UNITED TECHNOLOGIES MOSTEK

SOFTWARE PRODUCTS

ERRATA SHEET

ERRATA SHEET NUMBER: 4420308
DATE: MAY 1982
Publication Name: M/OS-80 OPERATIONS MANUAL
Publication Number: 4420064
Publication Date: FEB 1982
Prior Errata Bulletins: None

SUMMARY OF CHANGES:

REPLACEMENT PAGES:

Attached are a set of replacement pages for the M/OS-80 Operations Manual. These pages will correct errors in the existing pages. Please remove the old pages, replace them with the new pages, and discard the old pages. The replacement pages are easily identifiable by the date at the bottom of the page.

Attached also is a new title page, reflecting a product designator number change. The old number was MK77984-4. The new number is MK71010 for Version 3 (M/OS-80 V3).

BUG NOTICES:

Attached is a list of known bugs and other problems in V3.01. These problems will be fixed in the next revision of M/OS-80 V3.

The following bugs have been noted in M/OS-80 V3.01. They will be fixed in a future revision.

M/OS-80:

System call 0CH deselects disk and clears file buffers. May cause some CP/M programs to fail.

M/OS-80 system calls do not return registers A=L and B=H, as some CP/Ms do. This may cause some CP/M programs to fail. The CP/M manual cautions against using this "feature", which was retained for compatibility with PL/M.

Disk write errors cause track 0 (the boot track) to be overwritten in systems using the DDT/DCF firmware.

If, in response to the "OPI not found - enter disk -" prompt, the operator enters an invalid drive descriptor, the system hangs up, and will require a reboot.

System call 01H does not echo control characters.

System call 96H does not turn off the motor on 5.25" drives.

System calls to the RDR: and PUN: drivers actually call the console driver, unless MOSGEN has been used to install another driver.

The M/OS-80 file name parser does not allow "/" inside a file name.

UTILITIES:

FORMAT: always selects drive A: when exiting.

FORMAT: allows attempts to format a single-sided drive as double-sided.

FORMAT: will not reformat a double-sided double-density 5.25" diskette to single-sided single-density unless the diskette has been bulk erased.

DSKDUMP: directly to disk block does not always properly flush the disk buffer when Quitting immediately after a write. Simple solution is to do 2 or 3 read before Quitting. (DSKDUMP works properly with files).

M /OS-80
FLEXIBLE DISK OPERATING SYSTEM
OPERATIONS MANUAL
V3 MK71010

Revised May 82

PREFACE

This manual describes M/OS-80, the Mostek Z80 disk operating system. M/OS-80 is Mostek's answer to the growing need for a general-purpose operating system that can take advantage of the wide variety of pre-written applications packages available for microcomputer systems. Although newly enhanced and updated, M/OS-80 has several years of field experience and has proven itself in a number of diverse areas.

Many of the concepts and techniques used may be familiar to the experienced user. However, expertise is not required to use this system. Using M/OS-80 does require some simple understanding of running computer programs, and how files are referenced: i.e., what a file looks like, how to erase old files, and how to keep proper duplicate copies (backups) of program and data files.

This manual consists of four main sections and several appendices:

1. System startup, Conventions, and Operations
2. Console Commands and Utility Summary
3. Internal Operations
4. System Calls

A. DDT Debug firmware (V3)
B. System Setup - V3

The first section contains that basic information which is needed to run the system. Discussions of basic concepts, system operation, disk handling, operator instructions, file conventions, and command syntax are found here. In addition, there is an explanation of system errors and recovery procedures.

The second section is an alphabetical reference to the commands available to the operator. Starting with a table of contents and keyword index, it is intended to be referenced only for commands the operator actually needs, although it pays to skim this section to be aware of what is available.

The third section is for the programmer, especially the machine-level programmer. Here, the individual system calls that are used by all applications programs are listed and explained. Included in this section is a chart that describes the inter-system compatibility of M/OS-80 and various other operating systems.

The fourth section gives detailed descriptions of the M/OS-80 system calls. It describes in detail which registers are used and which need to be set up before calling the system.

Appendix A describes the DDT (Designer's Development Tool) debug firmware used with M/OS-80 V3. DDT is used to diagnose hardware, software, and general systems problems in addition to providing a means of developing software for the system.

Appendix B is an outline and procedural manual that can be used to configure a M/OS-80 V3 system. Included in this section are descriptions of the various part numbers, version numbers, and alternate configurations available.

1.9 SYSTEM ERRORS

If the disk I/O handler intercepts an error while attempting to access a disk, the error message (as shown below) is sent to the operator via the console. No message is sent until the system has made at least five automatic retries.

In versions using the condensed error messages:

```
Disk error nn Drive d:, bbbbbb, ss
Enter R, I, E, or ^C: *
```

In versions using the expanded error messages:

```
Disk error nn, Drive d: Block bbbbbb, Status ss
---------> Explanatory message   <-----------
Entry R(etry), I(gnore), E(rror), ^C(ancel) *
```

In the above messages:

nn	Disk error code number (i.e., 81).
d:	Drive identifier for disk where error occurred.
bbbbbb	Logical block attempted in I/O operation.
ss	Device status code.

In either case, the cursor is left at the "*" and the operator is expected to respond with one of the valid operator responses as shown below:

^C — (Control C) M/OS-80 returns control to the OPI command processor, thus ending the current active task or application program. This response is used in situations where there can be no recovery made for the type of error which has occurred.

R — Retry disk operation. This response is made for those operations which are operator-correctable and have been remedied prior to entering "R".

I — M/OS-80 ignores the error and returns control to the task or application program making the disk I/O request as if the operation were successful.

E — M/OS-80 passes an error code (in register A) back to the task or application making the disk I/O request. It is the responsibility of the application to deal with the error.

Certain errors have standard messages as described in the following table.

```
81 - Disk not ready.
82 - Disk will not home.
83 - Seek position.
84 - Read error.
85 - Write error.
86 - Read-after-write Error.
87 - Illegal block number.      *
88 - Illegal command.           *
89 - Disk is write-protected.   **
```

Fig. 1-4: Standard Disk Error Messages

Errors 81 through 86 may be remedied by the operator once corrective action has been performed. For example, after closing a diskette drive door or putting the disk in right-side up, the operator may type R for RETRY so that the system can attempt the disk I/O operation again.

* These errors are due to a non-changing situation. Retry will make no difference, especially since this can only occur due to program or system error.

** This error is due either to a true hardware write-protect tab, or changing a disk at an improper time. If the hardware tab is the problem, it may be replaced and Retry attempted. Otherwise, only ^C (Control C) will be effective.

The M/OS-80 Floppy Driver for the DCF Prom returns a status code which is shown in the disk error message as described above. The following bit table is provided to decode that status byte .

BIT	Error if One (set)
7	Invalid drive, track or sector.
6	Disk unit not ready.
5	Track seek error.
4	Sector not found.
3	CRC error.
2	Data lost.
1	Disk is write-protected.
0	Attempt to read a deleted sector.

Fig. 1-5: Disk Error Bit Decode Table

1.11. HARDWARE CONFIGURATIONS

M/OS-80 can be run on a variety of MOSTEK hardware configurations. The following table is a summary of those configurations:

	V3	V5	V6
CPU	MDX-CPU1(A) MDX-CPU2(A) SDB-80(E)	MDX-CPU3 MDX-CPU4 SDE-COMBO	MDX-CPU3 MDX-CPU4 SDE-COMBO
MEMORY	MDX-DRAM32Ax2 RAM-80(E)	(on cpu)	(on cpu)
CONSOLE	MDX-EPROM/UART MDX-SIO(2) UART on SDB	MDX-EPROM/UART MDX-SIO(2) STI on CPU3/4 SIO on COMBO	MDX-EPROM/UART MDX-SIO(2) STI on CPU3/4 SIO on COMBO
PRINTER	MDX-PIO MDX-SIO(2) PIO on SDB	MDX-PIO MDX-SIO(2) on CPU3/4 PIO on COMBO	MDX-PIO MDX-SIO(2) on CPU3/4 PIO on COMBO
FLOPPY DISK	MDX-FLP1 MDX-FLP2 FLP-80(E)	MDX-FLP2 on COMBO	MDX-FLP2 on COMBO
HARD DISK			MDX-SASI1 MDX-SASI2
FIRMWARE	DDT/DCF	PHANTOM	PHANTOM

Fig. 1-6: Required Hardware Configurations

Specific set-up details for the various configurations are described in the appendix.

If the user desires to use combinations of hardware and software drivers not listed above, the use of the MOSGEN package is recommended. This package permits the reconfiguration of M/OS-80 to accomodate user-written drivers and non-standard hardware. Refer to the MOSGEN Operations Manual for further details.

2. CONSOLE COMMANDS AND UTILITIES

This section describes each of the built-in commands and program utilities which are provided in the M/OS-80 package.

2.1. SUMMARY OF BUILT-IN COMMANDS

COPY	Disk-to-disk copy
DISKRD	Disk read-only test
DSKDUMP	Absolute disk dump
DUMP	File dump
EDIT	File editor
ERASE	Conditional file erase
FORMAT	Initialize a diskette
LABEL	Set or alter disk labels
LOAD	Load Intel Hex or .COM files
PPG	Prom Programming utility
PRINT	Printer file list
SORTDIR	Prepare sorted directory file
SPLIT	Split large text files into smaller pieces
SPOOL	Start print spool task
STRIP	Strip parity, nulls & rubouts from text file
SXFER	Single disk file transfer
WRTSYS	Write system boot to disk
XDIR	Extended directory
XFER	File copy or concatenate
XFLP	Move ASCII files from FLP80-DOS diskettes
XSTAT	Directory sizing & validator

Fig. 2-1: Utilities (.COM files)

COPY	[<destination>:=<source>:]
DISKRD	[d:]
DSKDUMP	<file>
DUMP	[<file>]
EDIT	[<file>]
ERASE	<gname>
FORMAT	
LABEL	[<disk:<command>=<value>]
LOAD	<file>
PPG	<file>
PRINT	<file> [/<options>] [<title>]
SORTDIR	<outfile>=<gname>
SPLIT	<file> [<max size (IN k)>]
SPOOL	<file>
STRIP	<file>
SXFER	<gname>
WRTSYS	d:[file]=d:[file]
XDIR	[d:]
XFER	[[/<opts>] <file>=<file>[,<file>,<file>...]]
XFLP	<FLP80-DOS filename>
XSTAT	[d:]

Fig. 2-2: Command Syntax Summary

2.14. EXEC - EXECUTE AT ADDRESS

SYNTAX:

EXEC address

-Where address is a valid hex address (0 - FFFFH)

DESCRIPTION:

EXEC allows user to directly execute any valid hex address directly from M/OS-80. EXEC is especially useful to execute a program, (which the user has possibly altered) without re-loading the file from disk. EXEC could also be used to execute a user program which did not begin execution or reside at the TPA address of 100H. EXEC is used in the PROM-based system to enter Mostek's DDT PROM debugger program by typing **EXEC E11D**.

Example:

EXEC 100

User executes the program residing at 100H.

EXEC E11D

User executes the DDT program (PROM based system ONLY).

FILES:

EXEC is built-in to the operating system.

WARNINGS:

Since EXEC DIRECTLY executes the address given it does NOT scan the command line or set any buffers. It does NOT have the same function as the implied running of a .COM file from the operating system.

2.15. FORMAT - INITIALIZE M/OS-80 DISKS

SYNTAX: FORMAT

DESCRIPTION:

FORMAT is used to write sector ID marks and clear all existing data on floppy or hard disks. It should be used on all new disk to ensure that all directory areas are cleared and that all areas can be read later.

> Note: Be sure to cover the floppy diskette write-protect notch (if present) before attempting to initialize any floppy diskette.

Once the program has started the following dialog takes place:

```
Mostek MDX Formator (FORMAT 00.xx)
Drive to be formatted (A...x) [^C to exit]?
```

The operator then types a letter (A thru x, where x is the last drive in the system) to indicate which drive will contain the disk to be formatted.

If the drive selected is a hard disk, the system responds:

```
To format hard disk in drive (n) unit (y)
The disk must use a SASI controller
--Press F to Format, Press ^C to Exit--
```

The program then asks:

```
Enter section interleave (in decimal)
```

The operator should respond by typing the desired interleave value, or bit carriage return to select a default interleave value of zero. The formatting begins and operator prompt is output.

```
Formatting
```

This concludes hard disk formatting.

If the drive selected is a floppy disk drive, the system responds:

```
Format disk for (S)ingle or (D)ouble density?
```

The operator types S for single density (IBM 3740) format or D for double density (MFM) format.

```
To format disk in drive n
Load disk and
        --Press F to format, Press ^C to exit--
```

If the controller is set up for 5.25" drives, the following question is asked:

Enter tracks/side (in decimal) >

If the operator selects double density and the controller does not support double density, the following error message is printed on the console:

*** Cannot Format double density with this controller.

The program then asks:

Enter sector interleave (in decimal) >

The operator should respond by hitting carriage return if the disk is to be used on foreign (i.e. non MOSTEK) system. If the disk is to be used exclusively on MOSTEK supplied hardware, for optimum throughput we suggest a sector interleave of 3 for 8" single density, and 5 for 8" double density. For 5.25" disks, the default is designed for optimum throughout.

The system then types a message similar to the following:

Formatting Single-sided Single-density
8" disk in drive n, port xx, unit y.

Where n is the drive name, xx is the floppy controller board's address and y is the unit address of the controller for the drive.

Once the system has identified the drive as to density and number of tracks and sides, the following dialog takes place:

Whole (w) or Partial (p)?

At this point the system attempts to determine if the entire diskette is to be formatted or just a part. To format a segment of the disk, type p.

If w is typed, the system proceeds to format the disk. If p is typed, the following message is displayed:

Enter Starting track (in decimal) >

At this time the operator is requested to enter the track location to start formatting the diskette.

Once the Starting Track has been established, the system displays the following message:

[tracks left=nn], press return for all
Enter Number of tracks to format (in decimal) >

At this time the operator is requested to enter the number of tracks to format from the point specified in the starting track prompt. If the operator hits return at this point all remaining tracks from the starting track to the end of the disk will be formatted.

Example: If the operator decides to format track 0 and 1 on a disk, the value for starting track would be 0 and the value for number of tracks to format would be 2.

During the formatting process, the track being formatted is printed on the console device as follows:

```
(000)123456789 (010)123456789 (020)123456789....
```

Once all tracks have been formatted on either hard or floppy disks, the following message is printed on the console:

```
--Diskette completed--

Drive to be formatted (A..x) [^C to exit]?
```

If desired, the process can be repeated on the same or another disk. To repeat, enter drive number. To end the program and return to the operating system, hit control-C.

FILES:

FORMAT.COM

3.1 SYSTEM ORGANIZATION

M/OS-80 is a RAM/Disk-resident operating system that is loaded from disk when a cold boot sequence is initiated. During system configuration, the memory location of M/OS-80 is fixed, and the bootstrap loader is automatically setup to load M/OS-80 properly.

The system resides in high memory above user programs. By defined use of low memory (0-100H) all user programs call M/OS-80 through a standard sequence which is transparent to actual memory size. Additional memory simply expands the available user area.

The system is broken down as follows:

RAM Memory areas:

Top -	IOSYS	- Device Driver System.
	DOS	- Disk Operating System.
Variable -	OPI	- Operator Interface Linkage (If RESIDENT)
100H -	TPA	- Transient Program Area.
0-FFH	Low Memory	- Reserved Space.

Allocation of Disk Areas:

Track 0-1	System Area
Tracks 2-n	Disk File Directory
Remaining:	User File Region

These system segments work together to handle all standard disk and character-device I/O for the user. The functions are described in detail later in this manual.

3.2. MEMORY STRUCTURE

3.2.1. IOSYS

The Input/Output System performs the various primitive I/O functions for the character-devices (console, printer, punch, reader) and for the disk devices. A program which does all of its own file management functions or does not use the disk might need only IOSYS.

3.2.2. DOS

The Disk Operating Subsystem performs all of the disk-oriented features of M/OS-80 including managing disk files (creating, opening, reading, and writing). In addition, M/OS-80 DOS is responsible for calling user programs, editing console input, allowing user control over I/O, and many other functions.

3.2.3. OPI

The Operator Interface is the set of routines that the operating system uses to communicate with the user. This sub-system permits the user to specify which of several system functions to perform. To execute one of these system functions, OPI invokes a user or system program based on the command line and passes parameters from the command onto the program. Those commands which invoke system-level RAM-resident programs are referred to as built-in commands. If a command is not recognized as being one of the built-in system functions, M/OS-80 tries to find and execute a disk-resident .COM module corresponding to the command. Any programs invoked are loaded into the user area as described in the "User Area" section below. OPI can be totally RAM-resident or may be reloaded between user programs, see the OPIRES and OPINRES commands. If OPI is non-resident, a small OPI linkage table is left resident in RAM.

3.2.4. Transient Program Area (TPA)

The RAM location where user programs are loaded and executed is referred to as the TPA (Transient Program Area) or user area. The TPA starts at 100(hex) and can extend as high as the bottom of the Operator Interface (OPI). If the OPI is non-RAM-resident, user programs can extend up to the bottom of the DOS (giving an additional 3K). All commands not built-into the DOS and user command programs are loaded and executed here. Built-in commands generally do not modify this area. **If OPI is non-resident, it will always use the TPA. Bootstrapping also uses the lower 3K of the user area.**

3.5 COMMAND STRUCTURE

A M/OS-80 command is a program that is designed to be loaded starting at 100H and stored on the disk as a memory image in a file with type .COM. Program (.COM) files may be either a MOSTEK-supplied system utility or a user program. Any filename is legal as a command; however, built-in names will not be recognized as user programs because the operator interface predefines them. Examples are: ATRIB, DIR, ERA, REN, SAVE, TYPE, etc.

The operator interface will pass the user command line to a command program. The invoked program need not use the command line or M/OS-80 itself, although most will use both.

For ease of programming, the first two arguments in the command line will be pre-formatted into standard file control blocks (FCBs) at 5CH and 6CH by the console processor. The rest of the command line, after the command name, will be placed at 80H.

Standard format for the command line is:

```
                          (Arguments)
        A.<Program name> [/][<operand>[/]<operand>]
       /          ^                      \
      /                                    \
M/OS-80      .COM filename          User (system) program
prompt                              parameters
```

For example:

To use XFER to copy all files from drive A to drive B:

```
A.XFER /V B:=A:*.*
```

The /V is seen as the operand "V" and the B:=A:*.* is interpreted as another operand. The user program itself decodes that portion of the command line.

If a name which does not match a .COM file on the current or library disk is used as the program name, a "Program Not Found" message is printed.

3.6. PROGRAMMER FACILITIES

3.6.1 System Calling Conventions

M/OS-80 reserves low memory (0-0FFH) for internal uses. M/OS-80 resides in high memory with OPI and BIOS. The user is allocated all memory from 100H to below M/OS-80 for unspecified uses.

A program is always loaded and starts execution at 100H. Once started, it can do whatever it wants; however, if it destroys M/OS-80 or BIOS it will have no way to use the disk or recover the operating system without the operator intervening.

M/OS-80 is entered through two special locations in low memory:

JMP 0 is used for returning to OPI and the operator or the next step in a batch.

CALL 5 is the normal system request call.

Standard conventions, when M/OS-80 is called thru CALL 5, are discussed in section 4. (**See Function Calls.**)

The region from 0-100H has the following data of specific interest to a program:

0..2	System return entry.
3	IOBYTE for device re-assignment.
5..7	System call entry.
6..7	Pointer to top of user area.
08H..4FH	Reserved for interrupt vectors.
30H-32H	Reserved for system.
38H-3AH	Illegal address trap.
40H-5BH	Reserved for system.
5CH..6BH	User FCB 1.
6CH..7BH	User FCB 2.
80H..0FFH	User buffer.

SYSTEM CALL SUMMARY - NUMERICAL (CP/M)

Code Hex	Code Dec	Function	Parmeters (in -> out)
17H	23	Rename File	DE=FCB(old,new) -> A=-1 or File num
18H	24	Login Vector	->A=bit vector (disk A=lsb)
19H	25	Current Disk	-> A=disk number
1AH	26	Set Buffer Address	DE=New buffer address
1BH	27	Allocation Vector	->BC=Address of map,DE = clusters, A=sectors/cluster

4.2. SYSTEM CALL SUMMARY - NUMERICAL

The following calls are not supported by CP/M 1.4.

Code Hex	Code Dec	Function	Parameters (in -> out)
1CH	28	Write Protect Disk	
1DH	29	Get W/P Vector	-> A = W/P vector
1EH	30	Set File Name Attrib	DE=FCB w/ atrib set
21H	33	Random Read	DE=random FCB-> A=0 ok, or 1,3,4,6
22H	34	Random Write	DE=random FCB-> A=0 ok, or 1,3,4,5,6
23H	35	Determine File Size	DE=random FCB-> A=-1 or File num
24H	36	Determine Random Record	DE=random FCB
25H	37	Reset Drive	DE=Drive(s) to be reset-> A=0 ok
28H	40	Write Random with Fill	DE=random FCB -> A=0 ok
80H	128	Read Console (no echo)	-> A=char
81H	129	Get User Register	BC=^URREG,SW,SDVT, SDT,SYSBOT
82H	130	Set User Ctrl-C	DE=abort,0-normal, -1=disable
83H	131	Read Logical Block	DE=blk, b=disk (msb if IL)
84H	132	Write Logical Block	DE=blk, b=disk (msb if IL)
85H	133	Spool Control	DE=Spooler FCB
86H	134	Format Name to FCB	DE=FCB, HL=string
88H	136	Chain to Program	DE=FCB
89H	137	Multiply	DE*HL -> DE
8AH	138	Divide	(HL/DE)->HL,BC (HL mod DE)->DE
8BH	139	Home Driver	B=disk
8CH	140	Eject Diskette	E=disk

SYSTEM CALL SUMMARY - NUMERICAL (cont.)

Code Hex	Code Dec	Function	Parameters (in -> out)
8DH	141	Get Version,Ser.no.	-> B=major rev./ C=minor rev. H=sub -Minor DE=Serial Number
8EH	142	Set CRT Function	D=col,E=row or D=spec,E=0
8FH	143	Set Date	B=day, D=mon, E=yr-1900
90H	144	Read Date	-> A=day, B=mon, C=yr-1900
91H	145	Set Time	B=sec, D=min, E=hr (24)
92H	146	Read Time	-> A=sec, B=min, C=hr (24)
93H	147	Set Program Return Code	A =code
94H	148	Set File Attributes	DE=fcb,B=atrib (PWRU/Sxx*)
95H	149	Read Disk Label	DE=fcb
96H	150	Turn Motors Off	
97H	151	Set System Bottom	E=Hi byte of address
98H	152	Read W/O Advance	DE=^FCB -> A=0 ok, or 1,2
99H	153	Write W/O Advance	DE=^FCB -> A=0 ok, or 1,2,-1
9AH	154	Test Block Allocated	
9CH	156	Directory Listing	DE=FCB to match
9DH	157	Set Console Options	E=Options byte
9FH	159	Get Master Disk	-> A=master disk
A0H	160	Mount Disk	E=disk
A1H	161	Dis-mount Disk	E=disk

4.3 SYSTEM CALL SUMMARY - CATEGORIES

4.3.1. General System Functions

HEX	DECIMAL	FUNCTIONAL
0	0	Return to System
81H	129	Get user register
89H	137	Multiply
8AH	138	Divide
8DH	141	Get Version number
97H	151	Set System Bottom
8		

4.3.2. Console I/O Functions

1	1	Get Console (and echo)
6	6	Get/ Put Console (raw/ no echo)
0AH	10	Input Buffered Line from Console
0BH	11	Test Console Character Ready
80H	128	Get Console (no echo)
2	2	Put Console
9	9	Print Buffer to Console
82H	130	Set User Control-C Handling
8EH	142	Set Cursor or Terminal Function
9DH	157	Set Console Options
3	3	Get Reader
4	4	Put Punch
5	5	Put List
7	7	Get I/O byte
8	8	Set I/O byte

4.3.3. Date and Time

8FH	143	Set Date
90H	144	Read Date
91H	145	Set Time
92H	146	Read Time

4.3.4. Disk Functions

1AH	26	Set Disk Buffer
25H	37	Reset Drive(s)
88H	136	Chain to Program
86H	134	Format FCB from String
93H	147	Set Program Return Code

4.3.5. Disk Select

0CH	12	Deselect Current Disk
0DH	13	Reset DOS Select A Drive
0EH	14	Select Disk
1CH	28	Write Protect Disk
1DH	29	Get Write/Protect Vector
8CH	140	Eject Disk
96H	150	Turn Motors Off
A0H	160	Mount Disk
A1H	161	Dis-mount Disk

4.3.6. Directory Maintenance

11H	17	Search Directory
12H	18	Search Next Entry
17H	23	Rename Files
1EH	30	Set File Name Attributes
94H	148	Set File Attributes
9CH	156	Directory Listing

4.3.7. Disk Status Functions

18H	24	Disk Login Vector
19H	25	Current Disk
1BH	27	Disk Cluster Allocation Map
95H	149	Read Disk Label
9FH	159	Get Master Disk

4.3.8. Files

0FH	15	Open File
10H	16	Close File
13H	19	Delete Files
16H	22	Create File
28H	40	Write Random with Zero Fill

4.3.9. File Data Functions

14H	20	Read File Block
15H	21	Write File Block
21H	33	Random Read
22H	34	Random Write
23H	35	Determine File Size
24H	36	Determine Random Record
98H	152	Read Without Advance
99H	153	Write Without Advance
9AH	154	Test Current Record Allocated

4.3.10. Direct Disk Access

8BH	139	Home Disk Drive
83H	131	Read Logical Block
84H	132	Write Logical Block

4.4. FUNCTION CALLS

M/OS-80 System Calls are designed to permit the systems-level programmer to directly interface with the various primitive functions of the operating system. These calls permit direct interaction with the various hardware devices attached to the system. To use them, a thorough understanding of the operating system and the hardware devices is usually necessary, as well as a fairly high level of understanding of assembly language.

All calls require that the system function code be loaded into system register C. The application program requesting the operating system service then makes a CALL to location 5. The operating system will then perform the function(s) as requested and return (if necessary) to the application. The application will not receive control again until the operating system has completed the requested function.

4.4.1. Function Details

Additional parameters passed to and from the operating system are passed in the other registers. Details of that protocol are shown below:

Format

Title:	Function code in hex and decimal. Function Description.
Entry parameters:	What registers are passed to function.
Return Parameters:	What registers or information are passed from function.
Example:	Example shows function use.

For purposes of this manual, the Z80 register set is noted by enclosing the register or register pair in braces [] to indicate that the register(s) specified contain the passed information. Registers enclosed in parenthesis indicate that the registers indicated contain the address of the passed information.

For example, [HL] would indicate the register pair HL contains the parameter while (HL) would indicate that the register pair HL contains the address of the parameter.

4.4.16 OEH/14 - Select Current Disk.

Select current disk, which is used for disk operations, when no specific disk is specified. No disk activity takes place until the disk is actually accessed. If a disk higher than those in the system is requested, then no change in the current disk occurs.

Entry Parameters: [E] has disk number (0-A,1-B,...)

Return Parameters: None.

Example:

```
LD  E,1         ;Select Drive B
LD  C,14
CALL 5
```

4.4.17. OFH/15 - Open File.

The file described by the File Control Block (FCB) passed in DE is opened for access. The file must be setup as described in section 3. (See **Creating a File Control Block**). A file must be either opened or created before it may be accessed.

Entry Parameters: (DE) has address of FCB of existing file.

Return Parameters: If no file found than [A] has OFFH (255); else [A] has directory block number of file found (0..3). There is one directory block number (starting at 0) for every four directory entries.

4.4.18. 10H/16 - Close File.

The file described by the File Control Block (FCB) pointed to by the DE register pair is closed and the disk directory updated. If the file cannot be found, an error code is returned. Files must be closed so that RAM FCB information can be posted to the disk directory.

Entry Parameters: (DE) has address of FCB of existing opened file.

Return Parameters: [A] has - 1 (255) if no file found or [A] has file number of file found (0..3). See OPEN (above) for explanation of file numbers.

4.4.19. 11H/17 - Search Directory

The disk directory is searched for the first occurrence of the file name specified in the FCB addressed by the register pair DE.

The FCB passed can have match characters so that the SEARCH/SEARCH NEXT call can return successive files with similar names. The match character "?" (3FH) can be in the disk file attribute byte (FCB+0), the name (+1..+8), the file type (+9..+11), or the extent byte (+12). The name and type matching is simple ASCII string matching. If disk byte contains "?", then all ASCII characters will match regardless of letter. This will return a set of files with ambiguous names. If the extent byte contains "?", then all extents will be returned.

NOTE: This call and the next call (Find Next Directory) will return directory entries whether they have been erased or not. Files are marked as being erased by replacing the attribute byte (FCB+0) with an 0E5H.

Entry Parameters: (DE) has FCB with search argument as described above.

Return Parameters: [A] has -1 (255) if no file found, or [A] has the file number of file found (0..3). See OPEN for discussion of file numbers.

4.4.25 17H/23 - Rename files.

Renames all files whose names match the FCB given. The FCB may contain match characters, in which case search is done first to test for possible duplicate name creation.

Entry Parameters: [DE] has address of FCB used to change all occurrances of the file named in the first 16 bytes to the file name in the second 16 bytes.

Return Parameters: [A] has file number of last file deleted, or -1 (255) if no files found.

4.4.26 18H/24 - Disk Login Vector.

The call returns a bit vector of disks currently active. Each bit indicates a logged-in disk with least significant bit (LSB) indicating disk A.

Entry Parameters: None.

Return Parameters: [HL] has bits for active drives.

The Format For Disks are as follows:

```
L:  /HGFE/DCBA
H:  /xONM/LKJI
```

x Represents non-existent drive

NOTE: To maintain compatibility with early versions, registers A and L contain the same values.

4.4.27 19H/25 - Get Current Disk.

The call returns the number of current disk, A=0, B=1,... etc.

Entry Parameters: None.

Exit Parameters: [A] has current disk.

4.4.28. 1AH/26 - Set Disk Buffer.

Set the RAM memory address of the buffer to be used for disk data operations and for directory search calls. The initial location of this buffer is 080H.

Entry Parameters: (DE) has address of disk buffer.

Return Parameters: None.

4.4.29. 1BH/27 - Disk Cluster Allocation Map.

Returns the starting address of the disk allocation map (bit map). Allocation maps are for system program use. For big directories, the argument returned contains the buffer pointer for the disk bit map. Bit maps return size information about the disk format that helps in displaying a disk directory.

Entry Parameters: None.

Return Parameters: (BC) returns address of RAM bit map (1 bit per cluster). [DE] returns number of clusters. [A] returns number of sectors per cluster. (HL) also returns address of RAM bit map.

4.4.30. 1CH/28 - Write-Protect Disk.

Sets the current disk to logical write-protected. Allows a program to totally inhibit write access to a disk. Holds until the disk is logged in again.

Entry Parameters: None.

Return Parameters: None.

4.4.31. 1DH/29 - Get Write-Protect Vector.

Returns the write-protect bit-vector. The bit order is the same as function 24. The vector bits are set either by the system directory mechanism when it finds the disk has been changed without a proper login, or by a program using function 28.

The vector format for disks are as follows:
L: /HGFE/DCBA/
H: /xONM/LKJI/

Return Parameters: [HL] has write-protect bits.

4.4.32. 1EH/30 - Set File Attributes.

Allows an alternate file attribute function to system function 94H. The high bits of the first 2 bytes of the file type (FCB+9,+10) are used to set/reset or the write-protect attribute and system attributes of the files whose name matches the FCB given.

Entry Parameters: (DE) has address of FCB with requested attribute set.

Return Parameters: [A] returns file number of files found or -1 (255) if no files found. See OPEN for discussion of file numbers.

4.4.33. 21H/33 - Random Read.

Provides a means for random file access without any additional file processing. A random FCB is required. (See section 3.2.3) The random record pointer in the FCB (FCB+33, +34,+35) is set to the record to be read and this function is called. No change is made to the record pointer, however, next record pointer and extent may be changed as required to position within the file.

For sequential reading, the random record pointer must be incremented by the user. To mix random and sequential reading and writing, function 36/24H should be used after sequential accesses to reset the random record pointer for future random reads.

Entry Parameters: (DE) has address of opened random FCB set to record to be read.

Return Parameters: [A] has:

- 0 - Read OK.
- 1 - Reading unwritten data.
- 3 - Cannot close current extent.
- 4 - Seek to unwritten extent.

4.4.34. 22H/34 - Random Write.

Provides a means for random file access without any additional file processing. A random FCB is required. (See section 3.8.12). The random record pointer in the FCB (FCB+33,+34,+35) is set to the record to be written and this function is called. No change is made to the record pointer; however, the next record pointer and extent may be changed as required to position within the file.

For sequential writing, the random record pointer must be incremented by the user. To mix random and sequential reading and writing, function 36H should be used after sequential accesses to reset the random record pointer for future random reads.

Entry Parameters: (DE) has address of opened random FCB set to record to be written.

Return Parameters: [A] returns:

0 - Write OK.
3 - Cannot close current extent.
4 - Seek to unwritten extent.
5 - Directory full.
6 - Disk full.

4.4.35 23H/35 - Determine File Size.

Provides a means of appending to a file. A random FCB is required. The random record pointer in the FCB (FCB+33, +34, +35) is set to the record past the last record in the file. Writing may proceed from this point by just writing normally.

For text files, the last previous record should be read and writing started at the logical end-of-file marker (^-Z).

Entry Parameters: (DE) has address of random FCB for an open file.

Return Parameters: FCB random-record pointer set.

4.4.36 24H/36 - Determine Random Record.

Provides a means for mixing sequential and random file access. The record pointer is set based on the next record pointer and extent. A random FCB is required.

To mix random and sequential reading and writing, this function should be used after sequential accesses to reset the random record pointer for random accesses.

The function is also useful while reading or writing a file sequentially to determine the random record pointer for use in building an index file.

Entry Parameters: (DE) has address of random FCB for an open file.

Return Parameters: FCB random-record pointer set.

4.4.36.1 25H/37 - Reset Drive(s).

This function instructs M/OS-80 to logically "log-off" the drive(s) specified. No disk access a ^C (control C) for specific drive(s). Bit pattern for drive(s) to be effected is placed in register pair DE as follows:

```
D: /HGFE/DCBA/
E: /xONM/LKJI/
```

Entry Parameters: [DE] has vector as shown above. 1 = drive to be reset, 0 = drive status to remain the same.

Return Parameters: [A] returns 0 if successful.

4.4.36.2 28H/40 - Write Random with Zero Fill.

Performs random write as discussed in call 24H/34 but pre-fills any unallocated block with zeroes prior to the data being written thus making sure all bytes of the block are pre-set with a known value.

Entry Parameters: (DE) has address of opened random FCB set to the record to be written.

Return Parameters: [A] returns:
- 0 - Write operation OK.
- 3 - Cannot close current extent
- 4 - Seek to unwritten extent
- 5 - Directory full
- 6 - Disk full

4.4.42. 85H/133 - Print Spooling Control.

Controls the printer spooler, thereby allowing simultaneous printing and other processing. The spooler can be either stopped or started, or queried under system control. Printer calls (function 5) cannot be used properly while the spooler is active.

Starting the spooler requires passing an opened FCB to this function.

To test if a system has a spooler installed, the status call may be called with [A] = 2. If no spooler exists, then [A] will not change, otherwise it will be either 0 or 1.

Entry Parameters: (DE) has: FCB of spool file (to start spooler)
0 - not active
-1 - active

4.4.43. 86H/134 - Format FCB From String.

Formats a standard FCB from a text string. This is used to setup an FCB from scratch.

This filename will be terminated by the first occurrence of: space, non-printing character, equal sign (=), slash (/) or comma (,).

Entry Parameters: (DE) has address of where FCB is to be built.
(HL) has address of string with filename.

Return Parameters: None.

Example:

```
LD C,86H
LD DE,FCB
LD HL,STR
CALL 5
...
FCB:    DEFS 33
STR:    DEFM '<gname>'
                        ;<gname>=name.ext
        DEFB 0
```

4.4.44. 88H/136 - Chain to Program.

Allows one program to chain to another. The standard system region from 5CH to 0FFH is not modified. If control-C was disabled by the chaining program, then control-C will remain disabled; however, if the prior program set a control-C routine, it will be reset. If the program will not fit into available RAM, the system returns to the operator with the message "Load error". The "chained-to" program will receive whatever return code was set into the A register; if none was set, [A] will have zero.

Entry Parameters: (DE) has address of FCB containing filename of program to run.

Return Parameters: [A] returns -1 if file not found.

Example:

```
LD C,88H
LD DE,PGMN
CALL 5
      ; ONLY RETURNS HERE ON ERROR
...
PGMN:    DEFB     0
; NOTE FCB FORMAT OF NAME
         DEFM     "NEWPROGCOM"
```

4.4.45. 89H/137 - Multiply.

Provides a basic 16-bit unsigned multiply routine.

Entry Parameters: [DE] has multiplicand.
[HL] has multiplier.

Return Parameters: [DE] returns [DE] * [HL].

Example:

```
LD     DE,20
LD     HL,10
LD     C,89H
CALL    5
      ;  HERE DE=200
```

4.4.46. 8AH/138 - Divide.

Provides a basic 16-bit unsigned divide routine.

Entry parameters: [DE] has integer divisor.
[HL] has integer dividend.

Return Parameters: [HL], [BC] returned with [HL]/[DE].
[DE] returned with remainder ([HL] mod [DE]).

Example:

```
LD     DE,3
LD     HL,10
LD       C,8AH
CALL   5
                          ; HERE [DE]=1 AND [HL]=3
```

4.4.47. 8BH/139 - Home Disk Drive.

Forces disk heads to return to home position.

Entry Parameters: [B] has drive number (0=current, 1 = A:,2 = B: ...).

Return Parameters: None.

4.4.48. 8CH/140 - Eject Disk.

Request removal of disk indicated. Either physically ejects disk or requests operator to remove it, depending on disk hardware features.

Entry Parameters: [E]=disk (0=current, 1=A, 2=B...).

Return Parameters: None.

4.4.49 8DH/141 - Get Version and Serial Numbers.

Returns the version and serial numbers of the system. Can be used by programs to check that require features are available.

Return Parameters: [B] has Major Version number.
[C] has Minor Version number.
[HL] has serial number.
[DE] has configuration number.

4.4.50 8EH/142 - Set Terminal Function.

Provides a hardware-independent way of controlling an advanced CRT terminal. Section 3.2.2 has further details.

Entry Parameters: [D] has column and [E] has row to set cursor.
-or-
[E] has 0 and [D] has special function.

(For switch functions, on=odd, off=even)
Special functions (in [D])

0-clear	1-home	2-backspace	3-forespace
4-up	5-down	6-clr eol	7-clr eos
8-highlight	9-low light	10-normal light	11-keyboard on
12-keyboard off	13-cur.pad on	14-cur.pad off	15-protect on
16-protect off	17-blink on	18-blink off	19-line send
20-page send	21-aux send	22-del char	23-insert char
24-del line	25-insert line		

Cursor pad translates:

Up-^W
Left-^A Right-^D
Down-^Z

4.4.62. 9AH/154 - Test File Block Allocated.

Test if the file block specified is allocated. The current block of the FCB passed in DE is tested.

Entry Parameters: (DE) has address of the FCB of an open file.

Return Parameters: [A] returns:
0-Block allocated
-1-Block not allocated

4.4.63. 9CH/156 - Directory Listing.

Display a standard directory listing for files matched by the FCB passed.

THIS FUNCTION IS ACTIVE ONLY IF OPI IS RESIDENT (see OPIRES); otherwise the error "DIR function not active" is displayed.

Entry Parameters: (DE) has address of the FCB of file match request.

Exit Parameters: Registers AF, BL, DE & HL are not saved in this call.

4.4.64. 9DH/157 - Set Console Options.

Allows setting certain special characteristics for console activity. Affects functions 1,2,6,9,10. Reference paragraph 3.7.5.

Option bit

7 - Disable print output toggle (^W/^T)
6 - No echo on any input
5 - n/a
4 - n/a
3 - No echo of RETURN as terminator
2 - Enable ESCAPE as line terminator
1 - Disk read after write
0 - Control-P Flag

The function allows changing only certain options by having a change-mask of bits to be changed and a new value for those bits as separate values.

Entry Parameters: [E] has mask of bits to change.
[D] has new values of changed bits.

4.4.65. 9FH/159 - Get Master Disk.

Returns the current values for the command library master (DCOM) and the batch file master (DBAT). The indicated commands are used to set these values (DCOM or DBAT).

The returned values are A=0, B=1, etc.

Return Parameter: [A] has master disk (DCOM).
[B] has batch master (DBAT)

4.4.66. AOH/160 - Mount Disk.

Requests the specified disk be mounted.

Entry Parameters: [E] has disk number (A=0...).

Return Parameters: [A] returns:
0-OK.
-1-Disk already mounted.
-2-I/O error on mount.

4.4.67. A1H/161 - Dismount Disk.

Requests the specified disk be dismounted. The disk will always be logically dismounted for the caller; however, in a multi-user system, the disk will only be physically dismounted after all users have dismounted it.

Entry Parameters: [E] has disk number (A=0...)

Return Parameters: [A] returns:
0-OK or, in multi-user systems, bit vector of users logged on the disk. (bit 0-user 1,bit 1-user 2...).