SECTION 6

USING AIM 65/40 DOS VERSION 1.0

The AIM 65/40 Disk Operating System (DOS) Version 1.0 (A65/40-7090) integrates the RM 65 FDC primitive subroutines with fundamental file management functions for use on the AIM 65/40 Microcomputer. These ROM-based functions, contained in a 4K-byte R2332 ROM, are accessible from the operator through the I/O and Debug Monitor/Text Editor ROMs as well as language ROMs and from an application program. Being ROM-based, DOS operation may proceed immediately upon power turn-on without waiting for separate loading of the DOS into RAM.

Text and program source code may be written to, and read from, disk with the Editor List and Read commands, respectively. Similarly, binary data and program object code may be written to, and loaded from, disk using the Monitor Dump and Load commands, respectively. Files containing source and object code for application programs written in AIM 65/40 Assembler, BASIC, FORTH, and PL/65 languages are, therefore, supported.

DOS primary commands are selected by the operator at the Monitor command level. These commands invoke the following functions:

Format a Disk
List the Directory
Backup a Disk
List a File
Delete a File
Recover a File

Files are created automatically upon writing a file to disk. A file name and the disk drive number are operator entered in response to system prompts.

Disk read or write errors, both at the DOS and FDC device level, are reported upon detection. User-alterable variables allow changing of default values to application unique values.

6.1 INITIALIZATION

After installing the AIM 65/40 DOS ROM on the RM 65 FDC module, the DOS is ready for use. Be sure that address range $8000-$8FFF is selected for off-board operation on the AIM 65/40 SBC module.

The DOS is initialized automatically through the I/O ROM Auto-start linkage. The FDC module primitive subroutine variables are initialized, the DOS variables are set to default values (see Section 6.7), the IRQ vectors are initialized, the Monitor key decode linkage is altered (to allow DOS to decode the [and] keys), the user-defined I/O vectors are loaded for floppy disk I/O, and other temporary variables are initialized. Control then returns to the I/O ROM and subsequently to the Monitor command level. DOS commands may be entered when the Monitor prompt is displayed (the DOS may not be operated without the Monitor/Editor ROMs installed).

AIM 65/40 DOS 1.0 is designed primarily to support the AIM 65/40 Monitor/Editor and language functions in response to commands entered from the keyboard with messages directed to the display/printer. When used in this manner, the Monitor/Editor ROMs must be installed. The DOS may also be used to write and read a data file under program control (see Section 6.5). In this case, the Monitor/Editor ROMs are not required unless a Monitor subroutine is used.

NOTE

The DOS uses two contiguous 256-byte buffers, the output buffer and the input buffer, to transfer data between RAM and a floppy disk (as well as for other intermediate data storage). The high byte of the output (or source) buffer is specified by the variable BUFFER (see Section 6.7). The input (or destination) buffer is located immediately above the output buffer. BUFFER is initialized by a cold reset to $3E which locates the buffers at the top of 16K bytes (i.e., $3E00-$3FFF). If

NOTE (Cont'd)

more RAM is available, you will probably want to move this buffer to the top of RAM, e.g., change the Buffer value to $7E for 32K bytes of RAM.

6.2 DISK FORMAT

A floppy disk must be formatted in the AIM 65/40 DOS 1.0 format before it can be (see Section 6.3.3) used. The DOS disk format (illustrated in Figure 6-1) reserves track 0 for future use, uses track 1 and one-half of track 2 for the directory, and the rest of the tracks for sector header and data storage. The sectors are formatted in an interleaved manner. Files are stored sequentially on the disk.

Trk No.	Sect No.	Function	Single-Density			Double-Density		
			Byte No.	File No. 5"	File No. 8"	Byte No.	File No. 5"	File No. 8"
0	All	Reserved	1-128	-	-	1-256	-	-
1	1	File Entry	1- 16	1	1	1- 16	1	1
		File Status	1			1		
		00=Directory End 01=Active File 02=Deleted File						
		File Name	2- 11			2- 11		
		Start Track	12			12		
		Start Sector	13			13		
		Length-Low Byte	14			14		
		Length-High Byte	15			15		
		Null (00)	16			16		
		File Entries	17-128	2- 8	2- 8	17-256	2- 16	2- 16
	2	File Entries	1-128	9-16	9- 16	1-256	17- 32	17- 32
	3- 8	File Entries	1-128	17-64	17- 64	1-256	33-128	33-128
	9-13	File Entries	1-128	-	65-104	1-256	-	129-208
	14-16	Free	1-128	-	-	1-256	-	-
2	All	Free	1-128	-	-	1-256	-	-

Figure 6-1. AIM 65/40 DOS 1.0 Disk Format

6.3 PRIMARY OPERATOR COMMANDS

The primary operator commands are selected by using one of two keys: [and]. The [key directly commands the List Directory function, (see Section 6.3.1) while the] key displays a UTILITY= prompt. One of five utility functions may then be commanded (see Sections 6.3.2 through 6.3.6). If an invalid key is pressed for a utility function, the utility menu is displayed, e.g.:

```
{]} UTILITY=<RETURN>
(F)ORMAT, (B)ACKUP, (L)IST, (D)EL, (R)EC
```

The <ESC> key is vectored through variable FESCIV (see Section 6.7) to jump to the Monitor command level. This vector is initialized upon cold reset and may be user-altered. The <ESC> key vector is also used to return control to the Monitor after reporting an error condition (see Section 6.6). Note that some other major functions reconfigure the Escape vector to return to their command level, e.g., the Text Editor and BASIC. In these cases, if the <ESC> key is pressed, the function in process will be terminated.

> **NOTE**
>
> Upon cold reset, the density selection corresponding to each disk drive is initially set to double. If a disk cannot be read at the selected density, DOS automatically switches to the other density, and attempts to read the disk again. If the read is successful, the selected density is saved for future reading from that disk drive (until a cold reset is performed). This process is essentially transparent to the operator except for a slight delay.
>
> If DOS cannot read the disk at either density, the operator will be prompted to enter the disk density (e.g., DENSITY=) upon the next disk

> **NOTE (Cont'd)**
>
> command. The DOS will then attempt to read the disk again at the commanded density (and, if necessary, the other density).

6.3.1 List the Directory

The List Directory function displays the file name of all files recorded on the disk, the sector length of the files and the active/deleted status of each file. The number of free sectors, i.e., the amount of space left on the disk, is also reported.

To list the directory, press the [key, then enter the disk drive number in response to the prompt.

Example:

To list the contents of the directory on disk drive No. 1:

```
{[} DISK=1
FILE NAME     LEN S
LOGGER1.OBJ   269 A
TEXT1         193 A
TEXT2          26 A
SEC LEFT= 178
```

The data output rate may be altered by pressing the Ø key (fastest) through the 9 key (slowest) while the directory is being listed, or by using the Monitor @ command prior to the List Direct command.

6.3.2 Backup a Disk

The Backup Disk function copies all the active files from one disk to another disk. Since deleted files on the input disk are not copied to the output disk, the newly created disk is essentially compressed (i.e., deleted files removed) to provide additional free space.

To backup a disk, press the] key to request the utilities prompt, then press the B key. Enter the disk drive number of the disk to copy from, then the disk drive number of the disk to copy to, in response to prompts.

Example:

To copy a disk on disk drive No. 1 to a disk on disk drive No. 2:

```
{]} UTILITY=B DISK=1
DISK=2
```

6.3.3 Format a Disk

The Format Disk function initializes a disk to prepare it for subsequent use (see Section 6.2). A new disk, i.e., an unformatted disk, must be formatted before a file can be written upon it. It may also be desired to initialize a previously formatted disk to remove all existing files before re-use.

The Format Disk function also verifies proper operation of the disk media before file storage use. The disk is first initialized by writing sector headers, gaps and CRC data, then a $EA byte pattern to all sectors. The directory is next initialized to all $00 bytes. If the NOVRFY flag (see Section 6.7) is non-zero (default value), the sector headers are verified, otherwise control returns to the Monitor command level. If the NOVRFY flag is zero, the sector headers are not checked, allowing a faster formatting of the disk.

To format a disk, press the] key to request the utilities prompt, then press the F key. Enter the desired disk number and density character when requested. The disk number may vary from 1 to 4 for single-sided disks and from 1 to 8 for double-sided disks (i.e., each side is treated as a separate drive number with odd numbers being the front side and even numbers being the back side). The density character may be either S (for single-density) or D (for double-density).

Since the initialization completely overwrites the disk, a command verification prompt, "ARE YOU SURE?", is displayed before continuing, to prevent inadvertent reformatting. Press Y to continue, or any other key to terminate the command.

The WAIT message is displayed during disk initialization, followed by the Monitor command prompt upon completion. The number of free sectors and bytes is dependent upon disk size and density while the initialization time is dependent upon the state of the NOVRFY flag:

Disk Size	Density Code	Approx. Format Time (Min:Sec) Verify NOVRFY≠0	Approx. Format Time (Min:Sec) No Verify NOVRFY=0	Free Sectors	Free Bytes
5"	S	1:45	0:20	535	68,480
5"	D	1:45	0:20	535	136,960
8"	S	5:00	0:30	1962	251,136
8"	D	5:00	0:30	1962	502,272

Example:

Initialize a 5" disk in disk drive No. 1 to single-density then list the directory to check it.

```
{]} UTILITY= F DISK=1 DENSITY=S
ARE YOU SURE?Y
WAIT

{[} DISK=1
FILE NAME  LEN S
SEC LEFT= 535
```

NOTE

1. An RM 65 DMA module (RM65-5104) must be used to write and read double-density on 8" disks.
2. If an RM 65 DMA module is used (with either 5" or 8" disk drives), refer to Section A.3 for operating instructions.

6.3.4 List a File

The List File function lists the contents of a file from a disk to another peripheral device. If the peripheral is another disk, a file may be copied from one disk to another.

To list a file, press the] key to request the utilities prompt, then press the L key. Enter the file name, disk drive number, and output device code as requested. Respond to output device code subprompts.

Example 1:

To list a text file to the display/printer:

```
{]} UTILITY=L FILE=TEXT3 DISK=1
OUT=<RETURN>
```

Example 2:

To list an object code file from disk drive No. 1 to disk drive No. 2:

```
{]} UTILITY=L FILE=PROG1.OBJ DISK=1
OUT=F FILE=PROG1.OBJ DISK=2
```

The data output rate to the display/printer may be altered by pressing the Ø key (fastest) through the 9 key (slowest) while the file is being listed, or by using the Monitor @ command prior to the List File command.

The listing may be suspended by pressing the <SPACE> bar, then resumed by pressing the <SPACE> bar again (or one of the other keys).

6.3.5 Delete a File

The Delete File Function changes a file from the active (A) state to the deleted (D) state. In the deleted state, the file cannot be accessed by file name by any function except Recover File (see Section 6.3.5).

To delete a file, press the] key to request the utilities prompt, then press the D key. Enter the file name and disk drive number in response to prompts. The Monitor prompt will be displayed upon completion. List the directory to verify the file deletion.

Example:

To delete file TEXT3 (currently active) from a disk on disk drive No. 1:

```
{]} UTILITY=D FILE=TEXT3 DISK=1
{[} DISK=1
FILE NAME  LEN S
TEXT1      193 A
TEXT2       94 A
TEXT3       63 D
SEC LEFT= 238
```

A deleted file may be returned to the active state by using the Recover File function.

6.3.6 Recover a File

The Recover File function changes a file from the deleted (D) state to the active (A) state. In the active state, the file can be accessed by file name by other DOS functions.

To recover a file, press] key to request the utilities prompt, then press the R key. Enter the file name and disk drive number in response to prompts. The Monitor prompt will be displayed upon completion. List the directory to verify file recovery.

Example:

To recover file TEXT3 (currently deleted) from disk drive No. 1:

```
{]} UTILITY=R FILE=TEXT3 DISK=1
{[} DISK=1
FILE NAME  LEN S
TEXT1      193 A
TEXT2       44 A
TEXT3       63 A
SEC LEFT= 238
```

6.3.7 Extension of UTILITY Functions

The DOS links to the UTILITY functions through two indirect jump vectors. These two vectors, MENUVC and MENULS, are user-alterable and may be changed to provide alternative linkage and functions. Upon depression of the] key, the DOS jumps through the before menu vector, MENUVC, to check for the five default command characters. If the command character is an F, B, L, D or R, the appropriate processing is performed (see Section 6.3.2 through 6.3.6). If the command character is not one of these letters, the DOS jumps through the after menu vector, MENULS, to display the command menu then redisplay the "UTILITY=" prompt.

Either one or both of these vectors may be altered to meet application dependent requirements.

6.4 FILE HANDLING UNDER OPERATOR CONTROL

A file may be written to, or read from, a disk under operator control by pressing F in response to the OUT=, or IN=, prompt displayed by other major functions, e.g., Monitor, Editor or language. Enter the file name and disk number in response to subprompts.

6.4.1 Using the AIM 65/40 Monitor

The Monitor Dump (D) function can be used to output machine code from memory to a disk file in either ASCII or binary

format, while the Monitor Load (L) function can be used to read the file from a disk into RAM (see Sections 4.8.2 and 4.8.1, respectively, in the AIM 65/40 System User's Manual).

a. Dumping a File in ASCII

Example:

Dump machine code, in ASCII, from addresses $0800 through $08FF to file PGM1*A on disk drive No. 1:

```
{D}
 FROM=0800 TO=08FF OFFSET=0000 MORE?N
 TYPE=A OUT=F FILE=PGM1*A      DISK=1
```

b. Dumping a File in Binary

Example:

Dump machine code, in binary, from addresses $0800 through $08FF to file PGM1*B on disk drive No. 1:

```
{D}
 FROM=0800 TO=08FF OFFSET=0000 MORE?N
 TYPE=B DUMP SYMBOLS?N OUT=F FILE=PGM1*B      DISK=1
```

c. Reading a File

Example:

Load file PGM1*B from disk drive No. 1 into RAM:

```
{L} OFFSET=0000 IN=F FILE=PGM1*B      DISK=1
```

6.4.2 Using the AIM 65/40 Editor

The Editor List (L) function can be used to output ASCII text from the Editor Text Buffer in RAM to a disk file while the Read (R) function can be used to read the file into the Text Buffer (see Sections 5.4.9 and 5.4.1, respectively, in the AIM 65/40 Systems User's Manual).

a. Writing a File

Example:

Write all the text in the Text Buffer to file PGM1 on disk drive No. 1:

```
={T}
={L}/. OUT=F FILE=PGM1       DISK=1
```

b. Reading a File

Example:

Read file PGM1 from disk drive No. 1 into the Text Buffer:

```
={R} IN=F FILE=PGM1       DISK=1
*END*
```

6.4.3 Using the AIM 65/40 Assembler

Source code may be read from a disk file into the AIM 65/40 assembler and/or object code may be written to a disk file from the assembler during the assembly process. If the object code output format is binary, the symbol table may also be saved on the disk file.

> **NOTE**
>
> The 8 key (Re-enter Assembler command) may not be used to assemble to/from disk.

Example 1:

Assemble source code from memory (i.e., the Text Buffer) and write the generated object code in ASCII to file PGM1*A on disk drive No. 2:


```
{7}ASSEMBLER V1.0
 FROM=1800  T0=1FFF
 IN=M
 OBJ TO MEM?N
 OUT=F FILE=PGM1*A       DISK=2
 TYPE=A
 LIST?N OUT=<RETURN>
 PASS 1
 PASS 2

 DONE
 ERRORS=0000
```

Example 2:

Assemble source code from file PGM1 on disk drive No. 1 and do not generate object code.

```
{7}ASSEMBLER V1.0
 FROM=1800  TO=1FFF
 IN=F FILE=PGM1       DISK=1
 OBJ TO MEM?N
 OUT=X
 LIST?N OUT=<RETURN>
 PASS 1
 PASS 2

 DONE
 ERRORS=0000
```

Example 3:

Assemble source code from file PGM1 on disk drive No. 1 and write the generated object code and symbol table, in binary, to file PGM1*BS on disk drive No. 1:

```
{7}ASSEMBLER V1.0
 FROM=1800  TO=1FFF
 IN=F FILE=PGM1       DISK=1
 OBJ TO MEM?N OFFSET=0000
 OUT=F TYPE=B DUMP SYMBOLS?Y FILE=PGM1*BS     DISK=1
 LIST?N OUT=<RETURN>
 PASS 1
 PASS 2

 DONE
 ERRORS=0000
```

> **NOTE**
>
> When assembling from a source file on disk and an error occurs (such as a File Not Found error), the AIM 65/40 must be reset to resume normal disk operation.

6.4.4 Using AIM 65/40 BASIC

The AIM/65/40 BASIC SAVE and LOAD functions may be used to write and read a BASIC application program, to and from a disk file, respectively (see Sections 4.3.10 and 4.3.5 in the AIM 65/40 BASIC User's Manual).

a. Saving a File

Example:

Save the BASIC program in RAM to file PGM3 on disk drive No. 1:

```
SAVE
OUT=F FILE=PGM3        DISK=1
```

b. Loading a File

Example:

Load a BASIC program into RAM file PGM3 on disk drive No. 1:

```
LOAD
IN=F FILE=PGM3         DISK=1
```

6.4.5 Using AIM 65/40 FORTH

The source code for an application program written in AIM 65/40 FORTH may be formatted to edit in the AIM 65/40 Text Editor or to input in the FORTH screen format. Refer to Sections 5.4.9

and 5.4.1 in the AIM 65/40 System User's Manual to write source code from the Text Editor to disk or to read source code from disk into the Text Editor. Refer to Section 12 in the AIM 65/40 FORTH User's Manual for FORTH screen format usage.

6.4.6 Using PL/65

The source code for an application program written in AIM 65/40 PL/65 may be read from disk into the PL/65 compiler and/or the generated object code (i.e., in 6502 assembly language) may be written to disk (see Section 2.6 in the AIM 65/40 PL/65 User's Manual).

a. Write a File

Compile source code from memory (i.e., the Text Buffer) and write the output assembler code to file PGM2L on disk drive No. 2:

```
{5}AIM 65/40 PL/65 V1.1
IN=M OUT=F FILE=PGM2L        DISK=1
PASS(1 OR 2)?2

ERRORS=00
```

b. Read a File

Compile source code from file PGM2L on disk drive No. 1 and output the generated assembly language to memory:

```
{5}AIM 65/40 PL/65 V1.1
IN=F FILE=PGM2L       DISK=1
OUT=M
PASS(1 OR 2)?2

ERRORS=00
```

6.5 FILE HANDLING UNDER PROGRAM CONTROL

Often it is desirable to store and/or retrieve data from a floppy disk entirely under program control, e.g., when collecting data in an unattended installation. While this can be easily done by writing a program which calls the primitive

routines (see Section 4), use of the primitive routines alone do not support the directory format used by AIM 65/40 DOS 1.0. DOS compatible data files written under program control allow easy examination and handling under operator control using DOS utility functions. This section describes a method to write data to, and read data from, a floppy disk in a format that is compatible with the AIM 65/40 DOS 1.0 file structure. To operate the DOS 1.0 under program control, the AIM 65/40 Monitor must be installed.

6.5.1 Writing and Reading Data Files

The AIM 65/40 DOS 1.0 file structure allows the writing and reading of data, to and from disk, independent of the data meaning (e.g., program source code, program object code, or symbol table) or data format (e.g., ASCII coding or machine code). The assembly listing in Figure 6-2 details a DOS compatible file handler. This handler contains both an output driver (to write a file to disk) and an input driver (to read a file from disk). The DOS subroutines and variables used by these drivers are listed in Tables 6-1 and 6-2, respectively.

To write a file, first set up the file name, disk number and side number, then call the write open subroutine (OPENW). Next, pass data a byte at a time to the disk file by loading each byte into the Accumulator and calling the I/O ROM OUTALL subroutine. When all data bytes have been output, call the write close subroutine (CLOSEW) to close the file. The CLOSEW subroutine automatically outputs the additional fill bytes (value=$00) to complete the last sector.

To read a file, first set up the file name, disk number and side number, then call the read open subroutine (OPENR). Next, read data from the disk file, a byte at a time, by calling the I/O ROM INALL subroutine and storing the read byte in the Accumulator into memory. When all data bytes have been received, call the read close file (CLOSER) to close the file.


```
PAGE 0001   ROUTINES TO READ AND WRITE FILES FOR AIM 65/40

ADDR OBJECT   SOURCE

                        ;
                        ; FDC ADDRESSES ( 5-18-82 )
                        DISK2=$885B
                        DISK1=$884A
                        TRKSK=$84DF
                        CLOPEN=$823A
                        FOPIN=$81A8
                        CLSOUT=$82B3
                        CLSIN=$8221
                        ; FDC VARIABLES
                        FINOUT=$522
                        DSTBUF=$D9
                        SRCBUF=$D7
                        BUFFER=$52F
                        ;
                        ; AIM 65/40 VARIABLES
                        OUTFLG=$274
                        INFLG=$273
                        ;
                              *=$4000
                        ;
                        ; TO OPEN A WRITE FILE , SET UP DISK,SIDE AND
                        ; NAME BEFORE CALLING THIS OPEN ROUTINE:
                        ;  DISK = $501 - NUMBER 0-7
                        ;  SIDE = $503 - 0 FOR FIRST SIDE
                        ;  FILENAME = $518 - 10 CHARACTERS PADDED WITH $20
                        ;  DENSITY = $505 - 0 FOR DOUBLE (OPTIONAL)
                        ;
4000 A2 01      OPENW   LDX #1              ;USE WRITE BUFFER
4002 8E 22 05           STX FINOUT
4005 20 5B 88           JSR DISK2           ;TURN ON WRITE DRIVE
4008 A9 00              LDA #0              ;SET UP DESTINATION BUFFER
400A 85 D9              STA DSTBUF
400C AC 2F 05           LDY BUFFER
400F C8                 INY                 ;WRITES USE BUFFER+1
4010 84 DA              STY DSTBUF+1
4012 20 DF 84           JSR TRKSK           ;RESTORE WITH DENSITY CHECK
4015 20 3A 82           JSR CLOPEN          ;OPEN THE FILE
4018 A9 10              LDA #$10            ;POINT OUTALL TO FLOPPY
401A 8D 74 02           STA OUTFLG
401D 60                 RTS
                        ;
                        ; TO WRITE DATA , CALL OUTALL WITH BYTE THE IN A
                        ; THE USER DRIVER WILL WRITE 128/256 BYTES AT A TIME
                        ;
                        ; TO CLOSE THE WRITE FILE
401E 4C B3 82   CLOSEW  JMP CLSOUT          ;CLOSE DISK FILE
```

Figure 6-2. Program Control File Handler

```
PAGE 0002   ROUTINES TO READ AND WRITE FILES FOR AIM 65/40

ADDR OBJECT   SOURCE

                ;
                ; TO OPEN A READ FILE , SET UP DISK,SIDE AND
                ; NAME BEFORE CALLING THIS OPEN ROUTINE:
                ;  DISK = $500 - DRIVE NO. 0-7
                ;  SIDE = $502 - 0 FOR FIRST SIDE
                ;  FILENAME = $50E - 10 CHARACTERS PADDED WITH $20
                ;  DENSITY = $504 - 0 FOR DOUBLE (OPTIONAL)
                ;
4021 A2 00      OPENR  LDX #0             ;USE READ BUFFER
4023 8E 22 05          STX FINOUT
4026 20 4A 88          JSR DISK1          ;TURN ON READ DRIVE
4029 A9 00             LDA #0             ;SET UP SOURCE BUFFER
402B 85 D7             STA SRCBUF
402D AC 2F 05          LDY BUFFER
4030 84 D8             STY SRCBUF+1
4032 20 DF 84          JSR TRKSK          ;RESTORE WITH DENSITY CHECK
4035 20 A8 81          JSR FOPIN          ;OPEN THE READ FILE
4038 60                RTS
                ;
                ; TO READ DATA , CALL INALL - BYTE RETURNED IN A
                ; CHECK FOR AN EOF CHARACTER IF NEEDED
                ;
                ; TO CLOSE THE READ FILE
4039 4C 21 82   CLOSER JMP CLSIN          ;CLOSE DISK READ FILE
                       END

ERRORS=0000
```

Figure 6-2. Program Control File Handler (Cont'd)

Table 6-1. Commonly Used AIM 65/40 DOS 1.0 Subroutines

Label	Address (Hex)	Function
CLOPEN	823C	Open the write disk file.
CLSIN	8223	Close the read disk file and turn off the drives.
CLSOUT	82BA	Close the write disk file and turn off the drives.
DISK1	884A	Set up and turn on the read drive.
DISK2	885B	Set up and turn on the write drive.
ERRLCL	84D2	Error handler for the AIM 65/40 DOS (includes FESC).
FESC	84DB	Turn off motors, reset name length and jump through FESCIV.
FOPIN	81AA	Open the read disk file.
ION	8447	Re-enable AIM 65/40 interrupts.
READIN	81ED	Read one byte into A (called through INALL).
TRKSK	84E6	Seek to the track in UCYL with verify and density checking.
WRTOUT	828F	Write out one byte from A (called through OUTALL).

Table 6-2. Commonly Used AIM 65/40 DOS 1.0 Variables

Label	Address (Hex)	Function
BUFFER	52F	MSB of pointer to DOS Buffer 1.
CURCMD	4AE	Last command executed by FDC device.
DSTBUF	0D9	Pointer to DOS Write Buffer.
ERSTAT	524	DOS error status.
ERRVEC	530	Pointer to the DOS error processing routine.
FBYTE	50A	Buffer 1 current byte (Read).
	50B	Buffer 2 current byte (Write).
FESCIV	532	Pointer to the DOS escape processing routine.
FDISK	500	Buffer 1 disk number (Read).
	501	Buffer 2 disk number (Write).
FSIDE	502	Buffer 1 side number (Read).
	503	Buffer 2 side number (Write).
FDEN	504	Buffer 1 density (Read).
	505	Buffer 2 density (Write).
FNAMES	50E-517	Buffer 1 file name (Read) - 10 characters.
	518-521	Buffer 2 file name (Write) - 10 characters.
FINOUT	522	Buffer indicator (0=Buffer 1, 1=Buffer 2).
LEN	52C	Number of remaining sectors (Read).
STFLG	4AD	FDC Device Status Register image.
SRCBUF	0D7	Pointer to DOS Read Buffer.
STORIO	55B	Disable (if Z=1) or re-enable all IRQ interrupt sources except for the FDC module.

If an error occurs during writing or reading using this handler, the IRQ interrupts are re-enabled, the disk drive motors are turned off, and control is returned to the AIM 65/40 Monitor command level. The source of the error can be determined by examining the command type (CURCMD), the FDC status register (STFLG), and the DOS error status (ERSTAT).

Note that the address of the error handler is placed in ERRVEC and the address to jump to after the error has been serviced is placed in FESCIV. If errors are to be handled under program control, these vectors must be loaded appropriately. The error handler must isolate and service the error as required. If the disk operation is to be terminated, the IRQ interrupts must be re-enabled (JSR ION) and the disk drive motors turned off (JMP FESC).

6.5.2 DOS Compatibility Considerations

The user is responsible for the meaning and coding of all data in files created under program control, including the definition and inclusion of control bytes. Sometimes the files are handled by knowing exactly how many bytes are written to a file, and therefore how many to expect upon reading it (which can be detected by logically ANDing LEN, LEN+1 and FBYTE then checking for a zero result. Any $00 pad bytes used to fill the last sector can be ignored.

Another approach is to include one or more special characters to indicate the end of a file. The end of file character(s) must not conflict with valid data values using their approach, however.

In order for a data file to be listed using the DOS list utility function (UTILITY=L, see Section 6.3.4), the data must be coded in ASCII and terminated with an ASCII End of File (EOF) character (byte value = $1A).

An example program using the handler shown in Figure 6-2 is listed in Figure 6-3. This example writes a series of ASCII characters to disk then reads the file. Since an $1A EOF terminator is used, the written file can be examined using the UTILITY=L DOS function. Constants are used for the file name, disk number and side number in this example.

6.6 DOS ERROR REPORTING

The DOS reports an error condition and value indicating the cause of the error whenever improper disk operation is detected. These errors may occur due to reasons such as improper disk drive operation, incorrect disk drive/FDC module connection, bad disk media, and operator error (e.g., wrong disk installed, wrong file name specified, wrong disk drive specified, attempted use of unformatted disk, etc.).

The error reporting format is:

```
DISK ERROR=
```

where:

XX = DOS detected errors plus seek error from the FDC device on the FDC module.

YY = The contents of the FDC device Status Register (less SEEK error) on the FDC module (see Section 4.5).

The error definitions are described in Table 6-3.

The error reporting linkage is through two vectors, ERRVEC (vector before error handling) and FESCIV (vector after error handling)--see Section 6.7. These vectors are initialized upon cold reset and may be altered to link to user-provided additional or alternative error handling.

Note that the default error handler return is to the Monitor (to ESCIN at $A32E) after reporting an error code.


```
PAGE 0001   TEST OF DISK FILE HANDLERS FOR AIM 65/40

ADDR OBJECT   SOURCE

                OUTALL=$F32B
                INALL=$F233
                OUTPUT=$F352
                OPENW=$4000
                OPENR=$4021
                CLOSEW=$401E
                CLOSER=$4039
                CHAR    =0
                      *=$3000
                ;
                ; THIS TEST PROGRAM WRITES OUT TWO SECTORS OF NUMBERS TO
                ; THE DISK . THE FILE IS THEN READ AND DISPLAYED , ALL
                ; UNDER PROGRAM CONTROL . THE FILE ( NAME IS FROM FNAME )
                ; IS SUITABLE TO BE LISTED FROM DOS .
                ;
                ; SET UP THE READ AND WRITE DISK PARAMETERS FIRST
                ;
3000 A9 00      TEST    LDA #0
                ; SET UP THE DRIVES
3002 8D 00 05           STA $500            ;READ DRIVE
3005 8D 01 05           STA $501            ;WRITE DRIVE
                ; SET UP THE SIDES
3008 8D 03 05           STA $503            ;READ SIDE
300B 8D 04 05           STA $504            ;WRITE SIDE
                ; DOWNLOAD FILENAME FROM FNAME
300E A2 0A      NAME    LDX #10
3010 BD 64 30   NAME1   LDA FNAME,X
3013 9D 0E 05           STA $50E,X          ;READ FILE NAME
3016 9D 18 05           STA $518,X          ;WRITE FILE NAME
3019 CA                 DEX
301A 10 F4              BPL NAME1
                ;
                ; NEXT WRITE OUT A TEST FILE
                ;
301C 20 00 40   TESTWR  JSR OPENW           ;OPEN FILE
301F A2 07      ISSUE   LDX #7
3021 A0 3C      NEWLIN  LDY #60
3023 A9 30      ZERO    LDA #'0
3025 85 00              STA CHAR
3027 A5 00      LOOP    LDA CHAR
3029 C9 3A              CMP #':
302B F0 F6              BEQ ZERO
302D 20 2B F3   WRITE   JSR OUTALL
3030 E6 00              INC CHAR
3032 88                 DEY
3033 D0 F2              BNE LOOP
3035 A9 0D              LDA #$0D            ;CR
3037 20 2B F3           JSR OUTALL
303A CA                 DEX
303B D0 E4              BNE NEWLIN
303D A9 1A      END     LDA #$1A            ;SHOULD BE EOF
303F 20 2B F3           JSR OUTALL
3042 20 1E 40   DONE    JSR CLOSEW
```

Figure 6-3. Example Program Using File Handler

```
PAGE 0002   TEST OF DISK FILE HANDLERS FOR AIM 65/40

ADDR OBJECT   SOURCE

                ;
                ; NOW READ IN THE TEST FILE
                ;
3045 20 21 40  TESTRD JSR OPENR          ;OPEN FILE
3048 20 33 F2  READ   JSR INALL          ;READ IN A CHARACTER
3048 C9 1A            CMP #$1A           ;EOF MARK
304D F0 0F            BEQ EOFOUT
304F C9 0D            CMP #$0D           ;ALWAYS SEND LF WITH CR
3051 D0 05            BNE OUTWIT
3053 20 52 F3         JSR OUTPUT
3056 A9 0A            LDA #$0A
3058 20 52 F3  OUTWIT JSR OUTPUT
305B 4C 48 30         JMP READ
305E 20 39 40  EOFOUT JSR CLOSER
3061 4C 14 A3         JMP $A314          ;COMIN1 ==>
                ;
                ; THE NAME OF THE FILE IS ...
3064 54 45      FNAME  .BYT 'TESTFILE01'
                .END

 ERRORS=0000
```

Figure 6-3. Example Program Using File Handler (Cont'd)

Table 6-3. AIM 65/40 DOS 1.0 Error Definitions

Code	Error	Definition
01YY	SEEK ERROR	The FDC device did not locate the data on the disk.
02YY	Undefined	
04YY	DISK FULL	The disk is full, thus, no more data may be written on it.
08YY	DIRECTORY FULL	The directory is full, thus, no more file names may be entered.
10YY	FILE NAME NOT FOUND	The specified file name is not in the directory.
20YY	FILE NAME EXISTS	The specified file name already exists for an active file.
80YY	DIRECTORY ERROR	The directory terminator was not found within the directory limits.
XX01	BUSY	The FDC device is busy.
XX02	Undefined	
XX04	LOST DATA	The CPU did not respond to DRQ in one byte time.
XX08	CRC ERROR	If RECORD NOT FOUND error exists, an error was found in one or more ID fields, otherwise a data field error exists.
XX10	RECORD NOT FOUND	The desired track, sector, or side were not found.
XX20	RECORD TYPE/WRITE FAULT	On read record, this bit indicates the record-type code from the data field address mark (1=Deleted Data Mark, 0=Data Mark). On a write, this bit indicates a write fault.
XX40	WRITE PROTECT ERROR	Writing was attempted to a write protected disk.
XX80	NOT READY	The drive is not ready.

6.7 DOS VARIABLES

The DOS variables, located on page zero and five are listed in Table 6-4. Some of these variables are for internal use only (to store temporary values) while others are user-alterable and constant after initialization. The default values for the user-alterable variables, initialized by a cold reset if the AIM 65/40 monitor is installed or the subroutine DOSSETT, are also shown in Table 6-4.

Table 6-4. AIM 65/40 DOS 1.0 Variables

Addr (Hex)	Label	No. Bytes	Init Value	Definition
ØD7	SRCBUF	2	-	DOS Source Buffer Vector
ØD9	DSTBUF	2	-	DOS Destination Buffer Vector
500	FDISK	1	ØØ	Buffer 1 Disk No.
501		1	ØØ	Buffer 2 Disk No.
502	FSIDE	1	ØØ	Buffer 1 Side No.
503		1	ØØ	Buffer 2 Side No.
504	FDEN	1	ØØ	Buffer 1 Density
505		1	ØØ	Buffer 2 Density
506	FTRACK	1	ØØ	Buffer 1 Track No.
507		1	ØØ	Buffer 2 Track No.
508	FSECTR	1	ØØ	Buffer 1 Sector No.
509		1	ØØ	Buffer 2 Sector No.
50A	FBYTE	1	ØØ	Buffer 1 Byte Count
50B		1	ØØ	Buffer 2 Byte Count
50C	SECLEN	1	ØØ	Buffer 1 Bytes/Sector
50D		1	ØØ	Buffer 2 Bytes/Sector
50E	FNAMES	10	ØØ -> ØØ	Buffer 1 File Name
518		10	ØØ -> ØØ	Buffer 2 File Name
522	FINOUT	1	ØØ	Buffer to use
523	CHAR	1	ØØ	Character Temporary Storage
524	ERSTAT	1	ØØ	Error Status
525	FILNM1	2	ØØ ØØ	Temporary Storage
527	FILNM2	2	ØØ ØØ	Temporary Storage
529	FILNM3	2	ØØ ØØ	Temporary Storage
52B	RETCNT	1	ØØ	Retry Count
52C	LEN	2	ØØ ØØ	Length Count for Read File
52E	INITFL	1	ØØ	Density Ask Flag*
52F	BUFFER	1	3E	Source Buffer Addr High Byte*
530	ERRVEC	2	CB 84	Before Error Handler Vector*
532	FESCIV	2	34 A3	After Error Handler Vector*
534	BEEPC	3	4C 67 F4	JMP BEEP*
537	SPACE	3	4C 7A F3	JMP BLANK*
53A	CRLFC	3	4C 5E AE	JMP CRLF*
53D	CRLOWC	3	4C 8F F3	JMP CRLOW*
540	OUTCL	3	4C 4E A5	JMP EBCRCL*

Table 6-4. AIM 65/40 DOS 1.0 Variables (Cont'd)

Addr (Hex)	Label	No. Bytes	Init Value	Definition
543	FNAMEC	3	4C 2A FA	JMP FNAME*
546	NOUTC	3	4C AC F3	JMP NOUT*
549	NUMAC	3	4C A4 F3	JMP NUMA*
54C	OUTALC	3	4C 2B F3	JMP OUTALL*
54F	OUTPUC	3	4C 52 F3	JMP OUTPUT*
552	RCHEK	3	4C 7D B0	JMP VARCHEK*
555	REDOC	3	4C 9B F2	JMP REDOUT*
558	WHERE	3	4C B9 AE	JMP WHEREO*
55B	STORIO	3	8D 80 FF	STA PRIRTY*
55E		1	60	RTS*
55F	MENUVC	2	62 80	Before UTILITY Decode Vector*
561	MENULS	2	85 80	After UTILITY Decode Vector*
563	NOVRFY	1	01	Format Disk No Verify Flag*

NOTES

*User-alterable

1. The DOS 1.0 variables are initialized by a cold reset (refer to the AIM 65/40 System User's Manual) only if the AIM 65/40 Monitor is installed.

APPENDIX A

DISK DRIVE CONFIGURATION

This appendix describes the actual configuration of some of the more popular disk drives. The drives included are:

Shugart SA400
Shugart SA450
Pertec FD200
Shugart SA800

A.1 5" DISK DRIVE CONFIGURATION

To operate the RM 65 FDC module in single- or double-density mode with these 5" drives, perform the following steps:

a. Set up switch S1 for common bank operation with both the Program ROM and I/O enabled.

S1-1 = OPEN
S1-2 = OPEN
S1-3 = CLOSED
S1-4 = CLOSED

b. Set up the jumpers for no DMA, no precompensation (which is not required for 5" double density) and the number of heads.

E1 = EITHER
E2 = EITHER
E3 = B for only single sided drives (SA400 or FD200)
A for double sided drives (SA450)

c. Set up the standard/mini-floppy headers (JB1, JB2) for mini-floppy drives (shown in Figure 2-2a).