



FDOS-III

OPERATOR'S MANUAL

505-345-9084 BoB Williams PRODUIT SUPPORT



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

1-1	Section		Page
SYSTEM ORGANIZATION	I PRELIM		 1-1
2-1 SOFTWARE MODULES 2-1 2-2 RESIDENT 2-1 2-3 EXECUTIVE 2-1 2-4 TEXT EDITOR 2-1 2-5 RELOCATING ASSEMBLER 2-2 2-6 LINKER 2-2 2-7 LIBRARY MANAGER 2-2 2-8 COPY 2-9 MEMORY-TO-DISK 2-2 2-10 SYSTEM I/O GENERATION 2-2 2-11 DISK LAYOUT 2-3 2-12 SYSTEM DISKETTE 2-3 2-13 USER DISKETTE 2-4 2-14 DISK FILES 2-4 2-15 LOCATION 2-4 2-16 FILE NAME 2-4 2-16 FILE NAME 2-4 2-18 DIRECTION LOCATION 2-5 2-19 CONTENTS DIRECTORY 2-5 2-20 SYSTEM DEVICE 2-6 2-21 SYSTEM DEVICE 2-6 2-21 SYSTEM DEVICE 2-6 2-21 SYSTEM DEVICE 2-6 2-6 3-4 OPERATION 3-1 3-1 3-2 COMMAND LINE 3-1 3-2 COMMAND LINE 3-1 3-3 ERROR MESSAGES 3-2 3-4 OPERATOR INTERRUPTION 3-3 3-3 3-3 STROR MESSAGES 3-2 3-4 OPERATOR INTERRUPTION 3-3 3-3 3-3 STROR MESSAGES 3-2 3-4 OPERATOR INTERRUPTION 3-3 3-3 3-3 3-3 STROR MESSAGES 3-2 3-4 OPERATOR INTERRUPTION 3-3 3-3 3-3 STROR MESSAGES 3-2 3-4 OPERATOR INTERRUPTION 3-3		LOADING FDOS-III	 1-1 1-2
2-1 SOFTWARE MODULES 2-1	II SYSTEM		 2-1
2-13 USER DISKETTE	2-11	SOFTWARE MODULES 2-2 RESIDENT. 2-3 EXECUTIVE 2-4 TEXT EDITOR 2-5 RELOCATING ASSEMBLER. 2-6 LINKER. 2-7 LIBRARY MANAGER 2-8 COPY. 2-9 MEMORY-TO-DISK. 2-10 SYSTEM I/O GENERATION	2-1 2-1 2-1 2-2 2-2 2-2 2-2 2-2 2-2
3-1 STARTING FDOS-III	2-14 2-17 2-18 2-19 2-20	2-13 USER DISKETTE DISK FILES 2-15 LOCATION. 2-16 FILE NAME DRIVE SPECIFIER. DIRECTION LOCATION CONTENTS DIRECTORY SYSTEM DEVICE. SYSTEM I/O DATA AREA	2-4 2-4 2-4 2-4 2-5 2-5 2-6 2-6
4-1 DISK INPUT/OUTPUT	3-1 3-2 3-3	STARTING FDOS-III	 3-1 3-1 3-2
4-2 DISK INPUT	IV RESIDE		0
	4-1	4-2 DISK INPUT	 1 - 1

TABLE OF CONTENTS

Secti	ion												Page
IV	CONTINUE	D											
	4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	-7 OP -8 OP -9 OP 10 RD 11 WT ISK HAND 13 UP 14 OP 15 OP 16 OP 17 RD	DAT (clenk (or enw (or enw (or enx (or enw (or	ose open expen expensed some expensed some expensed expen	outpuxist xist ew f ector a sec S.) xist ew f exist ew f esc ew f	ut fing ing ile r). ctor ing ile tor)	ile) file file for file for	for out;	in out) reconsting	put tpu adi iti	t). t). ng) ng)		4-3 4-4 4-4 4-5 4-5 4-6 4-6 4-7 4-7 4-7 4-8 4-9
٧	COMMANDS			H TAR	anda.	.0,1							5-1
	555555555555555555555555555555555555555	5-3 AS 5-4 BA 5-5 CH 5-6 CO 5-7 DE 5-8 DE 5-9 DU -10 EC -11 E) -12 HO -13 IN -14 LI -15 LO -16 ME -17 ME -19 PA -20 PI -21 RI -22 RI -23 RI	LOC MB TCH IGAT PY ELET & I LET JMP OHE OHE OHE ERGE ERGB	PACK								 	5-1 5-1 5-2 5-3 5-4 5-5 5-6 6-6 7-8 8-9 9-9 100 101 121 121

TABLE OF CONTENTS

Secti	on															Page
٧	CONTIN	IUED														
		5-25 5-26	VIEW XGEN		:	:			:	:		:	:			5-14 5-15
Appen	dix															
A B	COMMAN	D GLOSS	SARY .													A-1
C	DIAGNO	NITOR. STIC LI	ISTING												:	B-1 C-1
D	KF21DF	NT LIST	IING .													D-1

STATISTICS TO SUBAT

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 \emptyset , allow the drive to come up to speed and then GOTO the starting PROM address as detailed below.

Configu	ıration	PROM Resident	are needed.	Mini
System	iCOM Dash Number	Starting Address (in hex)	Comments	Mini- monitor (See appen dix B)
MDS	-53	Е8ØØН	Muss CPU cycles of Enter CI (consule	No
SBC 80/10 SBC 80/20	-56	Е8ØØН Е8ØØН	to the user U/D a	No No
Altair	-57	СØØØН СØØØН	2SIO Ports 1ØH,11H	Yes
Altair S-100 Bus	-58	СØØØН	Initialize I/O vectors, see 1-2	Yes
Poly	-59	СØØØН	With 4.0 monitor	Yes
So1	-6Ø	В8ØØН	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 AltairtmS-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 ØØ56H ØØ57H	ØC3H CI Address (Lo Byte) CI Address (Hi Byte)	JMP CO	Subroutine to return one character from console key board via the A-register carry bit reset.
ØØ58H ØØ59H ØØ5AH	ØC3H CO Address (Lo Byte) CO Address (Hi Byte)	JMP CO	Subroutine accepts a character from the C-register and outputs it to the console.
C4ØØH C4Ø1H C4Ø2H	ØC3H CI Address (Lo Byte) CI Address (Hi Byte)	JMP C1	Subroutine to return one character from console key board via the A-register, carry bit reset.
C4Ø3H C4Ø4H C4Ø5H	ØC3H CO Address (Lo Byte) CO Address (Hi Byte)	JMP CO	Subroutine accepts a character from the C-register and outputs it to the console.
Ø176H Ø177H	CO Address (Lo Byte) CO Address (Hi Byte)	CO	Vector CO restoration

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

2-13 USER DISKETTE

A User Diskette is divided into file directory and user file areas. Track Øis 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 \emptyset is actuated.

For example:

J0E3:1 #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 2 3 4 5	System I/O Data User Object Code User Object Code File Control Block (FCB) 1 thru 11 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 6	Name padded with spaces (code 20 hex) Attributes
7	Starting track address
8	Starting sector address
9-10 11	Length in sectors, most significant byte first (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 restrictionsØl - 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 \emptyset . 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 \emptyset .

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 \emptyset is assumed.

2-21 SYSTEM I/O DATA AREA

The system I/O Data is always assumed to be on track \emptyset , sectors 1 thru 3, of the drive \emptyset 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 ($\emptyset\emptyset$ H thru IFH), other than RETURN (\emptyset DH), 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 100H 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 ($\emptyset 3$ 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.

House one took service materials but originate from the beginning model and model and

DAME OF THE OWNER, OF THE OWNER, OF THE OWNER, OF THE OWNER, OWNER, OWNER, OWNER, OWNER, OWNER, OWNER, OWNER,

- most sill sacra encured wille leading as deposit file from
 - 1 unitally to result from distance.
 - 2 Attempt to write many information to a Trie than space
 - the ferror and the delay bold not ready

NO COMMITTEE STOTES AND

Operator tarefraction of langthy FDDS-III to octors may be accomplished by entering CTL-C (45 hea) from the input consols device. Upon termination, control is parent to the cossols for normal FDDS-III commend input.

Tempocary interruption of any function may be accomplished by entering CIL-E from the console. The functional process will recome upon the second CIL-C entry.

STOR

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

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 ITRK ISCTR	Input file Input file Input file	size in sectors (2 bytes)beginning track addressbeginning unit & sector address
ICNTR OSIZE OTRK OSCTR OCNTR	Controller Output file Output file Output file Controller	read buffer countersize in sectors (2 bytes)beginning track addressbeginning unit and sector addresswrite 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 \emptyset) the carry bit is returned as 1; otherwise, the carry bit is returned as a \emptyset .

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 ($\emptyset \emptyset$ -4C) from which input data is to be read.

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

The contents of ICNTR should be set to $\emptyset\emptyset$. 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

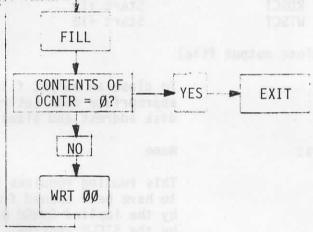
To open an output file, store the appropriate output file information into locations OSIZE, OTRK, OSCTR, and OCNTR. Each call to WRT will output, to disk, the byte contained in the same register that the microcomputer's debug or monitor program normally outputs a console data byte.

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

The contents of OCNTR should be set to $\emptyset\emptyset$. 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 (\emptyset 1-1A). 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 10 (\emptyset A 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-80/20	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

This routing requires the output file Comments:

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.

FILENAME - must be 5 character ASCII Parameters:

string stored in location FIELD with associated unit number as a hex value

stored in location DRIVE.

Comments: The A register indicates status

> successful Not Ø - failure

The location of field is start +7B HEX.

The location of drive is start +80 HEX.

OPENW (open existing file for output) 4-8

> Function: A file is opened for writing by the

WRT routine in the Resident Module.

See 4-7, PENR. Parameters:

See 4-7, OPENR. Comments:

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. \emptyset = successful - NOT \emptyset = 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.

WTSCT (write a sector) 4-11

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

 \emptyset = 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, allowi

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

 \emptyset = 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.

RDSCT (read=A=Sector) 4-17

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

 \emptyset = 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 \emptyset 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

TO/Fix 87-4

1,000,000

Mercan 128 bytes to specifical reptor of a

brumstein: 'L - register, track number (hea) - A - register, sector (mex) -

t - regretter. Detur frest t- t

and L register, but'er address

Comments: (n.this routine when acting the Older Comments Can Inches on Isbails are regulated, LEFLS and ERRITA

ENTER - one byte area that stores error code

100

ENTE must be first initialized to 8 before calling any disk

CHITM - two byte area that contains error condition routine is condition. This routine is given command when one of these serems about the contains a co

James - cellar public

Media error
Drive not ready
Dunitate filename

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 $\emptyset\emptyset$, 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 \emptyset .

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 \emptyset . 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 (\emptyset 2H) 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 Øl, thus setting it as a permanent, non-deletable, file.

CHGAT, MAIN, Ø(CR) CHGAT, MAIN(CR)

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

5-6 COPY

Format:

COPY(CR)

Purpose:

Copies contents of diskette in drive \emptyset 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 \emptyset , 1, 2, or 3.

If the unit number is omitted, \emptyset 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 IA) 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 \emptyset , 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 \emptyset .

HOME(CR) HOME,(CR) HOME,Ø(CR)

Returns the disk head, on drive unit \emptyset , to track \emptyset .

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 \emptyset .

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 \emptyset , 1, 2, or 3. If the unit

number is omitted, \emptyset 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 \emptyset .

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, beginning linenumber,

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 \emptyset .

RUN, MAIN, 6144Ø(CR)

Loads the contents of the object file MAIN into RAM memory with an offset bias of F000 hex.

RUN, MAIN, 100H

Loads the contents of the file MAIN into RAM with an offset of $100~{\rm HEX}.$

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 \emptyset .

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/O routines. When in SYSGN mode, data is accepted from the console and is stored on the systems area of the diskette in drive \emptyset (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)

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

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. $\emptyset\emptyset$ = BYTE NO. $\emptyset1$ = 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 20 by default. The first line displayed is line 0, if not specified otherwise.

Comments: Lines per frame and/or first line number may be omitted; and if so, are assumed to be 20 and 0 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 \emptyset).
- 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)

- Total street that that is "Airest the") .
- N Causes one segmenting frame to be displayed (. b.
- Cit (Command Bellium) Returns to E005-111 [recently of

MERK RG-R

NT) annumed TT, MROX : 40

Parquises

To gowernin the system region of a system diskette (See 2-12) in drive unit 0 from the copy of the caper FDUI-III Executive which is loaded rate the reader though device of filename is another.

et named

This crowned is used primarily to generate mes system districtes to new versions of the FDGS-III Encurive become available grayhon no system district exists.

All LO date is set to FFFFH or FFH (num-specified) and must be reset using the SYSEM command after using the WEW command.

off no system dishette exists, one can be generaled as follows:

- Load two copy of the FOOS-III Executive into PAN Removy and searchte it at memory location start.
 - I fine switch admit structly won a freend . S
 - 3. Place a capy of the FROS-III Executive into the reader input device and enter:

CHO DESIGNA

or place a distette with a copy of the FDS-III

(NO) I : DOME : NEW

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 \emptyset .

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 \emptyset onto the diskette in drive 1.

DELPK:u,filenamel,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, filenamel, 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 \emptyset .

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 l-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, beginning linenumber, list device 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

PRINT, Followard of Industry Frame, beginning fremmer of tylderige

prints the contents of the file on the list ortent

smarred fitween, some of the Lo. MARCH

renames the old file with the new filename

as I di sallon, emane (it, it.

loads the contents of the file into AAM for

Numero | Learning |

sets at the specified input, outset, and n parameters of given; loads the contents of the new object file. left the text object file. left the text of the does a branch to legalism.

System permits alteration of initialization data.

VIII, Tilerawe, itmesperioage, firstifne, listdevice

oft plays the contents of the specified file one

WISHEST PT_PTSEX

enables system generation of other 1500 FDOS versions as might become available in the fature.

parioni v

APPENDIX B

MINIMONITOR

B-1 MINIMONITOR

NOTE

The Minimonitor is only available for microcomputers employing an Altairtm 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

O KITHEFFEA

DELL'AND STREET DESTRUCTIONS

```
; 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 O(OR NOTHING), 1, 2, OR 3
; T=TRACK
; S=SECTOR
       -CLEAR DRIVE ELECTRONICS
jΑ
      -SEEK TO TRACK
; BU, T
; 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
       -TRKO TO TRK76 LOOP
       -UNIT SELECT TEST
; I
       -SEEK TEST ONCE (2 MIN)
لال ;
; 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

X 701.75

CHRISTIAN COLOR CO

Of the Syries and Shell Hitters of the Arthur

distribution for the first particle of the payment

derivous communication and an extension of the communication of the comm

The state of the s

AND DESCRIPTION OF THE PARTY OF

STATE - HOME A STATISH

```
OCOOOH
0000
               PROM
                        EQU
               ; ****
               RESIDENT 8080 ALTAIR/IMSAI/POLY 88 FDOS III VERSION 1.0
               ; ***
               ; ENTRY ADDRESSES-
                        POWER UP = C3E7 HEX
                        RE-ENTRY = C3E4 HEX
0001
               DATAO
                        EQU
                                 0C1H
0000
               DATAI
                        EQU
                                 OCOH
                                 OCOH
0000
               CNTRL
                        EQU
0000
               CCTRL
                        EQU
                                          CONSOLE CONTROL PORT
                        EQU
                                          CONSOLE DATA PORT
0001
               CDATA
                                 1
                                          CONSOLE DATA READY
0001
               CRRDY
                        EQU
                                 1
                                          ; CONSOLE XMIT READY
0080
               CTRDY
                        EQU
                                 80H
C400
               SCTCH
                        EQU
                                 0C400H
                                          SCRATCH RAM
C400
               VCTRS
                        EQU
                                 SCTCH
                                          ; I/O VECTORS
                        EQU
                                 SCTCH+30H
C430
               BASE
C47F
               STACK
                        EQU
                                 SCTCH+7FH
               į
                        EQU
                                 BASE
C430
               PASS
                                 BASE+1
C431
               OFILE
                        EQU
               OUNIT
                        EQU
                                 BASE+2
C432
0433
               IUNIT
                        EQU
                                 BASE+3
C434
               ISIZE
                        EQU
                                 BASE+4
C436
               ITRK
                        EQU
                                 BASE+6
               ISCTR
                        EQU
C437
                                 BASE+7
C438
               ICNTR
                        EQU
                                 BASE+8
0439
               OSIZE
                        EQU
                                 BASE+9
C43B
               OTRK
                        EQU
                                 BASE+11
C43C
               OSCIR
                        EQU
                                 BASE+12
C43D
               OCNTR
                        EQU
                                 BASE+13
C42F
               TITRK
                        EQU
                                 BASE-1
C43E
               TISZE
                        EQU
                                 BASE+14
C418
               ASMB
                        EQU
                                 VCTRS+24
                                 VCTRS+27
C41B
               START
                        EQU
C41E
               UPDIX
                        EQU
                                 VCTRS+30
```

```
ORG
                               PROM
0000
              ENTRY POINT WHEN Q IS TYPED
              ; LOADS FDOS AND BRANCHES TO FDOS S. A.
C000 C315C0
                       JMP
                               FDOS
                       JMP
                               VCTRS ; KEYBOARD INPUT VECTOR
C003 C300C4
              CI:
                               VCTRS+3 ; CONSOLE OUTPUT VECTOR
C006 C303C4
              CO:
                       JMP.
C009 C306C4
                               VCTRS+6 ; READER INPUT VECTOR
              RDRIN:
                       JMP
                               VCTRS+9 ; LIST OUTPUT VECTOR
COOC C309C4
              LO:
                       JMP
                                             ; PUNCH OUTPUT VECTOR
COOF C30CC4
                               VCTRS+12
              PO:
                       JMP
                                             ; SYSTEM MONITOR VECTOR
                               VCTRS+15
CO12 C30FC4 ---
              MNTR:
                       JMP
                               SP, STACK
CO15 317FC4
              FDOS:
                       LXI
                       CALL
                               FDOS1
CO18 CD5ECO
                       JMP
CO1B C31BC4
                               START
                       JMP
CO1E C354CO
              RSTV:
                               RESET
C021 C3E0C1
              XUSV:
                       JMP
                               XUS
CO24 C3EFC1
              XXUSV:
                       JMP
                               XXUS
C027 C3F7C1
              SEEKV:
                       JMP
                               SEEK+1
C02A C303C2
              RELGY:
                       JMP
                               RFLAG
C02D C305C2
              LOOPY:
                       JMP
                               LOOP
C030 C37AC0
              RSTRV:
                       JMP
                               RESTR
C033 C309C1
              RIV:
                       JMP
                               RI
C036 C394C1
              WRTV:
                       . IMP
                               WRT
C039 C338C2
              PASSV:
                       JMP.
                               IPASS ; ASMB INTERPASS FNC
                       CALL
C03C CD93C0
              ASSEM:
                               REDX
COSF CD7ACO
                       CALL
                               RESTR
C042 C318C4
                       JMP
                               ASMB
```

```
CO45 317FC4
                              SP, STACK
              UPDAT:
                      LXI
CO48 CD5ECO
                      CALL
                              FDOS1
CO4B 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, ODH
C05B C305C2
                      JMP
                             LOOP
CO5E CD54CO
              FDOS1:
                      CALL
                             RESET
                             H, O ; SET BIAS=O
C061 210000
                      LXI
                             H
CO64 E5
                      PUSH
                             H, 105
CO65 216900
                      LXI
C068 2234C4
                             ISIZE
                      SHLD
CO6B 2136C4
                     LXI
                             H, ITRK
                             M, 1 ; TRACK=1
CO6E 3601
                     MVI
C070 2C
                      INR
C071 3600
                             M, O ; SECTOR=O
                     MVI
0073 20
                             L ; READ BFR EMPTY
                      INR
CO74 3600
                     MVI
                             M, O
C076 CD93CO
                     CALL
                             REDX
CO79 C9
                     RET
                                   ; GO TO FDOS
CO7A 2A3EC4
             RESTR:
                     LHLD
                             TISZE ; RESTORE IFILE POINTERS
CO7D 2234C4
                             ISIZE
                     SHLD
C080 3A2FC4
                     LDA
                             TITRK
C083 3236C4
                             ITRK
                     STA
C086 3A33C4
                             IUNIT
                     LDA
C089 OF
                     RRC
COSA OF
                     RRC
CO8B 3237C4
                     STA
                             ISCTR
COSE 97
                             A
                     SUB
COSF 3238C4
                     STA
                             ICNTR
CO92 C9
                     RET
             SUBROUTINE TO READ A HEX FILE INTO MEMORY
             ; STARTS WITH ROUTINE REDO, USES ALL REGISTERS
```

	j				
	i				
C093 E1 C094 E3	REDX:	POP XTHL	н	; SWAP BIA	8 & RETURN
C095 E5		PUSH	Н		
C096 E1	REDO:	POP	Н	GET BIAS	
C097 E5	TALLAND.	PUSH	Н		
C098 CD00C1		CALL	RIX	GET CHAR	INTO A
CO9B 063A		MVI	B, 7: 7	VI	
CO9D 90		SUB	В		
C09E C296C0		JNZ	REDO		
COA1 57		MOV	D. A		
COA2 CDD7CO		CALL	BYTE		
COA5 CAC8CO		JZ	RED2		
COA8 5F		MOV	E, A		
COA9 CDD7CO		CALL	BYTE		
COAC F5		PUSH	PSW		
COAD CDD7CO		CALL	BYTE		
COBO C1		POP	В		
COB1 4F		MOV	C, A		
COB2 09		DAD	В		
COB3 CDD7CO		CALL	BYTE		
COB6 CDD7CO	RED1:	CALL	BYTE		
COB9 77		MOV	M, A		
COBA 23		INX	H		
COBB 1D		DCR	E		
COBC C2B6CO		JNZ	RED1		
COBF CDD7CO		CALL	BYTE		
COC2 C2D1C2		JNZ JMP	LER REDO		
COC5 C396CO COC8 CDD7CO	RED2:	CALL	BYTE		
COC8 CDD7CO COCB 67	REDZ.	MOV	H, A		
COCC CDD7CO		CALL	BYTE		
COCF 6F		MOV	LA		
CODO B4		ORA	H		
COD1 CAD5CO		JZ	RED3		
COD4 E9		PCHL			
CODS E1	RED3:	POP	Н		
COD6 C9		RET			
	i				
	3				
	i				
COD7 CDOOC1	BYTE:	CALL			
CODA CDEECO		CALL			
CODD 07		RLC			
CODE 07		RLC			
CODF 07		RLC	,		
COEO 07		RLC	C A		
COE1 4F		MOV	CA		
COE2 CDOOC1		CALL	RIX		

```
COE5 CDEECO
                       CALL
                               NBL
COES B1
                       ORA
                               C
COE9 4F
                               CLA
                       MOV
COEA 82
                               D
                       ADD
COEB 57
                       MOV
                               D, A
COEC 79
                       MOV
                               A. C
COED C9
                       RET
               ; SUBROUTINE TO CONVERT TWO HEX CHARACTERS
                TO ONE BYTE
COEE D630
                       SUI
                                404
               NBL:
COFO D8
                       RC
COF1 C6E9
                       ADI
                               OE9H
COF3 D8
                       RC
COF4 C606
                       ADI
                               6
COF6 F2FCCO
                       JP.
                               NIO
COF9 C607
                       ADI
COFB D8
                       RC
COFC C60A
               NIO:
                       ADI
                               10
COFE B7
                       ORA
COFF 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
                       CALL
              RIX:
                               RI
C103 DA12C0
                       JC
                               MNTR
C106 E67F
                       ANI
                               7FH
C108 C9
                       RET
C109 C5
              RI:
                      PUSH
                               В
                                       SAVE REG D-L
C10A E5
                       PUSH
                               H, ICNTR
C10B 2138C4
                      LXI
C10E 7E
                      MOV
                               A, M
CIOF A7
                      ANA
                               A
                                       3 CNT=0?
                               RI10 ; NO
C110 C26EC1
                      JNZ
C113 2E37
              RI5:
                      MVI
                               L, ISCTR AND OFFH
                                                       ; YES-INCR D. A.
```

```
CALL INCDA
C115 CD82C1
                          ISIZE
                    LHLD
C118 2A34C4
                          H
C11B 2B
                    DCX
                           ISIZE
C11C 2234C4
                    SHLD
                    MOV
                            A, L
C11F 7D
                            A
C120 A7
                    ANA
                           RI3
C121 C236C1
                    JNZ
0124 70
                    MOV
                           A, H
C125 A7
                    ANA
                            A
                           RI3
C126 C236C1
                    JNZ
                           H
ISIZE
C129 23
                    INX
                           H, ICNTR
                    SHLD
C12A 2234C4
C12D 2138C4
                    LXI
                           M, O
                    MVI
C130 3600
                                  SET EOF
                    STC
C132 37
                                  ; RESTORE D-L
C133 E1
             RI2:
                    POP
                            B
                    POP
C134 C1
C135 C9
                    RET
                           H, ISCTR ; XMIT UNIT/SECTOR
            RI3:
                    LXI
C136 2137C4
C139 CDEOC1
                    CALL
                            XUS
                                 ; MAKE SURE A DISK
C13C CD28C2
                            CHK
                    CALL
                                  ;SET CNTR=128
                            __
C13F 2C
                    INR
C140 3680
                            M, 128
                    MVI
                                  SET TRY CNT=5
                    MVI
                            C, 5
C142 OE05
                            L, ITRK AND OFFH ; SEEK TRACK
C144 2E36
                    MVI
                           SEEK
C146 CDF6C1
                    CALL
                                   READ DATA
                           A, 3
C149 3E03
             RI6:
                    MVI
                           LOOP
C14B CD05C2
                    CALL
                            DATAI ; DD MARK?
C14E DBCO
                    IN
                    ANI
                            80H
C150 E680
                                   ; NO
                            RI4
C152 CA5BC1
                    JZ.
C155 CD03C2
                            RFLAG
                    CALL
                            RI5
C158 C313C1
                    JMP
                                   ; CRC ERROR?
C15B DBCO
             RI4:
                    IN
                            DATAI
                    ANI
                            8H
C15D E608
                            RI10 ; NO
                    JZ
C15F CA6EC1
                            RFLAG
C162 CD03C2
                    CALL
                            C
                                   ; DECR CNTR
C165 OD
                    DOR
C166 C249C1
                    JNZ
                            RI6
                    MVI
                            A, 1
C169 3E01
                            CHK1
C16B C32FC2
                    JMP
                    IVM
                                   READ BYTE INTO A
                            A, 40H
C16E 3E40
             RI10:
                    OUT
                            CNTRL
C170 D3C0
C172 DBCO
                    IN
                            DATAI
                    MOV
                            CA
C174 4F
                                   STROBE BUFFER
                    MVI
                            A, 41H
C175 3E41
C177 CD05C2
                    CALL
                            LOOP
                            L, ICNTR AND OFFH ; DECR
                                                         READ COUNTER
                    MVI
C17A 2E38
```

```
DCR M
MOV A,C
 C17C 35
 C17D 79
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 CASDC1 JZ INCDB
C18B 2D DCR L
C18C C9 STV8 AUTURENTI RETURNING TO UT SWILLOWING
C18D 7E INCDB: MOV A, M
C18E E6C1 ANI OC1H
C190 77 MOV M, A
C191 2D DCR L
C192 34 INR M
C193 C9 RET
          BOT PHYS: SECTOR
                 SUBROUTINE TO WRITE A BYTE TO DISK
                 EXPECTS CHAR TO BE IN C-REG
 STEEL SYMP & TARK BORES
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
C1AO 34 INR M ; INCREMENT BFR CNT
C1A1 7E MOV A, M
C1A2 FE8O CPI 128 ;=128?
C1A4 C2DFC1 UN7 WBT4 NO
C1A4 C2DEC1
                          JNZ WRT4 ; NO
MVI M,O ; CLEAR COUNT
C1A7 3600
                           LXI H, OSCTR ; XMIT UNIT/SECTOR
C1A9 213CC4
                 WRT1:
C1AC CDEOC1
                           CALL XUS
                           CALL CHK ; MAKE SURE A DISK
MVI C.5 ; SET TRY CNT=5
DCR L ; SEEK TRACK
CALL SEEK
C1AF CD28C2
C1B2 0E05
C1B4 2D
C1B5 CDF6C1
C1B8 3E05 WRT2:
C1BA CD05C2
                           CALL
                           MVI A,5 ; WRITE DATA CALL LOOP
                 MVI A,7 ; READ FOR CRC
CALL LOOP
C1BD 3E07
C1BF CD05C2
                                     DATAI ; CRC ERROR?
C1C2 DBCO
                           IN
```

```
ANI 8H
JZ WRT3 ; NO
CALL RFLAG
DCR C ; DECR TRY CNT
JNZ WRT2 ; TRY AGAIN
MVI A, OFH ; WRITE AS DD
C1C4 E608
C1C6 CADBC1
C1C9 CD03C2
CICC OD
C1CD C2B8C1
C1DO 3EOF
C1D2 CD05C2
                         LOOP
                   CALL
               CALL WRTN ; INCREMENT DA & CHK SIZE
C1D5 CD11C2
                   JMP WRT1
CALL WRTN ; INCREMENT DA & CHK SIZE
POP H ; RESTORE D-L
C1D8 C3A9C1
            WRT3: CALL
WRT4: POP
C1DB CD11C2
CIDE E1
C1DF C9
                   RET
            ; SUBROUTINE TO TRANSMIT UNIT/SECTOR BYTE
                                NOTE BY SHEET
                   MOV A, M
C1E0 7E
            XUS:
                        1FH ; EXTRACT LOG SECTOR
C1E1 E61F
                   ANI
C1E3 E5
                   PUSH H
LXI H, TBL
C1E4 215EC2
                          H, TBL-1 ; GET TABLE PNTR
                          L ; MAKE SECTOR PNTR
C1E7 85
                   ADD
C1E8 6F
                   MOV
                          L, A
C1E9 4E
                   MÜV
                         C, M ; GET PHYS SECTOR
C1EA E1
                   POP
                          H
         MOV A, M
C1EB 7E
            ANI OCOH
C1EC E6CO
                   ORA C ; MERGE UNIT & PHYS SCTR
OUT DATAO
MVI A, 21H
C1EE B1
C1EF D3C1
            XXUS:
C1F1 3E21
            JMP LOOP
C1F3 C305C2
     SUBROUTINE TO SEEK TRACK IN A
           - 1
           SEEK: MOV
                         A, M
C1F6 7E
                  OUT DATAO
MVI A,11H
C1F7 D3C1
C1F9 3E11 MVI
C1FB CD05C2 CALL LOOP
                 MVI A,09
C1FE 3E09
C200 C305C2 JMP
                         LOOP
         E-PH; NEF TEE
            SUBROUTINE TO RESET FLAG
C203 3E0B RFLAG: MVI A.OBH
            ; SUBROUTINE TO ISSUE CMD & LOOP ON BUSY
            HORRIS THE LATER NO.
```

```
C205 D3C0
              LOOP:
                     OUT
                             CNTRL
C207 97
                      SUB
                             Α
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
                     CALL
C213 CD82C1
                             INCDA
                     LHLD
C216 2A39C4
              WRTN2:
                             OSIZE
C219 2B
                     DCX
C21A 2239C4
                     SHLD
                             OSIZE
C21D 7D
                     MOV
                             A. L
C21E A7
                     ANA
                             A
C21F C0
                     RNZ
0220 70
                     MOV
                             A, H
                             A
C221 A7
                ANA
C222 CO
                   RNZ
                    MVI
C223 3E02
                     JMP
C225 C32FC2
                             CHK1
              SUBROUTINE TO CHECK IF A DISK, ELSE ERROS
C228 DBCO
             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
                             CA
C232 CD06C0
                     CALL
                             CO
C235 C312C0
                     JMP
                             MNTR
              ; INTERPASS FUNCTIONS
             ; IF BIT O OF (PASS) IS EQUAL TO 1, THEN BIT O OF
                 (PASS) IS SET TO 0 AND 31H, ASCII 1, IS RETURNED IN
                 A-REG. IF BIT O OF (PASS) IS EQUAL TO O, THEN (PASS)
                 IS SET TO OO AND SOH, ASCII O, PLUS (PASS) SHIFTED
                 RIGHT 1 BIT POSITION IS RETURNED IN A-REG. IF (PASS)
                IS EQUAL TO OO, JMP UPDAT OCCURS.
```

C238	3A30C4	IPASS:	LDA	PASS
C23B	1F		RAR	
C23C	D24BC2		JNC	PASS2
C23F	3A30C4		LDA	PASS
C242	3D		DOR	A
0243	323004		STA	PASS
0246	3E01		MVI	A, 1
C248	035002		JMP	PASS3
C24B	A7	PASS2:	ANA	A
0240	CA45C0		JZ	UPDAT
C24F	CD7ACO		CALL	RESTR
C252	3A30C4		LDA	PASS
C255	1F		RAR	
C256	F5		PUSH -	PSW
C257	97		SUB	A
C258	323004		STA	PASS
C25B	F 1		POP	PSW
C25C	0630	PASSS:	ADI	30H
C25E	C9		RET	

; PHYSICAL SECTOR TABLE. IS IN ORDER ; OF LOGICAL SECTOR NUMBER.

		i			
C25F	01	TBL:	DB	1	
0260	OA		DB	OAH	
0261	13		DB	13H	
C262	02		DB	2	
0263	OB I		DB	OBH	
C264	14		DB	1 4H	
C265	03		DB	3 - 4	
C266	OC		DB	OCH	
0267	15		DB	15H	
C268	04		DB	4	
0269	OD		DB	ODH	
C26A	16		DE	16H	
C26B	05		DB	5	
0260	OE		DB	OEH	
C26D	17		DB	17H	
C26E	06		DB	6	
C26F	OF		DB	OFH	
C270	18		DE	18H	
C271	07		DB	7	
C272	10		DE	10H	
C273	19		DE:	19H	
C274	08		DB	8	
C275	11.		DB	11H	
C276	1A		DB	21111	
C277	09			9.00	
0278	12		DB	12H	

C279 C27A C27B	00 00 00	; ; ;	NOP NOP NOP	INPUT R	OUTINE		
C27C C27E C280 C283 C285 C287	DB00 E601 C27CC2 DB01 E67F C9	CIX:	IN ANI JNZ IN ANI RET	CCTRL CRRDY CIX CDATA 7FH		INANI	18H 11H C27C 11H
C288 C28A C28D C28F	OEOD CDO6CO OEOA C3O6CO	CRLF:	MVI CALL MVI JMP CONSOLE	C, ODH CO C, OAH CO	ROUTINE		
C292 C294 C296 C299 C29A C29C	DB00 E680 C292C2 79 D301 C9	COX:	IN ANI JNZ MOV OUT RET	CCTRL CTRDY COX A, C CDATA		DMI DMI 52 097	10H 02H C292 11H
C2AE C2AF C2B0 C2B1 C2B4	23 1A 77 23 13 1A 77 23 13	INIT: INIT1:	LXI LXI MOV INX LDAX MOV INX LDAX MOV INX INX INX INX INX INX INX	B, OC3OBI H, VCTRS D, VECTR M, B H D M, A H D M, A H D C INIT1 SP, STAC CRLF			

C2BA OE3E C2BC CD06C0 C2BF CDD9C2 C2C2 FE54 C2C4 CA80C3 C2C7 FE4D C2C9 CA58C3 C2CC FE47 C2CE CAE1C2 C2D1 OE3F C2D3 CD06C0 C2D6 C3B4C2	LER:	MVI CALL CPI JZ CPI JZ CPI JZ MVI CALL JMP	C,3EH CO CECHO /T/ TSTM /M/ MEM /G/ GO C,/?/ CO MNTRX
C2D9 CD03C0 C2DC 4F C2DD CD06C0 C2E0 C9	CECHO:	CALL MOV CALL RET	CI C, A CO
C2E1 CDE8C2 C2E4 CD88C2 C2E7 E9	GO:	CALL CALL PCHL	PARAM CRLF
C2E8 210000 C2EB CDD9C2 C2EE FE0D C2F0 C8 C2F1 FE2C C2F3 C8 C2F4 29 C2F5 29 C2F6 29 C2F7 29 C2F8 DAD1C2 C2FB CDEECO C2FE DAD1C2 C301 B5 C302 6F C303 C3EBC2	PARAM: PARM1:	LXI CALL CPI RZ CPI RZ DAD DAD DAD DAD DAD DAD DAD DAD JC CALL JC ORA MOV JMP	H, O CECHO ODH /, / H H H LER NBL LER L, A PARM1
C306 CDD9C2 C309 CDEEC0 C30C 07 C30D 07 C30E 07 C30F 07 C310 F5	; BYTEC: BYTC1:	CALL CALL RLC RLC RLC RLC PUSH	CECHO NBL PSW

C311 CDD9C2 C314 CDEECO C317 C1 C318 BO C319 C9	j	CALL CALL POP ORA RET	CECHO NBL B B
C31A F5 C31B CD2AC3 C31E 4F C31F CD06C0 C322 F1 C323 CD2EC3 C326 4F C327 C306C0	; BYTEO:	PUSH CALL MOV CALL POP CALL MOV JMP	PSW BYTO1 C, A CO PSW BYTO2 C, A
C32A OF C32B OF C32C OF C32D OF C32E E6OF C33O FEOA C332 FA37C3 C335 C6O7 C337 C630	BYTO1: BYTO2: BYTO3:	RRC RRC RRC ANI CPI JM ADI ADI RET	OFH OAH BYTO3 7 3OH
C33A CD88C2 C33D 7C C33E CD1AC3 C341 7D C342 CD1AC3 C345 C9	; HLCO:	CALL MOV CALL MOV CALL RET	CRLF A, H BYTEO A, L BYTEO
C346 CD3AC3 C349 OE3D C34B CD06C0 C34E 7E C34F CD1AC3 C352 OE20 C354 CD06C0 C357 C9	DSPYM:	CALL MVI CALL MOV CALL MVI CALL RET	HLCO C, '=' CO A, M BYTEO C, 20H CO
C358 CDE8C2 C35B CD46C3 C35E CDD9C2	; MEM: MEM1:	CALL CALL CALL	PARAM DSPYM CECHO

C361 C363 C366 C368 C36B C36E	FEOD CAB4C2 FE2O CA6FC3 CD09C3 77		CPI JZ CPI JZ CALL MOV	ODH MNTRX 20H MEM9 BYTC1 M, A
C36F C370		MEM9:	INX JMP	H MEM1
C375 C377 C378 C37A	DB00 E601 C0 DB01 FE03 CA12C0 C9	; KBINT:	IN ANI RNZ IN CPI JZ RET	CCTRL CRRDY CDATA 3 MNTR
C421		; HIGH	EQU	SCTCH+21H
C380 C384 C385 C388 C386 C387 C391 C392 C393 C396 C397 C398 C398	CDE8C2 E5 EB CDE8C2 2221C4 EB CD88C2 3600 7B BD C29BC3 7A BC CA9FC3 23 C38FC3	TSTM2:	CALL PUSH XCHG CALL SHLD XCHG CALL MVI MOV CMP JNZ MOV CMP JZ INX JMP	PARAM H PARAM HIGH CRLF M, O A, E L TSTM3 A, D H TSTM4 H TSTM4
C39F C3A1 C3A2 C3A3 C3A4 C3A5 C3A6 C3A9 C3AC C3AD C3BO	1E01 E1 E5 34 7B BE C4C2C3 3A21C4 BD C2BEC3 3A22C4	TSTM4: TSTM7: TSTM1:	MVI POP PUSH INR MOV CMP CNZ LDA CMP JNZ LDA	E, 1 H H M A, E M TSTM6 HIGH L TSTM5 HIGH+1

	C3B7 C3B8	C2BEC3		CMP JNZ INR CALL JMP	H TSTM5 E KBINT TSTM7	
		23 C3A3C3	TSTM5:	INX JMP	H TSTM1	CO MARK CATE ASSERT CO
	C3C5 C3C6	CD46C3 7B CD1AC3 C388C2		CALL MOV CALL JMP	DSPYM A,E BYTEO CRLF	
)))			THEORE CORE THEOR CT
1	C3CC	C303C0	RDIX:	JMP	CI	50 1900 - 5000 900 I
	C3CF	C306C0	Lox:	JMP	co	NION CADIFA
	C3D2	C306C0	POX:	JMP	СО	NOW LOADER
	СЗЕ4		J. 000	ORG	PROM+3E	
			j je	I/O VECT	TOR TABL	E
1			; : *****	MONITOR F	RE-ENTRY	ADDRESS ******
	C3E4	C3B4C2	;	JMP MNTE		es estua de au vici
			1			
	C3E7	C39DC2	; *****	MONITOR JMP		IG ADDRESS ******** ; *********
	C3EC	9202	VECTR:	DW DW	COX	CONSOLE IN VECTOR CONSOLE OUT VECTOR
		CCC3		DW	RDIX	; PAPER TAPE READER VECTOR
		CFC3 D2C3		DW DW	LOX POX	; LINE PRINTER VECTOR ; PUNCH VECTOR
		B4C2		DW	MNTRX	MONITOR VECTOR
		0901		DW	RI	DISK READ VECTOR
	C3F8			DW	WRT	DISK WRITE VECTOR
		4000		DW	40H	;ASSEM/EDIT VECTOR;
	CSFC	4000		DW	40H	EXECUTIVE VECTOR
	C3FE	4300		DW	43H	; UPDAT VECTOR
			i	END		

C418	ASSEM	C03C	BASE	C430	BYTC1	0309
COD7	BYTEC	C306	BYTEO	C31A	BYT01	C32A
C32E	BYT03	C337	CCTRL	0000	CDATA	0001
C2D9	CHK	C228	CHK1	C22F	CI	C003
C27C	CNTRL	0000	CO	C006	COX	C292
C288	CRRDY	0001	CTRDY	0080	DATAI	0000
00C1	DSPYM	C346	FDOS	C015	FDOS1	CO5E
C2E1	HIGH	0421	HLCO	C33A	ICNTR	C438
C182	INCDB	C18D	INIT	C29D	INIT1	C2A6
C238	ISCTR	C437	ISIZE	C434	ITRK	C436
C433	KBINT	C373	LER	C2D1	LO	COOC
C205	LOOP1	C20A	LOOPY	CO2D	LOX	CBCF
C358	MEM1	C35B	MEM9	C36F	MNTR	0012
C2B4	NBL	COEE	NIO	COFC	OCNTR	C43D
C431	OSCIR	C43C	OSIZE	C439	OTRK	C43B
C432	PARAM	C2E8	PARM1	C2EB	PASS	0430
C24B	PASS3	C25C	PASSV	0039	PO	COOF
C3D2	PROG	CO4E	PROM	C000	RDIX	C3CC
C009	REDO	C096	RED1	COB6	RED2	0008
COD5	REDX	C093	RESET	C054	RESTR	CO7A
C203	RFLGV	CO2A	RI	C109	RI10	C16E
C133	RIB	C136	RI4	C15B	RI5	C113
C149		C033	RIX	C100	RSTRV	C030
CO1E		C400	SEEK	C1F6	SEEKV	C027
C47F		C41B	TBL	C25F	TISZE	C43E
C42F	TSTM	C380	TSTM1	C3A3	TSTM2	C38F
C39B	TSTM4	C39F	TSTM5	C3BE	TSTM6	C3C2
C3A1	UPDAT	C045	UPDTX	C41E	VCTRS	C400
CBEA	WRT	C194	WRT1	C1A9	WRT2	C1B8
C1DB	WRT4	C1DE	WRTN	C211	WRTN2	C216
C036	XUS	C1EO	XUSV	CO21	XXUS	C1EF
C024	1					
THE HELBI	29 3 (202)					
	COD7 C32E C2D9 C27C C288 OOC1 C2E1 C182 C238 C433 C205 C358 C2B4 C431 C432 C24B C3D2 C009 C0D5 C203 C133 C149 C01E C47F C42F C3PB C3PA C3PA C3PA C3PA C3PA C3PA C3PA C3PA	COD7 BYTEC C32E BYTO3 C2D9 CHK C27C CNTRL C288 CRRDY OOC1 DSPYM C2E1 HIGH C182 INCDB C238 ISCTR C433 KBINT C205 LOOP1 C358 MEM1 C2B4 NBL C431 OSCTR C432 PARAM C24B PASS3 C3D2 PROG C009 REDO C0D5 REDX C203 RFLGV C133 RI3 C149 RIV C01E SCTCH C47F START C42F TSTM C39B TSTM4 C3A1 UPDAT C3EA WRT C1DB WRT4 C036 XUS	COD7 BYTEC C306 C32E BYTO3 C337 C2D9 CHK C228 C27C CNTRL 00C0 C288 CRRDY 0001 00C1 DSPYM C346 C2E1 HIGH C421 C182 INCDB C18D C238 ISCTR C437 C433 KBINT C373 C205 L00P1 C20A C358 MEM1 C35B C2B4 NBL C0EE C431 OSCTR C43C C432 PARAM C2E8 C24B PASS3 C25C C3D2 PROG C04E C009 RED0 C096 C0D5 REDX C093 C203 RFLGV C02A C133 RI3 C136 C149 RIV C033 C01E SCTCH C400 C47F START C41B C42F TSTM C380 C39B TSTM4 C39F C3A1 UPDAT C045 C3EA WRT C194 C1DB WRT4 C1DE C036 XUS C1E0	COD7 BYTEC C306 BYTEO C32E BYTO3 C337 CCTRL C2D9 CHK C228 CHK1 C27C CNTRL 00C0 C0 C288 CRRDY 0001 CTRDY 00C1 DSPYM C346 FDOS C2E1 HIGH C421 HLCO C182 INCDB C18D INIT C238 ISCTR C437 ISIZE C433 KBINT C373 LER C205 L00P1 C20A L00PV C358 MEM1 C35B MEM9 C2B4 NBL C0EE NIO C431 OSCTR C43C OSIZE C432 PARAM C2E8 PARM1 C24B PASS3 C25C PASSV C3D2 PROG C04E PROM C009 REDO C096 RED1 C005 REDX C093 RESET C203 RFLGV C02A RI C133 RI3 C136 RI4 C149 RIV C033 RIX C01E SCTCH C400 SEEK C47F START C41B TBL C42F TSTM C380 TSTM1 C39B TSTM4 C39F TSTM5 C3A1 UPDAT C045 UPDTX C3EA WRT C194 WRT1 C1DB WRT4 C1DE WRTN C036 XUS C1EO XUSV	COD7 BYTEC C306 BYTEO C31A C32E BYT03 C337 CCTRL 0000 C2D9 CHK C228 CHK1 C22F C27C CNTRL 00C0 C0 C006 C288 CRRDY 0001 CTRDY 0080 00C1 DSPYM C346 FDOS C015 C2E1 HIGH C421 HLC0 C33A C182 INCDB C18D INIT C29D C238 ISCTR C437 ISIZE C434 C433 KBINT C373 LER C2D1 C205 L00P1 C20A L00PV C02D C358 MEM1 C35B MEM9 C36F C2B4 NBL C0EE NIO C0FC C431 OSCTR C43C OSIZE C439 C432 PARAM C2E8 PARM1 C2EB C24B PASS3 C25C PASSV C039 C3D2 PROG C04E PROM C000 C009 RED0 C096 RED1 C0B6 C0D5 REDX C093 RESET C054 C203 RFLGV C02A RI C109 C133 RI3 C136 RI4 C15B C149 RIV C033 RIX C100 C01E SCTCH C400 SEEK C1F6 C47F START C41B TBL C25F C42F TSTM C380 TSTM1 C3A3 C39B TSTM4 C39F TSTM5 C3BE C3A1 UPDAT C045 UPDTX C41E C3EA WRT C194 WRT1 C1A9 C1DB WRT4 C1DE WRTN C211 C036 XUS C1E0 XUSV C021	COD7 BYTEC C306 BYTEO C31A BYTO1 C32E BYTO3 C337 CCTRL 0000 CDATA C2D9 CHK C228 CHK1 C22F C1 C27C CNTRL 00C0 C0 C006 C0X C288 CRRDY 0001 CTRDY 0080 DATAI 00C1 DSPYM C346 FDOS C015 FDOS1 C2E1 HIGH C421 HLC0 C33A ICNTR C182 INCDB C18D INIT C29D INIT1 C238 ISCTR C437 ISIZE C434 ITRK C433 KBINT C373 LER C2D1 L0 C205 L00P1 C20A L00PV C02D L0X C358 MEM1 C35B MEM9 C36F MNTR C2B4 NBL C0EE NIO C0FC OCNTR C431 OSCTR C43C OSIZE C439 OTRK C432 PARAM C2E8 PARM1 C2EB PASS C24B PASS3 C25C PASSV C039 PO C3D2 PROG C04E PROM C000 RDIX C009 RED0 C096 RED1 C0B6 RED2 C0D5 REDX C093 RESET C054 RESTR C203 RFLGV C02A RI C109 RI10 C133 RI3 C136 RI4 C15B RI5 C149 RIV C033 RIX C100 RSTRV C01E SCTCH C400 SEEK C1F6 SEEKV C47F START C41B TBL C25F TISZE C42F TSTM C380 TSTM1 C3A3 TSTM2 C39B TSTM4 C39F TSTM5 C3BE TSM6 C3A1 UPDAT C045 UPDTX C41E VCTRS C3EA WRT C194 WRT1 C1A9 WRT2 C1DB WRT4 C1DE WRTN C211 WRTN2

APPENDIX D

RESIDENT LISTING

C X FORCESTA

BESTDENT 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 O(OR NOTHING), 1, 2, OR 3
; T=TRACK
; S=SECTOR
       -CLEAR DRIVE ELECTRONICS
; A
BU,T -SEEK TO TRACK
      -READ TO BUFFER FROM PRESENT TRACK
; DU, S
; FU, S
       -WRITE FROM BUFFER TO PRESENT TRACK
       -RD/WRT (BFR) CONTINUOUS ON PRESENT TRACK
; GU, S
; HU
       -TRKO TO TRK76 LOOP
; I
       -UNIT SELECT TEST
; JU
       -SEEK TEST ONCE (2 MIN)
       -SEEK TEST CONTINUOUS
; KU
       -SEEK TEST READ ONLY
; LU
; MU
       -DD MARK TEST ONCE
       -RETURN TO MONITOR
i N
: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

- PROB IT DIMENSETIC FOR ALTERNATIONAL DESCRIPTION
 - HOOK MOUTHOOL THE THATE I
 - LOAD A SCHAFER DISCRIPT INTO THE DRIVE UNIT TO
 - CENTRAL THE DESIRED TEST TO BE PERFORMED.
 - CENTENDED VERSES MAY BE MANUALLY ADDRESS.
 - "D- DOW AMERICAN VILL
 - E RO E . I . (DMINTOM ROTO FROMEN TIME-U:
 - MONGSTOT .
 - AUTOBE-8 I
 - COLUMN DRIVE ELECTRONICS
 - HOLY -SERVICE TABLE
 - DULS BEER TO BUFFER FROM PRESENT TRACK
 - PLACE THE SHARE AT RESIDENCE WHEN THE PARTY OF THE
 - CONTINUES REPORT (DER) CONTINUOUS ON PRESENT TRACK
 - SOUTH ATTEST OF BOOKEN DE
 - U -- PEREN YEST ONCE (2 MIN
 - SOCIAL LINES LEST CONTINUOUS
 - V MAN MARKS TOWN WINDOW THE
 - JOHO TEST JOHN GO- UP .
 - SCTINOS OT MOUTES- N

penders on tarting

- of ORC EMBOR ON BEAD 5 TIMES OILTRY LIGHTLASTRY
- 02 CRO ERROR ON WRITE 5 TIMES DZITRKO (UNIT/SCTR)
 - CARTON CONTRACTOR OF THE PARTY OF THE PARTY
 - (ITTSYS) (ITTSE) SOME TOSUM TIME AC
 - CREASE CHIEF CONTRACTOR CONTRACTOR MORROR CONTRACTOR MORROR
 - TOTAL SOURS BOOK DO AND
 - CHIOSA MONNO NEWS IN ROY
 - MUNICIPAL IN LOCAL LOCALINA

```
C000
               PROM
                       EQU
                               OCOOOH
               ; ****
               ; RESIDENT 8080 ALTAIR/IMSAI/POLY 88 FD0S III VERSION 1.0
                            LOADS FIRS AND BRANCHES
               ; ENTRY ADDRESSES-
                       POWER UP = C3E7 HEX
                       RE-ENTRY = C3E4 HEX
    OLE OUTPUT VECTOR
               UCTRESS PRESENT INPUT
0001
               DATAO
                       EQU
                               OC1H
              DATAI
0000
                       EQU
                               OCOH
0000
                               OCOH
               CNTRL
                       EQU
                                        CONSOLE CONTROL PORT
0000
               CCTRL
                               0
                       EQU
0001
               CDATA
                       EQU
                               11100
                                        CONSOLE DATA PORT
                                        CONSOLE DATA READY
0001
               CRRDY
                       EQU
                                        CONSOLE XMIT READY
0080
               CTRDY
                       EQU
                               80H
C400
               SCTCH
                       EQU
                               OC400H ; SCRATCH RAM
C400
               VCTRS
                       EQU
                               SCTCH
                                       ; I/O VECTORS
C430
               BASE
                       EQU
                               SCTCH+30H
C47F
               STACK
                       EQU
                               SCTCH+7FH
                               BASE
              PASS
C430
                       EQU
C431
              OFILE
                       EQU
                               BASE+1
C432
              OUNIT
                       EQU
                               BASE+2
C433
               IUNIT
                       EQU
                               BASE+3
C434
                       EQU
               ISIZE
                               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
                       EQU
              OCNTR
                               BASE+13
C42F
              TITRK
                       EQU
                               BASE-1
C43E
              TISZE
                       EQU
                               BASE+14
                       EQU
                               VCTRS+24
C418
              ASMB
C41B
                       EQU
                               VCTRS+27
              START
C41E
              UPDTX
                       EQU
                               VCTRS+30
```

0000 PROM ORG 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 COOC C309C4 LO: JMP VCTRS+9 ; LIST OUTPUT VECTOR COOF C30CC4 P0: JMP VCTRS+12 PUNCH OUTPUT VECTOR C012 C30FC4 MNTR: JMP VCTRS+15 ; SYSTEM MONITOR VECTOR C015 317FC4 FDOS: LXI SP, STACK CO18 CD5ECO CALL FDOS1 JMP CO1B C31BC4 START JMP CO1E C354CO RSTV: RESET C021 C3E0C1 XUS XUSV: JMP. C024 C3EFC1 XXUSV: JMP XXUS CO27 C3F7C1 SEEKV: **JMP** SEEK+1 C02A C303C2 RELGV: JMP RFLAG CO2D C305C2 LOOPV: JMP LOOP-C030 C37AC0 RSTRV: JMP RESTR RI 0033 030901 RIV: JMP WRT WRTV: C036 C394C1 JMP C039 C338C2 PASSV: JMP **IPASS** ; ASMB INTERPASS FNC C03C CD93C0 ASSEM: CALL REDX COSF CD7ACO CALL RESTR SILVO C042 C318C4 JMP ASMB

```
SP, STACK
CO45 317FC4 UPDAT:
                      LXI
 CO48 CD5ECO
                      CALL
                             FD0S1
                      JMP
                            UPDTX
 CO4B C31EC4
         CALL THE
 CO4E CD93CO PROG:
                      CALL
                             REDX
 C051 C312C0
                      JMP
                             MNTR
 C054 3E81
              RESET:
                      MVI
                             A, 81H
 CO56 CD05C2
                      CALL
                             LOOP
 C059 3E0D
                      MVI
                             A, ODH
 C05B C305C2
                             LOOP
                     JMP
                            PUSH
 COSE CD54CO FDOS1:
                             RESET
                      CALL
                             H, 0
 C061 210000
                                    ; SET BIAS=0
                      LXI
                                     00 2800
 C064 E5
                      PUSH
 C065 216900
                      LXI
                             H, 105
                             ISIZE HOSP COVERS AND
 C068 2234C4
                      SHLD
 C06B 2136C4
                      LXI
                             H, ITRK
                             M. 1
                                    ; TRACK=1
 CO6E 3601
                      MVI
 C070 2C
                                    ; SECTOR=0
                      INR
                             23g
 C071 3600
                      MVI
                             M, O
 C073 2C
                             LIAG
                                    READ BFR EMPTY
                      INR
                                       STREET CONTES
 C074 3600
                      MVI
                             M, O
 CO76 CD93CO
                     CALL
                             REDX
                                  : GO TO FDOS
 C079 C9
                      RET
 C07A 2A3EC4
              RESTR:
                     LHLD
                             TISZE ; RESTORE IFILE POINTERS
                                   0380AG 1003
 CO7D 2234C4
                     SHLD
                             ISIZE
 C080 3A2FC4
                      LDA
                             TITRK
 C083 3236C4
                      STA
                             ITRK
 C086 3A33C4
                      LDA
                             IUNIT
 C089 OF
                      RRC
 COSA OF
                      RRC
 CO8B 3237C4
                      STA
                             ISCTR
 COSE 97
                             A
                      SUB
 COSF 3238C4
                      STA
                             ICNTR
 C092 C9
                      RET
```

;SUBROUTINE TO READ A HEX FILE INTO MEMORY ;STARTS WITH ROUTINE REDO, USES ALL REGISTERS

```
REDX: POP H ; SWAP BIAS & RETURN
                CO93 E1
                                                    XTHL
                CO94 E3
                CO95 E5
                                                                       PUSH H
                                                                                                   H ;GET BIAS
                C096 E1
                                                      REDO:
                                                                             POP
                                                                                          RIX ; GET CHAR INTO A
                                                      PUSH
CALL
MVI
                C097 E5
                C098 CD00C1
                C09B 063A
                                                                SUB
JNZ
MOV
                                                                                                    В
                CO9D 90
                C09E C296C0
                                                                                                    REDO
                COA1 57
                                                                                                   D. A
                                                              CALL
                                                                                                    BYTE TERM 1838 1800
                COA2 CDD7CO
                                                                                              RED2
E, A
BYTE
                                                             JZ
MOV
                COA5 CAC8CO
                COA8 5F
                                                               CALL
PUSH
                COA9 CDD7CO
               COAC F5
COAD CDD7CO
CALL BYTE
COBO C1
COB1 4F
COB1 4F
COB1 B
COB1
                                                                                            PSW
              BYTE
H, A
BYTE
L, A
                COC8 CDD7CO RED2: CALL
                                                    MOV
                COCB 67
                COCC CDB7CO
                                                                 MOV
                                                                           CALL
                COCF 6F
CODO B4 ORA
                COD1 CAD5CO JZ
                                                                                                  RED3
                                                                                                  RED3
H
RIX
NBL
                COD4 E9
                                                                       PCHL
                                                  RED3: POP
                CODS E1
                COD6 C9
                                                                       CALL
                                                      BYTE:
                COD7 CDOOC1
                                                                       CALL
                CODA CDEECO
                CODD 07
                                                                         RLC
                CODE 07
                                                                            RLC
                CODF 07
                                                                             RLC
                COEO 07
                                                                           RLC
                COE1 4F MOV C, A
      COE2 CDOOC1 CALL RIX
```

```
CALL
ORA
MOV
ADD
      COE5 CDEECO
                                   NBL
                                   C BZISI
      COES B1
      COE9 4F
                                   C.A H
                                  D. A J.A VOM
A.C A AMA
A.C A AMA
      COEA 82
                         MOV
      COEB 57
      COEC 79
      COED C9
                          RET
                    SUBROUTINE TO CONVERT TWO HEX CHARACTERS
                  TO ONE BYTE

NBL: SUI 'O'

RC

ADI OE9H
                    ; TO ONE BYTE STUDIES
                NBL: SUI
      COEE DAGO
      COFO D8
      COF1 C6E9
      COF3 D8
                           ADI 6
      COF4 C606 ADI 6
COF6 F2FCCO JP NIO
COF9 C607 ADI 7
COFB D8 RC
COFC C60A NIO: ADI 10
COFE B7 ORA A
COFF C9 RET
      COF4 C606
                    ; SUBROUTNE TO READ A BYTE FROM DISK
                   PLACES CHAR IN A-REG. ENTRY AT RI READS 8 BITS,
                   ENTRY AT RIX READS 7 BITS.
                          CALL RI
JC MNTR
ANI 7FH
RET
      C100 CD09C1
                    RIX:
                         CALL
      C103 DA12C0
                   JC MNTR
      C106 E67F
      C108 C9
                   ;
                   i
                  RI: PUSH B ; SAVE REG D-L
PUSH H
LXI H, ICNTR
MOV A, M
      C109 C5
      C10A E5
      C10B 2138C4
                          MOV A, M
ANA A ; CNT=0?
JNZ RI10 ; NO
MVI L, ISCTR AND OFFH ; YES-INCR D. A.
      C10E 7E
C10F A7
C110 C26EC1 ANA
G113 2E37 RI5: MVI
```

```
CALL INCDA
LHLD ISIZE
DCX H
SHLD ISIZE
MOV A,L
ANA A
JNZ RI3
MOV A,H
ANA A
C115 CD82C1
C118 2A34C4
C11B 2B
C11C 2234C4
C11F 7D
C120 A7
C121 C236C1
C124 7C
                     ANA
C125 A7
                                A
                               RI3
C126 C236C1
                    JNZ
C12A 2234C4 SHIT
                                H
                                ISIZE ADD BY BEILD BEILD
                       SHLD
                                H, ICNTR
                LXI
C12D 2138C4
M, O ; SET EOF : RESTORE D-L
C130 3600
C136 2137C4 RI3: LXI H, ISCTR ; XMIT UNIT/SECTOR
C139 CDEOC1 CALL XUS
C13C CD28C2 CALL CHK ; MAKE SURE A DISK
                                CHK ; MAKE SURE A DISK
L ; SET CNTR=128
                     CALL
INR
MVI
C13F 2C
C140 3680 MVI
C142 0E05 MVI
C144 2E36 MVI
C146 CDF6C1 CALL
C149 3E03 RI6: MVI
C14B CD05C2 CALL
C13F 2C
                               C,5 ;SET TRY CNT=5
                              L, ITRK AND OFFH ; SEEK TRACK
                             SEEK
A, 3
                                        READ DATA
                              LOOP
C14E DBCO
                               DATAI ; DD MARK?
                      IN
                                80H
RI4
C150 E680
                        ANI
                                RI4 ; NO
RFLAG
RI5
C152 CA5BC1 JZ
C155 CD03C2 CALL
C158 C313C1 JMP
C15B DBCO RI4: IN
C15D E608
C15F CA6EC1
C162 CD03C2
                                DATAI ; CRC ERROR?
                       IN
ANI
JZ
CALL
                                8H
RI10 ; NO
RFLAG
                                C ; DECR CNTR
RI6
A, 1
                       DCR
C165 OD
                        JNZ
C166 C249C1
C169 3E01
                        MVI
                       JMP CHK1
C16B C32FC2
C16E 3E40 RI10: MVI A,40H ; READ BYTE INTO A
C170 D3C0 OUT CNTRL
C172 DBC0 IN DATAI
C174 4F MOV C,A
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 C
                                                     ; DECR READ COUNTER
```

```
C17C 35 DCR M
C17D 79 MOV A, C
C17E B7 ORA A
    C17F C333C1 JMP RI2
              UNIZ WRTZ TRY ADALM
NVI A OFH IMRITE AS DD
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
   C18B 2D DCR L
C18C C9 RET
C18D 7E INCDB: MOV A, M
C18E E6C1 ANI OC1H
C19O 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
                             MOV A, C
PUSH H
OUT DATAO ; OUTPUT HAR
    C194 79 WRT:
             WRT:
   C195 E5
   C196 D3C1
C198 3E31
  C198 3E31
C19A CD05C2
CALL LOOP
C19D 213DC4
LXI H, OCNTR
C1AO 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, O ; CLEAR COUNT
LXI H, OSCTR ; XMIT UNIT/SECTOR
                             MVI
                                       A, 31H
   C1AC CDEOC1
                             CALL XUS
   C1AF CD28C2
                             CALL CHK ; MAKE SURE A DISK
   C1B2 0E05
                             MVI C,5 ;SET TRY CNT=5
DCR L ;SEEK TRACK
CALL SEEK
   C1B4 2D
   C1B5 CDF6C1
                   WRT2: MVI A,5 ; WRITE DATA CALL LOOP
   C1B8 3E05
   C1BA CD05C2
   C1BD 3E07
                             MVI , A,7 ; READ FOR CRC
                             CALL LOOP
   C1BF CD05C2
   C1C2 DBCO
                             IN
                                       DATAI ; CRC ERROR?
```

```
C1C4 E608 ANI 8H
C1C6 CADBC1 JZ WRT3 ; NO
C1C9 CD03C2 CALL RFLAG
C1CC OD DCR C ; DECR TRY CNT
C1CD C2B8C1 JNZ WRT2 ; TRY AGAIN
C1D0 3E0F MVI A, OFH ; WRITE AS DD
C1D2 CD05C2 CALL LOOP
C1D5 CD11C2 CALL WRTN ; INCREMENT DA & CHK SIZE
JZ INCDB
                   SUBROUTINE TO TRANSMIT UNIT/SECTOR BYTE
                              MOV 199 A. M. IMA
                             MOV TOO A, M THA
C1EO 7E
                   XUS:
                             PUSH H
LXI H, TBL-1 ; GET TABLE PNTR
ADD L ; MAKE SECTOR PNTR
MOV L, A
MOV C, M ; GET PHYS SECTOR
POP H
C1E1 E61F
C1E3 E5
C1E4 215EC2
C1E7 85
C1E8 6F
C1E9 4E
C1EA E1 POP H
C1EB 7E MOV A, M
C1EC E6CO ANI OCOH
C1EE B1 ORA C ; MERGE UNIT & PHYS SCTR
C1EF D3C1 XXUS: OUT DATAO
C1F1 3E21 MVI A, 21H
C1F1 3E21 MVI A, 21H
C1F3 C305C2 JMP L00P
   TWO ARE TWO; SUBROUTINE TO SEEK TRACK IN A
; 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, OBH
               ; SUBROUTINE TO ISSUE CMD & LOOP ON BUSY
```

```
CNTRL
A
CNTRL
DATAI
C205 D3C0 LOOP: OUT
C207 97 SUB
C208 D3C0 OUT
C20A DBC0 LOOP1: IN
              RAR
JC
                             LOOP1
C20C 1F
C20D DAOAC2
C210 C9
                        RET
               SUBROUTINE TO INCR DISK ADDR & CHK OFILE SIZE
                       MVI L, OSCTR AND OFFH
CALL INCDA
C211 2E3C WRTN:
                                INCDA
OSIZE
H
C213 CD82C1
               WRTN2: LHLD DCX
                       SHLD OSIZE
MOV A, L
ANA A
RNZ
MOV A. H
C216 2A39C4
C219 2B
              DCX
SHLD
C21A 2239C4
C21D 7D
C21E A7
C220 7C MOV A, H
C221 A7 ANA A
C222 C0 RNZ
C223 3E02 MVI A, 2
C225 C32FC2 JMP CHK1
C21F C0
                            CHK1
               SUBROUTINE TO CHECK IF A DISK, ELSE ERROS
               CHK: IN
                       IN DATAI
ANI 20H
C228 DBCO
C22A E620
                      RZ
MVI A,3
C22C C8
C22D 3E03
             ROUTINE TO PRINT ERR(E)
             CHK1: ORI 30H ; CONVERT TO ASCII
MOV C, A
CALL CO
JMP MNTR
C22F F630
C231 4F
C232 CD06C0
C235 C312C0
               ; INTERPASS FUNCTIONS
               ; IF BIT O OF (PASS) IS EQUAL TO 1, THEN BIT O 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 OO, JMP UPDAT OCCURS.
```

C238 C23B	3A30C4	IPASS:	LDA BAR	PASS
0230	D24BC2		JNC	PASS2
C23F	3A30C4		LDA	PASS
C242	3D		DOR	A
0243	323004		STA	PASS
C246	3E01		MVI	A, 1
C248	035002		JMP	PASS3
C24B	A7	PASS2:	ANA	Α
C24C	CA45C0		JZ	UPDAT
C24F	CD7AC0		CALL	RESTR
C252	3A30C4		LDA	PASS
C255	1F		RAR	
C256	F5		PUSH	PSW
C257	97		SUB	A
C258	323004		STA	PASS
C25B	F1		POP	PSW
C25C	C630	PASS3:	ADI	30H
C25E	C9		RET	

; PHYSICAL SECTOR TABLE. IS IN ORDER ; OF LOGICAL SECTOR NUMBER.

TBL:	DB	1
	DB	OAH
	DB	13H
	DB	2
	DB	OBH
	DB	14H
	DB	3
	DB	OCH
	DB	15H
	DB	4
	DB	ODH
	DB	16H
	DB	5
	DB	OEH
	DB	17H
	DB	6
	DB	OFH
	DB	18H
	DB	7
	DB	10H
	DB	19H
	DB	8
	DB	11H
	DB	1AH
	DB	9
	DB	12H
	TBL:	DB D

```
NOP
C279 00
                    NOP
C27A 00
C27B 00
                    NOP
                    CONSOLE INPUT ROUTINE
            CIX: IN CCTRL
ANI CRRDY
JNZ CIX
IN CDATA
C27C DB00
C27E E601
C280 C27CC2
C283 DB01
                            7FH
C285 E67F
                    ANI
C287 C9
                    RET
                    C2D9 CD03C0 CECHO: CALL . CI .
C2DC AF MOV . C. A
C2D0 CD06C0 . C6LL . C0
             i
                    MVI
                            C, ODH
             CRLF:
C288 OEOD
                            C0
C, OAH
                    CALL
C28A CD06C0
                    MVI
C28D OEOA
                            CZE1 CDERCZ GO. CAO
C28F C306C0
                   JMP
                  CONSOLE OUTPUT ROUTINE
                            CCTRL
C292 DB00
             COX:
                    IN
                            CTRDY MARAS
C294 E680
                    ANI
                            COX
A, C
                    JNZ
C296 C292C2
                    MOV
C299 79
C29A D301
                     OUT
                            CDATA
                           CPI
RZ
RAD
C29C C9
                    RET
                           B, OC3OBH
H, VCTRS
D, VECTR
M, B
                    LXI
C29D 010BC3
             INIT:
C2A3 11EAC3
                    LXI
                    LXI
                            M, B
H
D
M, A
H
D
C2A6 70
             INIT1: MOV
                            H JAS
C2A7 23
                    INX
C2A8 1A
                    LDAX
C2A9 77
                    MOV
                    INX
C2AA 23
C2AB 13
                    INX
                            D
C2AC 1A
                    LDAX
                            M, A
C2AD 77
                    MOV
                            HIAS COSTVE
C2AE 23
                   INX
C2AF 13
                    INX
                            INIT1
C2BO OD
                    DCR
C2B1 C2A6C2
                     JNZ
                    LXI
C2B4 317FC4 MNTRX:
                            SP, STACK
C2B7 CD88C2
                    CALL
                            CRLF
```

C2BA OE3E		MVI	C, SEH		
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	4G4		
C2CE CAE1C2		JZ	GO		
C2D1 OE3F	LER:	MVI	0,171		
C2D3 CD06C0		CALL	CO		
C2D6 C3B4C2		JMP	MNTRX		
	;				
	i				
C2D9 CD03C0	CECHO:	CALL	CI		
C2DC 4F		MOV	C, A		
C2DD CD06C0		CALL	CO		
C2E0 C9		RET			6920
	,				C28A
	i				
C2E1 CDE8C2	GO:	CALL	PARAM		
C2E4 CD88C2		CALL	CRLF		
C2E7 E9		PCHL			
	i				
	; = a = a = a = a				
C2E8 210000	PARAM:	LXI	H, O		
C2EB CDD9C2	PARM1:	CALL	CECHO		
C2EE FEOD		CPI	ODH		
C2F0 C8		RZ	1729		
C2F1 FE2C		CPI RZ	,		
C2F3 C8 C2F4 29		DAD	Н		
C2F4 29 C2F5 29		DAD	HIXL		
C2F6 29			HTX.J		
C2F7 29		DAD	HIXI		
C2F8 DAD1C2		JC	LER		
C2FB CDEECO		CALL	NBL		
C2FE DAD1C2		JC	LER		
C301 B5		ORA	Vom		
C302 6F		MOV	L, A		
C3O3 C3EBC2		JMP	PARM1		
	į	Œ.			
	i				
C306 CDD9C2	BYTEC:	CALL	CECHO		
C309 CDEECO	BYTC1:	CALL	NBL		
C30C 07		RLC			
C30D 07		RLC			
C30E 07		RLC			
C30F 07		RLC			
C310 F5		PUSH	PSW		

8

C311	CDD9C2		CALL	CECHO		
C314	CDEECO			NBL		
0317			POP	B		
C318			ORA	В		
C319			RET	VOM		
~~~	0,7		1.1			
		;				
0010	F5	BYTEO:	DUCH	PSW		
		BTIEU.		PVTO1		
	CD2AC3		CALL	BYT01		
C31E			MOV			
	CD09C0		CALL	CO		
C322			POP	PSW		
	CD2EC3		CALL	PSW BYTO2 C, A CO		
C326			MOV	C, A		
C327	C309C0		JMP	CO		
		;				
		i				
C32A	OF	BYTO1:	RRC			
C32B	OF		RRC			
C32C	OF		RRC RRC			
C32D	OF		RRC			
C32E	E60F	BYTO2:	ANI	OFH		
0330	FEOA		CPI H	OAH		
C332	FA37C3		JM	BATUS		
C335	C607		ADI	BYTO3		
		BYTO3:		зон		
C337		B1103.	RET	SOH		
0007	CZ		FLISTO			
		,				
	000000		0.01.1	1 VW		
		HLCO:	CALL	CRLF		
C33D			MOV	A, H BYTEO		
	CD1AC3		CALL	BYTEO		
C341	7D		MOV	A, L		
	CD1AC3		CALL	BYTEO		
C345	C9		RET			
		j				
		i				
C346	CD3AC3	DSPYM:	CALL	HLCO		
C349	OE3D		MVI	C, 4=4		
C34B	CD06C0		CALL	CO		
C34E	7E		MOV	A, M		
C34F	CD1AC3		CALL	BYTEO		
C352	0E20		MVI	C, 20H		
	CD06C0		CALL	co		
C357	020000		RET	SHO		
		;	HOIH			
		,				
CORO	CDE8C2		CALL			
		MEM:		PARAM DSPYM		
	CD46C3	MEM1:	CALL			
CSOE	CDD9C2		CALL	CECHO		

C363 C366 C368	FEOD CAB4C2 FE20 CA6FC3 CD09C3 77		CPI JZ CPI JZ CALL MOV	MNITEV		
	23 C35BC3	MEM9:	INX	H MEM1		
C375	EGUI	2	IN 05	CCTRL		
C378	CO DBO1 FEO3		RNZ IN CPI	CDATA 3		
C37C C37F	CA12C0 C9	1	JZ RET			
C421		; HIGH	EQU	SCTCH+2	1H	
C383 C384	EB	TSTM:	PUSH	H ISO		
C388 C38B	CDE8C2 2221C4 EB CD88C2		CALL SHLD XCHG CALL	PARAM HIGH		
	3600 7B	TSTM2:	MVI	M, O A, E		
C396 C397	C29BC3 7A BC		JNZ	TSTM3 A, D H TSTM4		
C39B	CA9FC3 23 C38FC3	TSTM3:	INX	H TSTM2		
C39F C3A1 C3A2		TSTM7	MVI POP PUSH	HILLIAG		
C3A3 C3A4 C3A5	34 7B	TSTM1:	INR	M A,E M		
C3A6	C4C2C3 3A21C4		CNZ LDA CMP	TSTM6 HIGH L		
	C2BEC3 3A22C4		JNZ LDA	TSTM5 HIGH+1		

C3B3 BC C3B4 C2BEC3 C3B7 1C C3B8 CD73C3 C3BB C3A1C3		CMP JNZ INR CALL JMP	H TSTM5 E KBINT TSTM7			
C3BE 23 C3BF C3A3C3	TSTM5:	JMP	H TSTM1			
C3C2 CD46C3 C3C5 7B C3C6 CD1AC3 C3C9 C388C2	TSTM6:	CALL MOV CALL JMP	DSPYM A,E BYTEO CRLF			
C3CC C303C0	; ; ; RDIX:					
C3CF C306C0	Lox:		CO			
C3D2 C306C0	POX:	JMP	CO			
C3E4	C108	I/O VEC	TOR TABL	E ADDRESS		
C3E4 C3B4C2	#853 F863	JMP MNT	RX			
C3E7 C39DC2		JMP	INIT	; ***	****	
C3EC 92C2 C3EE CCC3 C3F0 CFC3 C3F2 D2C3 C3F4 B4C2 C3F6 09C1 C3F8 94C1 C3FA 4000 C3FC 4000 C3FE 4300			COX RDIX LOX POX MNTRX RI WRT 40H	; CONSOLE ; CONSOLE ; PAPER T ; LINE PR ; PUNCH V ; MONITOR ; DISK RE ; DISK WR ; ASSEM/E ; EXECUTI ; UPDAT V	OUT VECTOR  APE READ INTER VECTOR  VECTOR AD VECTO ITE VECT  VE VECTO  VE VECTO	TOR ER VECTOR CTOR R OR OR;
		END				

ASMB	C418	ASSEM	COSC	BASE	C430	BYTC1	C309
BYTE	COD7	BYTEC	C306	BYTEO	C31A	BYT01	C32A
BYT02	C32E	<b>BYTO3</b>	C337	CCTRL	0000	CDATA	0001
CECHO	C2D9	CHK	C228	CHK1	C22F	CI	C003
CIX	C27C	CNTRL	0000	CO	C006	COX	C292
CRLF	C288	CRRDY	0001	CTRDY	0080	DATAI	0000
DATAO	0001	DSPYM	C346	FDOS	C015	FDOS1	COSE
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	COOC
LOOP	C205	LOOP1	C20A	LOOPY	CO2D	LOX	CBCF
MEM	C358	MEM1	C35B	MEM9	C36F	MNTR	C012
MNTRX	C2B4	NBL	COEE	NIO	COFC	CONTR	C43D
OFILE	C431	OSCTR	C43C	OSIZE	C439	OTRK	C43B
OUNIT	C432	PARAM	C2E8	PARM1	C2EB	PASS	C430
PASS2	C24B	PASS3	C25C	PASSV	C039	PO	COOF
POX	C3D2	PROG	CO4E	PROM	C000	RDIX	C3CC
RDRIN	C009	REDO	C096	RED1	COB6	RED2	C0C8
RED3	COD5	REDX	C093	RESET	C054	RESTR	CO7A
RFLAG	C203	RFLGV	C02A	RI	C109	RI10	C16E
RIZ	C133	RIS	C136	RI4	C15B	RI5	C113
RI6	C149	RIV	C033	RIX	C100	RSTRV	030
RSTV	CO1E	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	CSEA	WRT	C194	WRT1	C1A9	WRT2	C1B8
WRT3	C1DB	WRT4	C1DE	WRTN	C211	WRTN2	C216
WRTV	C036	XUS	C1EO	XUSV	C021	XXUS	C1EF
XXUSV	C024						