C136      RI4    C15B      RI5    C113
RI6    C149      RIV    C033      RIX    C100      RSTRV  C030
RSTV   C01E      SCTCH  C400      SEEK   C1F6      SEEKV  C027
STACK  C47F      START  C41B      TBL    C25F      TISZE  C43E
TITRK  C42F      TSTM   C380      TSTM1  C3A3      TSTM2  C38F
TSTM3  C39B      TSTM4  C39F      TSTM5  C3BE      TSTM6  C3C2
TSTM7  C3A1      UPDAT  C045      UPDTX  C41E      VCTRS  C400
VECTR  C3EA      WRT    C194      WRT1   C1A9      WRT2   C1B8
WRT3   C1DB      WRT4   C1DE      WRTN   C211      WRTN2  C216
WRTV   C036      XUS    C1E0      XUSV   C021      XXUS   C1EF
XXUSV  C024
```