USERS GUIDE

Disk Operating System (SDOS)

SD SYSTEMS

TABLE OF CONTENTS

GENERAL INFORMATION

SDOS is the SD SYSTEMS' DISK OPERATING SYSTEM. This system basically controls the disk input and output and also manages the disk files. SDOS is designed to permit users of SD SYSTEMS microcomputers to create and utilize both random and sequential disk files efficiently.

This manual is both a reference and programmers guide which has been divided into the five sections (see Table of Contents) to facilitate these uses.

SECTION 1

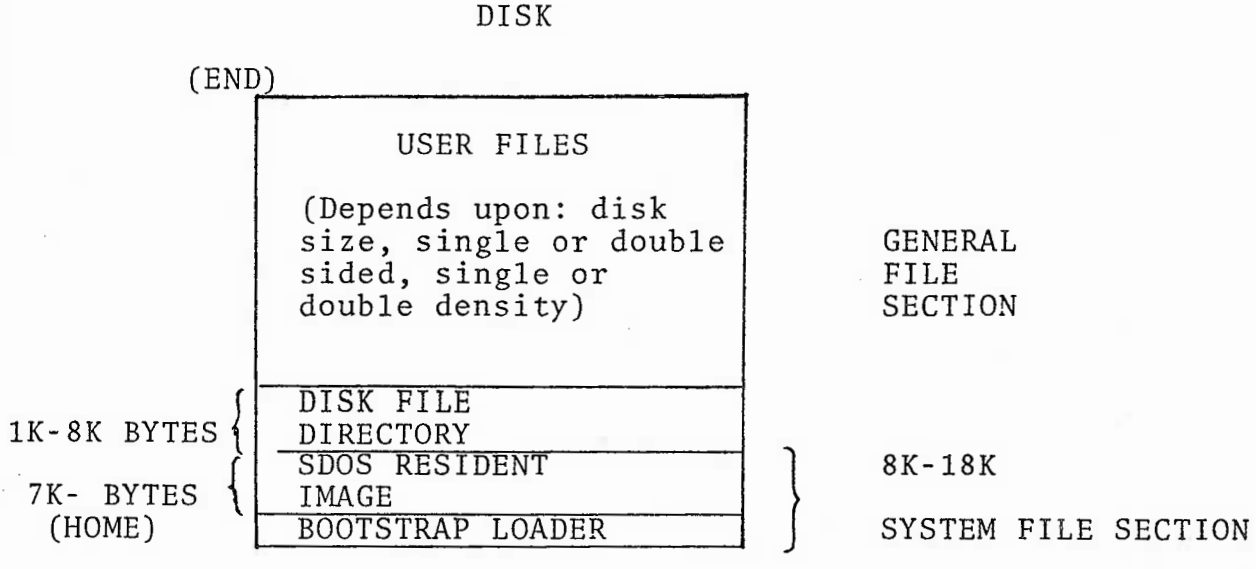
SDS SYSTEM ARRANGEMENT

1.0 DISK UTILIZATION

Disks used with SDOS are divided into two (2) sections. The first section is the System Section which contains the SDOS Resident Image and the boot strap loader (see SDOS Disk Utilization Map below) and the Disk File Directory.

The second section is the General File Section which contains the User Files. The Directory Command (DIR) lists only the contents of the User File area.

NOTE: The size of the User File area is dependent upon the size of the disk, the number of sides used (1 or 2) and the data density (single or double).



SDOS DISK UTILIZATION MAP

1.1 MEMORY UTILIZATION

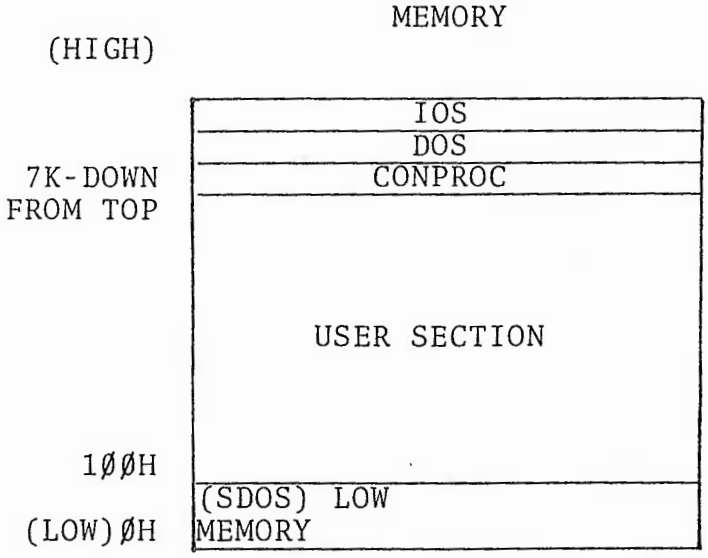
Memory is divided into two (2) main sections (the SDOS Section and the User Section).

The SDOS Section includes:

IOS-
DOS-
CONPROC- } Top 7K locations of memory

LOW MEMORY-Locations ØH-1ØØH- See Memory Utilization Map

The User Section is located between the bottom of DOS and the top of low memory (1ØØH), as CONPROC may be overlayed by the user program without affecting the execution of the program.



SDOS MEMORY UTILIZATION MAP

IOS- The Input/Output System contains the basic I/O functions for the console, printer, punch, reader and disk drives.

DOS- The Disk Operating System handles management, creation, opening, and reading of and writing to the disk files. In addition, DOS edits the console input.

CONPROC- The console is the usual user interface with the system. The Console Processor (CONPROC) calls a program and passes the parameters from the entered command to the program. CONPROC handles Internal, Utility and user commands. The Internal commands are placed in high memory when the system is loaded and run there. The Utility Commands and user programs are loaded in from the disk (as needed) into the User Area. If the user program makes a linkage call to a utility program the CONPROC cannot be overlayed or linkage will not be made.

USER PROGRAM SECTION- The User and Utility Programs are run in this area.

LOW MEMORY- These first 256 locations are reserved by SDOS for the following special purposes.

Ø(H)- System warm restart on user program exit

5(H)- SDOS entry addresses user requests (JP DOS)

8(H)- 3F(H)-Reserved for interrupt vectors
38H-3AH- Illegal address trap

4Ø(H)-5B(H)- Reserved for system

5C(H)-7F(H)- Standard user file control blocks

8Ø(H)-FF(H)- Standard user I/O buffer

1.2 FILES

A file is a place for storing information. A computer file is delineated by:

1. The code in which the data is encoded (HEX, ASCII*,etc).
2. The storage medium (punched card, magnetic tape, disk, etc)
3. The method of accessing the information (sequential or random).

1.3 FILE NAMES AND FILE REFERENCES

1.3.1. FILE NAMES - When a file is created it is given a primary file name of up to 8 characters and optionally, a secondary file name of up to 3 characters (e.g. eeeeeeee.yyy). It is the file name which enables a file to be referenced later. Any ASCII character (printable) may be used in either file name except the following:

$ * ? = / . , : "space"

Lower case characters of file names are converted by the system to upper case characters.

There are several secondary file names assigned by and/or expected by some system programs.

These are listed below:

.BAS - *BASIC source
.BAK - *Editor back-up
.$$$ - Temporary
.COM - Executable command program
.HEX - *Intel hex format (ASCII)
.PRN - *Print-out
.SYS - System image
.CMD - *Batch command file
.OBJ - S D Systems relocatable, linkable hex object (from ZASM

* ASCII - American Standard Code for Information Interchange

When an executive command file is referred to without specifying a disk drive for the file the current drive will be searched first. If not found (and the current drive is not the A drive) the A drive will then be searched for the file.

1.3.2. FILE REFERENCES (TWO TYPES)

1. The first type of file reference is called a specific (individual) file reference, which includes the primary file name (and optionally, the secondary file name) which refers to one file on a particular disk.

   The specific file reference locates the file with that name.

2. The second type of file is called an ambiguous (general) file reference that contains either a ? or an * which are seen by the system as replacement characters. This is used in certain CONPROC functions and utility programs to refer to a group of files.

   The ? and * are both the equivalent substitute characters for any character which appears in the same position in either the primary or secondary file name. However, the * also replaces any characters to the right, up to the . between the primary and secondary file names, or up to the end of the secondary file name if replaced after the ".".

e.g. If you have the following files;

FIL.OBJ
FILEA.BAS
FILEB.OBJ
FILEB.BAS
FILEA.OBJ
FIL.BAS

and the reference FILE?.OBJ is entered in a command line the following files will be found.

FILEA.OBJ
FILEB.OBJ
(but not FILEA.BAS or FILEB.BAS etc.)

NOTE: All file names can be referenced by using any combination of upper and lower case characters.

1.4 I/O DEVICE NAMES - The following I/O devices are available under SDOS:

DEVICE	NAME	DATA TRANSFER
Console	CON:	Input & Output
Card Reader	RDR:	Input
Paper-tape punch	PUN:	Output
Line Printer	LST:	Output
	PRT:	Output
Dummy Device	DUM:	
Disk	A:,B:, C:,D:	Input & Output

The following structure is used for I/O device names:

AAA:

AAA = a three character name

The device names listed above are those used by SDOS.

Only the console, printer and disk are active when shipped from SD Systems.

Any combination of up to four 8" and 5 1/4" disk drives may be connected under SDOS. All disk files can be accessed in either random or sequential order.

SECTION 2

SDOS OPERATION

2.0 STARTUP

Turn on the SD computer and the terminal. Place a disk with the SDOS System in drive A*. If you are operating with an RS-232 console, press the carriage RETURN key (CR) once to set the baud rate. If you are using the VDB console press the RESET switch once.

The computer will respond with a prompt "." Enter a "C" (upper case) and a carriage RETURN (CR). This boots up the system, which is now ready ([A]).

This procedure is called a cold boot and loads the operating system into memory. Upon completion of the cold boot drive A is the currently selected drive and the disk in drive A has been logged in.

If you have previously created a batch file named STARTUP.CMD (which is a batch command file) and it is on the boot disk (the disk from which the system was cold booted) it will automatically be run at this time. The programs stored in this file will be run sequentially (see Batch Utility Command-p. 26). If, when creating the STARTUP.CMD file, a password request program was entered as the first command, a request for a password

* Cold Boot can only be done from drive A

will be displayed as shown below.

```
[A] @ STARTUP
BATCH VERSION 01.00

[A] SIGNON
ENTER PASSWORD
```

2.1 DISK DRIVE SELECTION (when in the console processor)

To select a disk drive other than the currently selected drive simple enter the drive name (A, B, C or D) followed by a colon (:) and press the carriage RETURN key (cr).

e.g. If the currently selected disk drive is A and access to disk drive B is desired enter the following: B: (cr) The console will display [B] to indicate that drive B is now the currently selected disk drive.

Selecting a drive in this manner causes the disk in that drive at the time of selection to be logged in so that the system knows what file space is available (unused) on the disk.

NOTE: Drive selection can be done in conjunction with the entry of any command on the console.

2.2 CHANGING DISKS AFTER COLD BOOT

When another disk is subsequently inserted into the currently selected drive the disk MUST be logged-in. The reason, for logging-in the subsequently inserted disk, is that the bit map (which tells the system where unused space is available on the disk) for the previously inserted disk, is still resident in memory. The subsequently inserted disk will have a different bit map and if this disk is written to, it probably will be done improperly if it is not logged-in.

To log-in the subsequent disk, press the CTRL (Control) key and then enter an upper case C. This procedure is called a warm boot and will log-off all disks and then log-in the subsequently inserted disk on the currently selected drive (and the disk in drive A; if it is not the currently selected drive).

2.3 CONTROL FUNCTIONS

Certain non-printing (control) characters control specific console and printer operations. These characters are described in the following sections.

NOTE: A control character is entered on the console by holding down the CTRL key and simultaneously depressing the appropriate additional key (i.e., A for CTRL-A, C for CTRL-C, etc.). Control characters are not printed but they are displayed on the console by the up arrow (^) followed by the appropriate character; i.e. (^R).

CONSOLE CONTROL CHARACTERS

Once the system is running, basic user interaction is with the console. The CONsole PROCessor (CONPROC) program takes care of file maintenance by means of the INTERNAL commands and the utility programs which are described in Section 3 of this manual. The prompt given by the system is the disk identifier for the current drive (e.g., [A] for drive A). A command may be entered anytime the prompt is displayed.

While entering a command the standard buffer input mode is active and certain control characters are usable. The following is a list of control characters and their functions:

Backspace Delete	Either of these keys will backup the CRT cursor and delete the last character entered.
Underscore	Prints slash "/" and then echos previously entered data at the console each time it is entered. When another character is entered a trailing "/" is printed followed by the character.
RETURN ^M	Either of these will terminate a command line.
^R	Retype current line (after many corrections).
^S	Pause during device I/O. This is primarily used to stop and re-start a listing on the console. Any key may be typed to resume processing, but only ^S can be used to pause.

^U Deletes the current line (CRT Only). Moves cursor to the left margin and does not do a linefeed.

^X Deletes current line (goes to next line).

PRINTER CONTROL CHARACTERS

There are three control characters which are used to control output to the printer. They are:

^P Send all console output to the printer as well as to the console. This is a toggle switch action. If the printer is off, it is turned on by simultaneously pressing the control (CTRL) key and the P key. After a CTRL-P is entered, entering another CTRL-P cancels the first one.

^\ (ICH) Turn off all the output to the printer. This control character can be output by a user program but will have no effect if issued from the console.*

^] (IDH) Send all output to the printer as well as to the console. This control character can be output by a user program but will have no effect if issued from the console.*

NOTE:

1. * These characters will not be sent to VDB-8024 under SDOS.

SECTION 3

SDOS CONSOLE PROCESSOR COMMANDS

3.0 COMMAND STRUCTURE AND SYNTAX

The Command (.COM) files can be executed at anytime when the system prompt is displayed by entering (at the console) the primary file name of the command without the secondary file name (.COM is assumed), and the remaining elements of the command line. Then the console processor determines if it is an internal command. If it is not an internal command, the currently logged-in disk drive (and disk drive A, if it is not the currently logged-in drive) will be searched until the system finds the utility or user command file. When the program is found it is loaded into memory starting at 1ØØ H, with the remainder of the command line passed to the program as control information. Execution of the command starts at 1ØØH. If the command file is not found the following message is displayed on the console: PROGRAM NOT FOUND

Each command line can start with a disk drive specifier (This is optional). The command (without the secondary name, .COM, which is assumed) is then entered. The remaining content of the command line is determined by the individual commands. The following conventions have been observed in the structuring of the command lines.

1. Any combination of upper and lower case characters may be entered in a command line (as they will be automatically translated by the system into upper case characters).

2. The REN, WRTSYS, XFER commands use the following structure:

   Destination-File-Ref=Source-File-Ref

3. An equal sign acts as a delimiter to separate source and destination file names. Spaces may be used to separate file names when concatinating files.

4. Options are preceded by a space and a slash [/] .

NOTE: In explaining the structure of command lines the following symbols and combinations have been used.

[ ] Brackets are used to indicate a parameter which is optional. An item so enclosed may be entered at the user's discretion.

{ } Braces are used to indicate a choice of items. One of the enclosed items must be selected (see the WRTSYS and XFER commands).

[{ }] Brackets enclosing braces indicate an optional choice of items. (See the DIR command).

— Character underline is used to show which characters must be entered if used.

3.1 INTERNAL COMMANDS

The Internal Commands reside in SDOS high memory once the system has been cold booted.

3.1.1. ATRIB

Enables the user to set up, change or delete parameters which protect a file from being read, written to or erased.

COMMAND LINE STRUCTURE:

ATRIB file.ref [+] [p....] (cr)

Where:

file.ref — is the reference (which may be ambiguous) to the file(s) to which attributes are to be assigned or deleted.

\+ — This parameter is required only when assigning additional protective parameters to a file and can only be used at that time.

[p...] — are the protective parameters to be assigned to the file

- P — Protects a file from being erased
- R — Protects a file from being read by a TYPE or DUMP command. However, without additional parameters, it can be erased or executed.
- W — Protects a file from being written into. However, without additional parameters it can be erased or executed.

(cr) — carriage RETURN

Attributes already assigned to a file will be deleted by assigning new attributes without using the + sign. All attributes assigned to a file will be deleted by entering only ATRIB and the file reference in the command line.

EXAMPLE:

The following disk directory display shows most files with one or more protective parameters.

```
FILE11.FAX   2K   1   R
FILE22.FAX   5K   1   RW
FILE33.FAX   7K   1   RWP
FILE44.FAZ   3K   1
```

To add write protection to FILE11.FAX simply enter:

```
ATRIB FILE11.FAX +W (cr)
```

To remove read and write protection from FILE22.FAX simple enter:

```
ATRIB FILE22.FAX (cr)
```

To assign erase protection to FILE44.FAZ simple enter:

```
ATRIB FILE44.FAZ P (cr)
```

If the disk directory is displayed again it will appear as follows:

```
FILE11.FAX  2K   1   RW
FILE22.FAX  5K   1
FILE33.FAX  7K   1   RWP
FILE44.FAZ  3K   1   P
```

3.1.2. DIR

Lists the disk files giving their size (in K bytes), number of extents and the file attributes.

COMMAND LINE STRUCTURE:

DIR [{ Y: / file.ref }] (cr)

Where:

Y: Specifies the disk drive whose directory of files is to be displayed. If a drive is not specified, the directory of the currently selected drive will be displayed.

file.ref Specifies the file for which the size, number of extents and attributes are to be displayed. This file reference may include the replacement characters ? and/or *.

e.g. Entering: DIR FILE??.* will cause the display of the following headings and files.

NAME	EXT	SIZE	EXTENTS	ATTRIBUTES
FILE11	FAX	1K	1	PW
FILE21	FAX	1K	1	R
FILE31	FAX	1K	1	
FILE41	FAZ	1K	1	
Totals	4FILES	4K	4	

Entering: DIR FILE41.FAZ will cause the following to be displayed on the console.

NAME	EXT	SIZE	EXTENTS	ATTRIBUTES
FILE41	FAZ	1K	1	
Totals	1 File	1K	1	

NOTE: Pressing any key during the displaying of the directory will cause it to abort (displaying the totals for the files displayed up to that point).

Entering a CTRL S while the directory is being displayed will cause the display to pause until another key is pressed.

The size parameter denotes the rounded up number of K (1024 bytes) actually written on the diskette. However, the number of K allocated for that file may be different as indicated by XSTAT. Minimum file allocations for different diskette formats are as shown below:

TYPE	MIN FILE ALLOCATION
Single side Single density	1K
Double side Single density	2K
Single side Double density	2K
Double side Double density	4K

3.1.3. ERA

Deletes the file(s) from the disk directory.

COMMAND LINE STRUCTURE:

ERA file.ref (cr)

Where:

file.ref	Specifies the file(s) which will be deleted from the disk directory. This file reference may include the replacement characters ? and/or *. The disk space previously used by the erased files will then be available.

NOTE: Caution is recommended when replacement characters are used, as it is possible to delete a large number of files this way.

e.g. Entering ERA FILE??.* (cr) will erase all the following:

FILE11.FAX
FILE21.FAX
FILE31.FAX
FILE22.EXA
FILE33.SAM
FILE44.EI

3.1.4. GET

Enables the user to load into memory (and optionally run) any .HEX absolute object file. The file will be loaded into the address specified by the INTEL hex object format.

COMMAND LINE STRUCTURE:

GET file.ref [/G] (cr)

Where:

file.ref	Specifies the hex files which you wish to get (load into memory). If the secondary file name is omitted .HEX is assumed.
/G	Specifies, that upon loading, the program is to be run immediately starting at the first address specified in the .HEX file.

3.1.5. MON (cr)

Returns the system to SD Monitor, where any of the SD Monitor commands may be used. Once in SD Monitor, control may be returned to SDOS by entering:

C (cr) or

G Ø (cr) (go starting at location Ø).

3.1.6. REN

Enables the user to rename a file (either the primary name, secondary name or both).

COMMAND LINE STRUCTURE:

REN file.ref1=file.ref2 (cr)

Where:

file.ref1 — Is the new file reference. This file reference may include the replacement characters ? and/or *. NOTE: All characters from the primary and secondary file names (referred to by the old file.ref) will be substituted for the ambiguous characters of the new file reference.

file.ref2 — Is the old file reference. This file reference may include the replacement characters ? and/or *.

NOTE: Initially, the disk is searched for any file(s) which would satisfy the file.ref1. If any files satisfying file.ref1 are found, the REN command is terminated. This check is the only check made and therefore it is possible,using an ambiguous file.ref2,to rename more than one file with the same name.

CAUTION: Do not rename any file with an existing file name as you will only be able to access the file with that name which appears first in the directory. If you should have more than one file with the same name you can gain access to all the files by renaming them.

e.g. With the following files

FILE.ABC

FI12.FAX

FI13.FAX

FI14.FAX

entering: REN SAME.TAG=*.FAX (cr) will cause the directory to display the following:

FILE.ABC

SAME.TAG

SAME.TAG

SAME.TAG

3.1.7. SAVE

Saves the specified number of 256 byte pages of the user area starting at address 100H.

COMMAND LINE STRUCTURE:

SAVE file.ref N (cr)

Where:

file.ref	Specifies the file name to be given to the contents saved from the user area. If the secondary file name is entered as ".COM" the file will be saved as a command file.
N	Is the number (decimal) of pages to be saved.

3.1.8. SET

The SET command provides a means of re-assigning the listing on console to the non-default device.

COMMAND LINE STRUCTURE:

```
          0 { for parallel list device }
SET L = 1 { for serial list device   } (cr)

          0 {for VDB-80 console }
SET C = 1 {for RS-232 console } (cr)
```

3.1.9. TYPE

Displays the contents of the ASCII file referenced on the console.

COMMAND LINE STRUCTURE:

TYPE file.ref (cr)

Where:

file.ref	Specifies the file whose contents are to be displayed on the console.

NOTE: Pressing any key will terminate the command. The display will pause when the CTRL Key and the S Key are pressed simultaneously. Pressing any other key will resume the display.

3.2 UTILITY COMMANDS

Utility Commands are resident on the disk as .COM files which are loaded into the user area as needed. Execution of Utility Programs alters the user area.

3.2.1. @ (BATCH)

The @ (BATCH) Command enables the user to execute a file of commands sequentially until the file is exhausted. This allows job stream processing.

The @ (BATCH) Command line can have either of two structures:

1. One time execution structure in which a temporary file ($$$$.CMD) is created and immediately executed upon entering a carriage RETURN (cr) on a line with no command.
2. Permanent file execution structure in which a permanent file is created (using the Text Editor) and whose commands can be executed at any time when the system prompt is displayed:

1. COMMAND LINE STRUCTURE (for one time execution):

   [X:] @ [=R] [=Y] (CR)

2. COMMAND LINE STRUCTURE (for repetitive execution):

   [X:] @ [=R] [=Y] file.ref [P1 P2 P3...P9] (CR)

Where:

X:	Specifies the disk drive location of the batch COM file (@.COM). This parameter is required only if the @.COM file is not located on the disk in the currently selected drive (or drive A if it is not the currently selected drive). Values are A, B, C and D.
=R	Is an optional nesting function switch which enables a batch command (.CMD) file to call another batch command (.CMD) file and to return to the first .CMD file when execution of the second .CMD file is completed. If this switch is not set there will be no return to the first file. The level of nesting is equal to the maximum number of commands pending at one time (128).
=Y	Specifies the disk drive on which the batch work file ($$$$.CMD) is located (optional). Defaults to the A drive.
file.ref	Is the batch command file (CMD) to be executed.
P1,P2...9	Are optional parameters to be passed to the .CMD file upon entering the @(BATCH) Command line.
(CR)	Carriage RETURN.

NOTE: @(BATCH) Command executes the commands in the .CMD file, sequentially until the file is exhausted. Each command is displayed on the console just before execution.

When the one time execution batch of commands are entered a temporary command file ($$$$.CMD) is created by the @(BATCH) Command. The prompt for this temporary file is an exclamation mark (!). All legal responses for the SDOS prompt ([A]) are valid. e.g.

```
[A]  @ (CR)
     !DIR (CR)
     !TYPE FILEA
     !PRINT FILEB
     ! (CR)
```

One Time Execution Made

When the repetitive execution batch file is created, the Text Editor-EDIT is used to enter any legal primary file name and a mandatory secondary file name of CMD. (e.g. EDIT SAMPLE.CMD)

If parameters are to be passed to the file commands from the @(BATCH) Command line, an up carat (^) followed by the number of the parameter (e.g. ^1) must be entered.

```
e.g.  EDIT  SAMPLE.CMD
            TYPE ^Ø   (CR)
            DIR ^1    (CR)
            REN ^2    (CR)
            PRINT ^3  (CR)
            PRINT ^1  (CR)
            @ ^4      (CR)
            Q         (CR) ends editing
```

If an ^Ø is entered after a file command, upon execution of the batch file the batch file itself will be references. e.g. The command TYPE ^Ø in the example above will cause the contents of the above file (SAMPLE.CMD) to be displayed on the console (as shown above). Parameters (^1-^9) are those which correspond to the position occupied by the parameters entered in the @(BATCH) Command line.

Batch commands can be linked by entering another @ (batch) command, as the last command in the command file.

In order to have an automatic startup upon cold booting the system, create (using the Text Editor) the file STARTUP.CMD with whatever commands are required by the system upon startup.

When a @ (BATCH) Command, referencing a permanent command file, is processed by the system, each word (characters separated by a character space or an = sign) after the file reference is considered to be a parameter. Complex parameters (ones which include character spaces + = signs) must be enclosed in single quotation marks.

Upon entering the following @ (BATCH) command line:
@ SAMPLE.CMD FILE.CMD 'PROGRAM.CMD=BLOCK.CMD'ITEMS.CMD RECORDS.CMD (cr)
with the SAMPLE.CMD file containing the commands listed on page 28 the following will take place.

The system will display on the console the contents of SAMPLE.CMD (see page 28) followed by:

```
DIR FILE.CMD
```

(displaying the directory listing along with the size)
Followed by:

```
REN PROGRAM.CMD=BLOCK.CMD
PRINT ITEMS.CMD  (ITEMS.CMD is also printed out)
```

```
PRINT FILE.CMD  (FILE.CMD is also printed out)
@ RECORDS.CMD  (This batch file is processed)
```

NOTES: If more parameters are inserted in the command line than are needed by the batch file being processed, the additional parameters are ignored. If there are fewer parameters in the command line than called for by the batch file being processed, those commands that require the additional parameters are ignored.

The error message "BATCH FILE ERROR" can be caused by either a physically write protected diskette, or the batch file disk being full (either space or directory).

3.2.2. B1-SIDED

Designates the disk in drive B as being one sided, thus the disk will be read and written to as a 1-sided disk.

COMMAND LINE STRUCTURE:

B1-SIDED (cr)

NOTE: B1-SIDED and B2-SIDED are not needed on systems using the Versafloppy II Disk Controller (e.g. SD-200) because the disk type is determined automatically.

3. .3. B2-SIDED

Designates the disk in drive B as being two sided, thus the disk will be read and written to as a 2-sided disk.

COMMAND LINE STRUCTURE:

B2-SIDED (cr)

NOTE: B1-SIDED and B2-SIDED are not needed on systems using the Versafloppy II Disk Controller (e.g. SD-200) because the disk type is determined automatically.

3.2.4. DCOPY

Copies all files from the disk in drive A to the disk in drive B. Diskettes in each drive must have identical formatting.

COMMAND LINE STRUCTURE:

[X:] DCOPY (cr)

Where:

X: Specifies the disk drive location of DCOPY.COM file. This parameter is required only if the .COM file is not located on the currently selected drive (or drive A if it is not the currently selected drive). Values are A and B.

Error Messages are as follows:

DRIVE A READ ERROR - Error occurred while trying to read diskette in drive A.

DRIVE B READ ERROR - Error occurred while trying to reread diskette in drive B.

DRIVE B WRITE ERROR- Error occurred while trying to write to diskette in drive B.

VERIFICATION ERROR- Data reread from drive B does not compare with data written to drive B.

3.2.5. DSKDIAG

Used with the SD-100/200 to test the drives for faulty operation.

COMMAND LINE STRUCTURE:

[X:] DSKDIAG (cr)

Where:

X Specifies the disk drive location of the DSKDIAG.COM file. This parameter is required only if the COM file is not located on the currently selected drive (or drive A, if it is not the currently selected drive). Values are A, B, C and D.

Entry of the command line will cause the following prompt

TEST # DRIVE # [TTDD] :

The user may then enter the following:

1XX - Read/Write Test; Reads and Writes each sector sequentially
2XX - Read Test; Reads each sector sequentially
3XX - Read/Write Test; Reads and Writes random sectors on the drive specified
4XX - Read/Write Test; Reads and Writes random sectors and random drive
5XX - Formats the diskette specified
10XX - Loads data from diskette into memory (no file reference)
11XX - Saves data from memory to diskette (no file reference)
FFØØ - Exits the program to location specified by user
[.] - Exits to monitor
(cr) - Exits back to SDOS

Where:

TT is the number 1-5, 10, 11 or FF

DD (or XX) is the drive specification as follows:

Type / Drive	#Sides Disk Size Density	1 Full Single	2 Full Single	1 Mini Single	2 Mini Single	1 Full Double	2 Full Double	1 Mini Double	2 Mini Double
A		00	10	20	30	40	50	60	70
B		B:SD 01	11	21	31	41	51	61	71
C		02	12	22	32	42	52	62	72
D		03	13	23	33	43	53	63	73

3.2.6. DSKDUMP

Displays the contents of selected block (on selected drive) in HEX.

COMMAND LINE STRUCTURE:

[X:] DSKDUMP [Y:] [file.Ref]_[E] (cr)

Where:

X — Specifies the disk drive location of the DSKDUMP.COM file. This parameter is required only if the COM file is not located on the currently selected drive (or drive A, if it is not the currently selected drive). Values are A, B, C, and D.

Y — Specifies the disk drive whose disk is to be dumped.

file.ref — If this field is not entered, then disk [Y] will be accessed by absolute or interleaved block addresses. If a file name has been entered then the block read will be the designated block number of that file rather than the absolute block number.

E — Extent (16K block) of file to be accessed (defaults to 0)

The response to the system request 'Block', asking for a block number (Hex), determines what happens. If just a hex value (i.e. 34) is entered block 34 (Hex) will be read into the working buffer and displayed on the console. If the number is immediately followed by the suffix I (i.e. 35I) then an interleaved read is performed by the system (i.e. the next block of the file that was read at 34, will be read and displayed; rather than the next physical block on the disk. Pressing any key will stop the display.

If one of the command letters (W, M, or S) are entered instead of the hex value, the corresponding operation will be performed.
M-modify S-show W-write

M (Modify)

Structure: MX, value [,value,...value] (cr)

Where:

X	is the first hex location in the block to be changed
Value	is either a hex value or a character string. Each value gets put in a successive location. e.g. If you enter: M20, 23, 'ABC', 10 (cr) you will modify locations 20H-24H of the block in the buffer.

S (Show)

Structure: S (cr)

When the character S is entered followed by the carriage RETURN (cr) the block in the buffer (with any modifications) is displayed.

W (Write)

Structures: W (cr)
WXXXX (cr)
WXXXXI (cr)

Where:

XXXX	is the block number
I	specifies an interleaved write

e.g. W- causes the block in the buffer to be written back to the disk from where it was read.

W35- writes the block in the buffer to block 35, (absolute block number if no file name was entered.)

W35I- writes the block in the buffer to block 35 Interleaved (absolute block number if no file name was entered).

3.2.7. DTYPE

Used with the SD-100/200 to display the system and drive configuration.

COMMAND LINE STRUCTURE:

[X:] DTYPE (cr)

Where:

X Specifies the disk drive location of the DTYPE.COM file. This parameter is required only if the COM file is not located on the currently selected drive (or drive A, if it is not the currently selected drive). Values are A, B, C and D.

If the system is an SD-100 the following message is output:

```
SD-100 Microcomputer System
Hardware is configured as:  Double (or single) sided
(where the number of sides is modified according to
hardware configuration)

System software is configured as follows:
Drive A is:  Single density double (or single) sided 8"
Drive B is:  Single density double (or single) sided 8"
Drive C is:  Single density double (or single) sided 8"
Drive D is:  SIngle density double (or single) sided 8"
```

(where the number of sides is modified according to the SDOS configuration)

If the system is an SD-200 the following message is output:

```
SD-200 Microcomputer System
Diskettes in the system are configured as follows:
Diskette in Drive A is:  (Diskette configuration)
Diskette in Drive B is:  (Diskette configuration)
Diskette in Drive C is:  (Diskette configuration)
Diskette in Drive D is:  (Diskette configuration)
```

(where only drives with the doors closed and diskettes inserted are listed)

3.2.8. DUMP

Displays (on the console) the contents of the specified file in both hexadecimal and ASCII.

COMMAND LINE STRUCTURE: [X:] DUMP file.ref (cr)

Where:

X	Specifies the disk drive whose diskette contains the DUMP.COM file. This parameter is required only if the COM file is not located on the currently selected drive (or drive A, if it is not the currently selected drive). Values are A, B, C and D.
file.ref	is the file to be dumped.

The hexadecimal contents of the dumped file together with the first address of a line are displayed to the left. The corresponding ASCII characters are displayed to the right. Pressing any key stops the dump.

NOTE: If a binary file is dumped the area of the console that normally displays the ASCII characters, is meaningless.

3.2.9. EDIT

Enables the user to create and modify files (both program and data). See the Text Editor, Z-80 Global Assembler, Linker Manual for details.

3.2.10. ERASE

Erase files, on the condition that the user enters Y when asked if the file displayed is to be erased.

COMMAND LINE STRUCTURE: [X:] ERASE file.ref (cr)

Where:

X	Specifies the disk drive whose diskette contains the ERASE.COM file. This parameter is required only if the COM file is not located on the currently selected drive (or drive A, if it is not the currently selected drive). Values are A, B, C and D.
file.ref	Specifies the ambiguous file (s) that are to be erased.

Example: When the following command line is entered ERASE FILE.XYZ (cr) the system responds by asking:

```
CONDITIONAL ERASE VERSION 00.01
FILEA.XYZ
```

User responds N(NO) or Y(YES).

The system then takes the action entered by the user. The system continues to list the next file until all files on the disk that match the ambiguous file reference have been listed.

3.2.11. FORMAT

Used with SD-100/200 to initialize a diskette.

COMMAND LINE STRUCTURE:

[X:] FORMAT {A B C D} [0-7] (cr) (single disk formatting)

[X:] FORMAT (cr) (multiple disk formatting)

Where:

X — Specifies the disk drive location of the FORMAT.COM file. This parameter is required only if the COM file is not located on the currently selected drive (or drive A, if it not the currently selected drive). Values are A, B, C and D.

A,B,C,D — is the disk drive in which the disk is to be formatted resides. If any other values is entered the program comes up with an error message, "DRIVE SPECIFICATION ERROR" and aborts.

Entry of either the Command Line Structures of this program will initially prompt the user (to specify the type) as follows:

FOR SD-200 SYSTEMS:

SD-200 Format Routine

Specify Type (0-7)

0- 8" - 1 sided-single density
1- 8" - 2 sided-single density
2- 5" - 1 sided-single density
3- 5" - 2 sided-single density
4- 8" - 1 sided-double density
5- 8" - 2 sided-double density
6- 5" - 1 sided-double density
7- 5" - 2 sided-double density
Diskette Type:

For SD-100 systems:

SD-100 Format Routine

Specify format as follows:

0 - Single-sided
1 - Double-sided,
Diskette type:

NOTE: If you enter full command line for single disk formatting (options entered) you will not be prompted as shown above.

NOTE: If the multiple disk formatting command line was entered, the above will be proceeded by the following:

FORMAT DRIVE B, MULTIPLE FORMAT OPTION

ENTER CARRIAGE RETURN TO ESCAPE (allowing escape from the program)

When an error message of "Diskette Specification Error" appears, you have three choices:

1. Enter a correct number.
2. Press the R key and obtain a redisplay of the type descriptions and numbers.
3. Enter any other key which will cause aborting of the program.

If in multiple format mode when the first diskette has been formatted, the type options will be redisplayed automatically. After inserting a new disk in the B drive only the type numbers need to be inserted. This is a multiple format option only.

When an error message of "Drive Not Ready" appears, you may enter an R which will cause the system to repeat its attempt to access the disk in drive B. Entry of another key will return to SDOS.

NOTE: The same effect can be accomplished when formatting the disk before cold booting using the SD Monitor Z command. The key sequence is as follows:

Z XY Z=character Z
X=one of the types listed above
Y=drive (0-3) in which the disk to be formatted resides

NOTE: Diskettes formatted with the Z command must use 'Z1Ø' on double sided SD-100 systems and 'ZØØ' on single sided SD-100 systems to be compatible with the SD-200 system.

3.2.12. LINK

Concantenates modules together and resolved global symbol references which provide communication between modules. See the Text Editor, Z-80 Global Assembler, Linker Manual for details.

3.2.13. LOAD

Loads the Hex object file referenced and converts the file to a COM file (binary). The hex object file is left intact.

COMMAND LINE STRUCTURE: [X:] LOAD file.ref (cr)

Where:

X — Specifies the disk drive location of the LOAD.COM. This parameter is required only if not located on the currently selected drive (or drive A, if it is not the currently selected drive). Values are A, B, C and D.

file.ref — Specifies the file to be loaded. This must be a file with the secondary name HEX.

3.2.14. WRTSYS

Provides a means of copying SDOS from one diskette to another.

COMMAND LINE STRUCTURE:

$$[X:]\ \underline{WRTSYS}\ [\ /S]\ \left\{\begin{matrix}Y:\\ file.ref1\end{matrix}\right\} = \left\{\begin{matrix}Z:\\ file.ref2\end{matrix}\right\} (cr)$$

Where:

X — Specifies the disk drive location of the WRTSYS.COM file. This parameter is required only if the COM file is not located on the currently selected drive (or drive A, if it is not the currently selected drive). Values are A, B, C and D.

/S — Specifies that the system is to be written from one disk to another disk,subsequently inserted in the same drive. A prompt will tell the user when to insert the second diskette.

Y — Specifies the disk drive which is to be written to. NOTE: If only the drive designation is given SDOS will be written to the system area.

file.ref1 — Specifies the file which is to be written to and must include the secondary file name of .SYS if no disk drive is specified before the file.

NOTE: When using a file.ref, precede it with the drive designation of the disk on which the file is to be found.

Z Specifies the source disk drive.

NOTE: If only the drive designation is given the system will be read from the system area.

file.ref2 Specifies the file which is to be read from and must include the secondary file name of SYS.

NOTE: The memory resident SDOS is changed only upon booting the system.

Examples: WRTSYS B:=A: (cr)

will copy SDOS from the system area of the disk on drive A to the system area of the disk on drive B.

WRTSYS A:MOD.SYS=B:OLD.SYS (cr)

will copy the SDOS from OLD.SYS file on the disk in drive B to MOD.SYS file on the disk in drive A.

3.2.15. XFER

Copies files from the file area of a disk (or other device) to file area of another disk (or device). This command has the following two structures:

1. COMMAND LINE STRUCTURE for Repetitive Execution:

   [X:] XFER (cr)

2. COMMAND LINE STRUCTURE for One Time Execution:

   [X:] XFER [/p1 /p2...{d: / file.ref1} = file.ref2 [,file.ref3...] (cr)

When the repetitive execution mode is entered the command prompts the user with an exclamation mark (!). The valid responses are the entry of the destination drive (and file), and the entry of one or more source files. This creates a temporary batch file of transfers (see Batch Command Page 26).

NOTE: The source file (or destination file) drive only needs to be entered if the file(s)is not on the currently selected drive.

Where:

X — Specifies the disk drive location of the XFER.COM file. This parameter is required only if the XFER.COM file is not located on the currently selected drive, (or drive A, if it is not the currently selected drive). Values are A, B, C and D.

/p1 /p2... are the combination of the following optional parameters.

A designates an ASCII file transfer

C designates a comparison of files (without actually copying them). This comparison is governed by the source file (file.ref2) and; if it is identical with file-ref1 for the length of file.ref2, the two files are perceived by the system to be the same.

F designates that illegal ASCII characters are to dropped

I designates that the ASCII end of file should be ignored. (CTRL-Z)

R designates that a read protected file is to be copied. When copying more than one source file the read protected source file will only be copied if it is the first source file.ref and the only source file-reference that is read protected.

S designates that all rub-outs and nulls are to be stripped from the file. This applies only to ASCII files.

T designates that the tabs are expanded. This applies only to ASCII files.

V designates that the files are to be verified after transfer

Z designates that the size statistics are not to be printed upon completion of the XFER.

d: specifies the destination disk drive. When used alone, the original primary and secondary names will be preserved.

file.ref1 is the destination file reference

file.ref2,3... are the source file references. If more than one file ref is given the files will be concantenated (linked). If ASCII files are linked the A parameter must be used to remove the end of file marks.

Example: XFER /A /F /Z A:FILE31.FAX=B:FILE22.EXA. B:FILE33.SAM (cr)

The ASCII files, FILE22.EXA and FILE33.SAM (located on disk drive B) will be concantenated and transferred to FILE31.FAX on disk drive A. Any illegal characters in any of the transferred files will be dropped (/F). After the transfer the size statistics will not be displayed (/Z).

NOTE: When concantenating files as in the example above, it is a good idea to put the disk specifier before each of the files. This is only required if the currently selected drive is other than the drive with the files to be concantenated.

3.2.16. XSTAT

Used to display the status of the disk in the currently selected drive.

COMMAND LINE STRUCTURE:

[X:]XSTAT [Y:] (cr)

Where:

X — Specifies the disk drive location of the XSTAT.COM file. This parameter is required only if the COM file is not located on the currently selected drive (or drive A, if it is not the currently selected drive). Values are A, B, C and D.

Y — Specifies the disk drive of the disk whose status is to be displayed. Values are A, B, C and D.

When the command line is entered the following is normally displayed.

```
XSTAT   Version 00.00

 XXXK   bytes TOTAL disk space (available for user files)*
        (This includes the directory space -8K)
  XXK   bytes user RAM memory (size of user area)*

  XXK   bytes disk space LEFT (unused space available
        for user files)  This is the number of unallocated
        clusters multiplied by four.

   XX   Directory Entries
```

NOTE: XX is the number of bytes

Additionally, if there are any empty files (null files) their names

*NOTE: Bytes are allocated in 4K clusters.

will be printed out. XSTAT also runs a validation on the disk directory to find any cross-linked files. These types of errors are caused by failure to warm boot (CTRL C) when changing disks in the currently selected drive.

The following error messages may appear as a result of the validation check:

XXH NOT ALLOCATED - You still have not entered a (CTRL C). So far no damage has occurred.

XXH LINKED CLUSTER file.Ref, file.Ref - same disk file cluster block has been allocated in both files. The most recent of the files should be OK. Delete at least the old file, and to be sure, delete both files. ASCII files can be checked by TYPEing or DUMPing them.

3.2.17. ZASM

Used to assemble Z-80 source programs (Mostek and Zilog definitions). This assembler outputs an assembly listing on the printer and an object code file on the disk. See the Text Editor, Z-80 Global Assembler, Linker Manual for details.

SECTION 4

PROGRAMMERS GUIDE

4.1 INTRODUCTION

SDOS is the operating system for the SD SYSTEMS computers. The system defines certain conventions and provides the programmer with I/O functions. These functions allow a program to ignore details of device operation and to specify a device symbolically.

The resident operating system consists of three (3) main parts, IOS, DOS and CONPROC. The usual sequence of system operations is to use CONPROC to run a user program, which then passes all system requests to DOS. DOS provides both character device I/O and disk block I/O.

The system conventions are:

1. Loading and starting execution of programs at 100H (see detailed Low Memory Utilization Map below). Once loaded, a program may use any part of memory, including that used by DOS or IOS. If the program destroys DOS or IOS it will not have any way to use the disk or recover the operating system without the user rebooting (jumping to FØØØ).

NOTE: If the GET command is used loading and execution can begin at locations other than 100H.

100	Beginning of User Programs
80-FF	Command Line Buffer (first byte=length)
7C-7F	Reserved
6C-7B	File Control Block 2 (FCB-2)
5C-6B	File Control BLock 1 (FCB-1)
40-5B	Disk Controller Parameters
8-3F	Interrupt Vectors
5	System Call Entry
4	Reserved
3	I/O Byte
0-2	SDOS Re-Entry (warm start)

DETAILED LOW MEMORY UTILIZATION MAP

BIT:	7	6	5	4	3	2	1	0
Device:	PRINTER		PUNCH		READER		CONSOLE	

SDOS uses a byte (address 0003) for specifying which physical device is assigned to each logical device. This byte is utilized by the internal set command (see 3.1.8.), and the DOS calls 7 and 8 (Section 4). Only the printer and console bits are presently implemented.

2. Communicating with DOS through entry locations in low memory (0-100H):

Ø-2H-provides a re-entry point (return) to CONPROC with the display of the prompt for the currently selected drive.

5-7H-provides the system call entry point. A jump instruction is stored at location 5. The address stored at 6 and 7 is the beginning SDOS address and thus marks the upper limit of user memory. The usual way to make system requests of SDOS, is to CALL location 5.

3. Use of File Control Blocks (FCB's)

FCB-1 will receive the first file-reference, if any, in the command line (after the command).

FCB-2 will receive the second file-reference, if any, in the command line (after the command). These FCB buffers however, are only 16 bytes and FCB-2 will be destroyed if FCB-1 is used and FCB-2 is not moved. Further details on FCB's are given on Page 66.

4.1 SYSTEM CALLS

A system call is made to the operating system to initiate a funcion, usually involving I/O. These functions are divided into the following three (3) sections:

A. I/O functions for all non-disk devices (including the console, printer, etc.)

B. I/O functions for disk files (including creation of, search for, renaming, deleting, opening and closing.)

C. Special purpose functions (including integer multiplication, division, system abort and program links).

A system call is accomplished by loading the C register with the number of the function and loading the entry parameters (if any) into the specified registers. When a CALL 5 instruction is executed SDOS will perform the desired function. When finished, the user program will be returned to with a RET instruction. The user's registers are returned unchanged, except those specified in Table A.

4.1.1. SDOS NON-DISK DEVICE SYSTEM CALLS

The system calls in this section are for non-disk device I/O. The number preceding each SDOS function is the number which must be loaded into the C register prior to the CALL 5 instruction. This number is given in decimal and hexadecimal (in parenthesis). These calls are summarized in Table A (page 77).

1-GET CONSOLE (with echo)

This call retrieves a single character (one byte) from the console. The byte is returned in the A register with the parity bit (Bit 7) reset (zero). SDOS returns control to the user program when a character has been read and echoed back to the CRT. Only the value in the C register is required as an entry parameter.

NOTE: A CTRL-Z (^Z) character is usually considered by a user program as an end of file mark. Most other control characters will not be echoed back to the CRT and some have special meanings for the operating system.

2-PUT CONSOLE

This call writes a single ASCII character (one byte) to the console. The character is placed in the E register before the call. SDOS waits until the console is ready to receive the character and then prints it. No parameters are returned by the call.

If a CTRL-P (^P) is entered while SDOS is outputting characters with this system call, all subsequent characters are sent to the console and the printer until another CTRL-P is entered (thus CTRL-P acts as a toggle switch). CTRL-W (^\) also causes subsequent characters to be sent to the console and the printer but must be encountered in a file to do so. CTRL-T (^]) in a file cancels the effect of either the CTRL-\ or the CTRL-P and causes characters to be sent only to the console. CTRL-\ and CTRL-] may be placed in a file so that when the file is being displayed on the console, the

file can stop and start the printer at the desired places.

CTRL-I is the tab character and is converted to spaces as it is displayed so that the cursor is positioned at one of the standard tab stops (column 1, 9, 17, 25, 33, 41, 49, 57, 65 or 73). The tab, however, is still stored internally in a file as a single ASCII character (Ø9H).

3- GET READER

This call reads one character from a paper tape or card reader. All 8 bits are read and returned in the A register (the parity bit is not masked). No entry parameters are required other than the value in the C register. Since no card or paper tape reader is connected to a standard SD Systems computer, the port assignments and method of interface (default is serial) for this system call are set up initially with the console as a reader.

NOTE: The console status is checked during the read for the CTRL-S (^S) toggle, enabling the user to stop/start the reading process at will. This is particularly useful for pausing during a paper tape jam.

4- PUT PUNCH

This call punches a single character on a paper tape punch. All 8 bits are punched (including parity). The character to be punched is placed in the E register before the call. SDOS waits until the punch is ready to receive the character. No parameters are returned by this call.

Note: The console status is checked during the read for CTRL-S (^S), enabling the user to stop/start the punching process. This is particularly useful for pausing during a paper tape jam.

5-PUT LIST

This call prints a single character (one byte) on the printer. The character to be printed is placed in the E register before the call, and SDOS waits until the printer is ready before returning to a user program. No parameters are returned by this call. Tabs are not expanded.

Note: The console status is checked during the print out for the CTRL-S (^S) character, enabling the user to stop/start the listing. This is useful for stopping the system to load a new box of line printer paper and then restarting the system.

7-GET I/O BYTE

For extra I/O devices, an IOBYTE has been provided. This system call returns the IOBYTE in the A register. The byte format is shown on Page 57. Four consoles can be designated and four each of paper-tape punches, readers and printers.

8-SET I/O BYTE

This call enables the user's program to set the IOBYTE. The E register contains the byte prior to the call. The byte format is shown on Page 57.

9-PRINT BUFFERED LINE

This call will output to the console a string of ASCII characters which end with the dollar sign ($). The DE register pair is loaded with the address of the beginning of the string before the call is made to SDOS. If the printer toggle is on, the line will also be sent to the printer.

10-(0AH)-INPUT BUFFERED LINE

This call reads an input line from the console. The DE register must be pointing to an available buffer before the call is made to SDOS. The first byte of the buffer must contain the maximum length of the buffer. Upon return from this call the second byte of the buffer will contain the actual length entered. The input line will be stored beginning at the third byte. If the buffer is not full, the byte at the end of the line will be zero filled.

When the input line is being entered, the following characters will have a special meaning:

CTRL-C (^C)	Warm boot back to SDOS.
CTRL-E (^E)	Physical CR-LF. The line is not terminated and nothing is entered into the buffer. This character is used to enter a line longer than can be entered on the console.
CTRL-P (^P)	Toggle printer/console link. This is an alternate action software switch which alternately sends all characters to the console and printer and allows the characters to be sent to the console only.

CTRL-R (^R)	Repeats what has been typed on the line.
CTRL-U (^U)	Deletes the entered line and goes back to beginning of buffer for new line.
CTRL-X (^X)	Deletes the current line.
Underscore	Deletes and echos characters. Delimits removed character(s) with //
RUBout, DEL Backspace	Deletes the previous character and backs up the cursor (used for CRT terminals).

11 (ØBH)-TEST CONSOLE READY

The console is tested for a typed character. If a character has been typed, FFH is returned in the A register. If no character has been typed, Ø is returned in the A register.

128 (8ØH)-READ CONSOLE (without echo)

This call is the same as READ CONSOLE (with echo). However, it does not echo the character after it is read. The byte is returned in the A register.

142 (8EH)-SET SPECIAL CRT FUNCTION

This call is used to perform two (2) special functions on the system console terminal. The call is executed by first loading the entry parameters into the DE register pair. No parameters are returned by this call.

Mnemonic	Function	Parameters for Register D	Parameters for Register E
Address	Address cursor on screen	1-8Ø	1-24
Clear	Clear CRT screen	Ø	Ø

When addressing the cursor, the D register should contain the column address (1 through 8Ø) and the E register should contain the row address (1 through 24) of the desired cursor position. The system call does not generate an error if these values are exceeded. On some CRTs addressing the cursor at a non-existent location may cause it to disappear from the screen.

For reference, the cursor location (1,1) is considered to be the upper left-hand corner and the cursor location (8Ø,24) the lower right-hand corner of the screen.

NOTE: Hex values must be entered.

4.1.2. SDOS DISK SYSTEM CALLS

SDOS separates the disk into areas called files. These files are referenced through file control blocks (FCBs). FCBs are 33 bytes long and have the following structure in which each of the numbers below stands for one byte:

FCBDK	Disk descriptor	Ø	(Ø=current disk, 1-drive A, 2-B, 3-C, 4-D
FCBFN	Primary file name	1...8	(right-filled with blanks)
FCBFT	Secondary file name	9...11	(right-filled with blanks)
FCBEX	File entry or extent	12	(initially=Ø; is incremented by 1 in every new entry of 16K bytes)
	Reserved	13...14	
FCBRC	Record count	15	(total number of 128 byte sectors or records)
FCBMP	Cluster allocation map	16...31	(allocated clusters)
FCBNR	Next record	32	(next record to be read or written; has the value Ø through 127)

Note: The directory entries on the disk consist of 32 byte FCBs. The last byte, FCBNR, which points to the next record, is omitted.

13 (ØDH)-RESET SDOS AND SELECT DRIVE A

SDOS is initialized, all disks are logged-off, and drive A is selected as the current drive. The other disks will be logged-on as soon as they are accessed.

14 (ØEH)-SELECT DISK DRIVE

The disk drive number in the E register is selected as the current disk. The drive number in the E register is Ø for drive A, 1 for drive B, 2 for drive C, or 3 for drive D.

15 (ØFH)-OPEN DISK FILE

The file whose FCB is pointed to by the DE register pair, is opened to allow reading or writing. The A register returns with -1 (FFH) if the file is not found, or the directory block number if the file is found. Block numbers start at Ø and there is one block number for every four directory entries. The DE register pair returns pointing to the directory entry in memory.

16 (1ØH)-CLOSE DISK FILE

The file, whose File Control Block is pointed to by the DE register pair, is closed, and the disk directory is updated. (The FCB containing updated cluster information is written to the disk). The A register is returned with -1 (FFH) if the file is not found on the drive, or the directory block number if the file is found on the drive. The file described by the FCB must have been previously opened or created. A file to which bytes have just been written must be closed using this function or the entire last

entry (or extent) will be unreadable (no cluster information) will be present for this entry in the directory).

17 (11H)-SEARCH DIRECTORY FOR FILENAME

The directory is searched for the first occurrence of the file specified in the FCB indicated by the DE register pair. ASCII question mark (? -3FH) in the FCB matches any character. The block number is returned in the A register, if found. If the file is not found, -1 (FFH) is returned in A. HL is returned pointing to the directory entry in memory.

Note: This call will get the directory entry whether it has been erased or not; i.e., these calls do not check to see if a file has been erased. Files are erased by placing a ØE5H in the first byte (FCBDK). The remainder of the FCB is unchanged.

18 (12H) -FIND NEXT DIRECTORY ENTRY

This call is the same as 17 (11H) above, except that it finds the NEXT occurrence of the filename in the directory. This may be either the next extent of a file occupying several extents, or another filename if the question mark (?) is used in the FCB. This call is made after system call 17. Other disk system functions can not be executed between these calls.

19 (13H)-DELETE FILE

The file specified by the FCB pointed to by the DE register pair is deleted from the disk directory. ASCII question mark (?) in the FCB matches any character. The number of directory entries deleted is returned in the A register.

2Ø (14H)- READ NEXT RECORD

The DE register pair points to a successfully opened FCB. The next record (128 bytes) is read into the current disk buffer. The FCBNR in the FCB is incremented to read the next record. One of the following codes is returned in the A register:

Ø - read completed

1 - end of file

2 - read attempted on unwritten cluster (random access files only)

21 (15H)-WRITE NEXT RECORD

The DE register pair points to a successfully opened FCB. The next record is written into the file from the current disk buffer. The FCBNR in the FCB is incremented to be ready to write the next record. One of the following codes is returned in the A register:

Ø - write completed

1 - entry error (attempted to close an unopened entry)

2 - out of disk space

-1 - (or FFH) out of directory space (see Table B)

22 (16H)-CREATE FILE

The file specified in the FCB pointed to by the DE register pair is created on the disk. The A register is returned containing the block number of the directory entry (see ØFH - Open Disk File), or -1 (FFH) if no more directory space is available.

23 (17H)-RENAME FILE

This call renames a disk file. The DE register pair points to the FCB to be renamed. The old primary file name and secondary file name are in the first 16 bytes and the new primary file name and secondary file name are in the second 16 bytes of the FCB. An ASCII question mark (?) in the FCB will match with any character. The A register returns containing the number of directory entries renamed.

24 (18H)-GET DISK LOG-IN VECTOR

The A register is returned specifying the disks that are logged-in. Each bit respresents one disk drive logged-in. If the bit is a one, then it is logged-in; otherwise it is off-line. The least significant bit is the A drive, next most significant (Bit 1) is drive B, etc. Since there are only four drives, the upper four bits are Ø's.

25 (19H)-GET CURRENT DISK

The number of the current disk drive is returned in the A register. Ø = drive A, 1 = drive B, 2 = drive C, 3 = drive D.

26 (1AH)-SET DISK BUFFER

The buffer address in the DE register pair is used for disk I/O. When a program is loaded, the disk buffer is initially located at 8ØH.

27 (1BH)-GET DISK CLUSTER ALLOCATION MAP

The BC register pair returns pointing to a bit map of the allocated clusters on the disk. The DE register pair returns containing the capacity of the current disk (number of clusters). The A register returns containing the number of records or sectors per cluster (8). This call is used by the XSTAT utility program.

131 (83H)-READ LOGICAL BLOCK

This call will read a logical block from the disk without any attention to the files it may contain (no FCB is specified). A block is defined to be one sector or record of 128 bytes. When this function is called, the DE register pair should contain the block number and the B register should contain the disk number (Ø for current drive, 1-4 for A-D.). The high bit of the B register contains a 1 for an interleaved and a Ø for a non-interleaved read. Interleaved means the block which is read is found in the order SDOS stores it (every fourth sector for small disks and every sixth sector for large disks). Non-interleaved means the block which is read is found in sequential order, the order it is physically stored on the disk. The A register is returned with the status of the read according to the following:

Ø - OK

1 - I/O error

2 - illegal request

3 - illegal block

An example will help to illustrate the use of these parameters. SDOS makes use of 4004 sectors on the 8" double sided floppy disks. The block numbers which can be loaded into the DE register are Ø through 4003 decimal, or Ø through FA3H. Suppose that DE is loaded with the value 2 and the B register with Ø (current disk, non-interleaved read). Thus, since the sectors are numbered beginning with 1, sector 3 would be read into memory in the disk buffer (located at 8ØH if it has not been changed). The same read with the B register loaded with 8Ø (current disk, interleaved read) would read sector ØDH (the third sector when they are read every sixth one).

132 (84H) - WRITE LOGICAL BLOCK

This call writes an absolute logical block or sector to the disk without consideration of files (no FCB is specified). The registers are set up and returned in the same way as they are for the Read Logical Block call.

134 (86H) - FORMAT NAME TO FILE CONTROL BLOCK

This call creates a File Control BLock. The HL register pair points to the start of the input line. The DE register points to the place in memory where the FCB is to be built. The line entered is structured:

X:primary filename.secondary filename

where x stands for one of A-D, the primary filename is up to 8 characters with a 3 character secondary name. The FCB is then built from this input line, converting lower case to upper case. The input line is terminated by an ASCII slash (/) or any character

with an ASCII value less than 21H (such as a space or carriage return).

On return the DE register pair points to the start of the new FCB. The HL register pair points to the terminator that ended the build operation.

135 (87H)-UPDATE DIRECTORY ENTRY

The last disk I/O function called must have been a directory call 17 or 18 (Search Directory or Find Next Directory Entry). The DE register pair points to the FCB used in the call 17 or 18. The directory entry is updated on the disk; this means that the entry is written back to the disk without the user having to specify a block. The user specifies a filename when calling 17 or 18. This is useful if it is desired to change a directory entry and write it back to the disk.

139 (8BH)-HOME DRIVE HEAD

The disk drive specified in the B register (Ø for current drive and 1-4 for drives A-D) is sent a command to HOME the head (the disk drive head returns to track Ø).

140 (8CH)-EJECT DISKETTE

This call ejects the disk whose number is given in the E register (Ø for current drive and 1-4 for drives A-D, respectively), only if the disk drive is equipped with the eject option. Otherwise, the call will have no effect.

4.1.3. SPECIAL PURPOSE SYSTEM CALLS

A number of special purpose SDOS system calls have been added for the programmer's convenience. These calls are explained in this section.

Ø - PROGRAM ABORT

This call aborts the current program and returns control to SDOS. This call has the same effect as jumping to location Ø.

129 (81H)-GET USER REGISTER POINTERS

This call provides for expansion of SDOS to a multiprogramming system. The BC register pair returns pointing to UCB (2)USER CONTROL BLOCK.

130 (82H)-SET USER CONTROL-C ABORT

When CTRL-C (^C) is typed, the system normally aborts and returns control to SDOS. This call allows the programmer to change the address to which control is transferred when CTRL-C is typed (i.e., a user may assign a new function to CTRL-C). The address is given in the DE register pair. Note that if DE contains a zero, the system abort is reset. Jumping to location Ø at any time still causes a return to SDOS, also with the CTRL-C being restored to its original function.

136 (88H)-LINK TO NEW PROGRAM

This call allows one command program to call another. The default command-line buffer and default FCB's for the new program must be set up prior to this call if that program expects to be able to use them. The DE register pair should contain the address of the FCB of the new program (which must have a secondary file name of COM). If the new program is NOT found, the A register returns containing -1 (FFH). In this case the first 8Ø bytes (from 1ØØH to 17FH) will be destroyed because this is used in reading the directory. If the program is found, execution begins at 1ØØH and no return is made to the original program.

137 (89H)-MULTIPLY INTEGERS

This call provides a 16 bit multiply. The DE and HL register pairs contain the two 16-bit factors, and the answer is returned in register DE (DE=DE*HL).

138 (8AH)-DIVIDE INTEGERS

This call provides a 16-bit divide. The DE register pair should contain the divisor and the HL register pair, the dividend. The quotient is returned in HL, and the remainder in DE (HL=HL/DE with DE=remainder).

151 (97H)-SET BOTTOM OF SDOS IN RAM

This call sets the bottom address of SDOS to a lower value than the one at which SDOS was originally loaded when it was booted up. The high byte of the new bottom address is placed into the E register prior to execution of the call. The low byte is assumed to be Ø; thus, the bottom of SDOS can never be located on any address other than a 256 byte boundary. If the value is -1 (FFH) or any other value greater than the high byte of the original bottom address when booting up, SDOS will restore the original bottom address. No parameters are returned by the call.

This function changes the system call jump at locations 5, 6, and 7. Programs using the address at locations 6 and 7 to determine the size of the present User Area will find this area to be reduced in size. A second set of jumps (9 bytes) will be loaded at the new bottom of SDOS which points to the old bottom so that system calls will still execute correctly. Note that SDOS is in no way relocated by this function and will reside in the same memory space as it did previously. The purpose of the call is to make it possible to attach a permanent patch space to SDOS for programs which are to become a permanent part of the operating system for as long as it resides in memory. The only way the patch space may be removed is by a second Set Bottom call.

TABLE A

SUMMARY OF SDOS SYSTEM CALLS

The following table lists all the system calls along with their entry and return parameters. The system calls are listed in numerical order, (of the number which is loaded into the C register).

NUMBER	FUNCTION	REGISTERS AND ENTRY PARAMETERS	REGISTERS AND RETURN PARAMETERS
Ø	PROGRAM ABORT	none	none
1	GET CONSOLE (with echo)	none	A-character (parity bit reset)
2	PUT CONSOLE	E-character	none
3	GET READER	none	A-character
4	PUT PUNCH	E-character	none
5	PUT LIST	E-character	none
6	(not used presently - reserved for expansion)		
7	GET I/O BYTE	none	A-I/O byte
8	SET I/O BYTE	E-I/O byte	none
9	PRINT BUFFERED LINE	DE-buffer address	none
10 (ØAH)	INPUT BUFFERED LINE	DE-buffer address	none
11 (ØBH)	TEST CONSOLE READY	none	A- -1 (FFH) if ready A-Ø if not ready
13 (ØDH)	RESET SDOS AND SELECT DRIVE A	none	none
14 (ØEH)	SELECT CURRENT DISK	E-disk drive no.	none
15 (ØFH)	OPEN DISK FILE	DE-FCB address	A-directory block A-1 (FFH) if not found
16 (1ØH)	CLOSE DISK FILE	DE-FCB address	A-directory block A- -1 (FFH) if not found
17 (11H)	SEARCH DIRECTORY FOR FILENAME	DE-FCB address	A-directory block A- -1 (FFH) if not found
18 (12H)	FIND NEXT ENTRY IN DIRECTORY	DE-FCB address	A-directory block A- -1 (FFH) if not found
19 (13H)	DELETE FILE	DE-FCB address	A-number of entries deleted
20 (14H)	READ NEXT RECORD	DE-FCB address	A-Ø if OK A-1 if end of file A-2 if tried to read unwritten records

NUMBER	FUNCTION	REGISTERS AND ENTRY PARAMETERS	REGISTERS AND RETURN PARAMETERS
21 (15H)	WRITE NEXT RECORD	DE-FCB address	A-Ø if OK A-1 if entry error A-2 if out of disk space A--1 (FFH) if out of directory space
22 (16H)	CREATE FILE	DE-FCB address	A-directory block A--1 (FFH) if out of directory space
23 (17H)	RENAME FILE	DE-FCB address	A-number of entries renamed
24 (18H)	GET DISK LOG-IN VECTOR	none	A-those disks currently logged-in
25 (19H)	CURRENT DISK	none	A-disk drive number
26 (1AH)	SET DISK BUFFER	DE-buffer address	none
27 (1BH)	DISK CLUSTER ALLOCATION MAP	none	BC-address of bitmap DE-number of clusters A-sectors/cluster
128 (8ØH)	READ CONSOLE (with no echo)	none	A-1 character
129 (81H)	GET USER REGISTER POINTER	none	BC-pointer to user register pointers
130 (82H)	SET USER CTRL-C EXIT (ABORT)	DE-address of C handler (Ø to reset; -1 to disable)	none
131 (83H)	READ LOGICAL BLOCK	DE=block number B-drive number B top bit-1 if interleaved	A-Ø if OK A-1 if I/O error A-2 if illegal request A-3 if illegal block
132 (84H)	WRITE LOGICAL BLOCK	DE-block number B-drive number B top bit -1 if interleaved	A-Ø if OK A-1 if I/O error A-2 if illegal request A-3 if illegal block
133 (85H)	(not used presently - reserved for expansion)		
134 (86H)	FORMAT NAME TO FILE CONTROL BLOCK	HL-address of string DE-FCB address	HL-address of terminat DE-FCB address
135 (87H)	UPDATE DIRECTORY ENTRY	DE-FCB address	none
136 (88H)	LINK TO PROGRAM	DE-FCB address	A-1 (FFH) if error; else execute at 1ØØH
137 (89H)	MULTIPLY INTEGERS	DE-factor 1 HL-factor 2	DE-product
138 (8AH)	DIVIDE INTEGERS	DE-divisor HL-dividend	BC,HL-quotient DE-remainder

NUMBER	FUNCTION	REGISTERS AND ENTRY PARAMETERS	REGISTERS AND RETURN PARAMETERS
139 (8BH)	HOME DRIVE	B-drive number	none
140 (8CH)	EJECT DISKETTE	E-drive number	none
141 (8DH)	GET VERSION OF OPERATING SYSTEM	none	B-version-number C-version-number
142 (8EH)	SET SPECIAL CRT FUNCTION	D-column address/ special function E-row address/Ø	none
147 (93H)	SET PROGRAM RETURN CODE	E-return code for next program	A-previously set return code
148 (94H)	SET FILE ATTRIBUTES	DE-FCB address B-new attributes	none
149 (95H)	READ DISK LABEL	none	DE-FCB address
151 (97H)	SET BOTTOM OF SDOS IN RAM	E-high byte of address of bottom of SDOS	none

TABLE B - SD SYSTEMS DISK CONFIGURATIONS

CONFIGURATION #	SECTORS/TRK (128 Bytes)	TRACKS/DISK	MAX FILES	MAX DISK SPACE
0	26	77	64	(F3H)X1K→243K
1	26	77X2	128	(F7H)X2K→494K
2	18	35	64	(48H)X1K→72K
3	18	70	64	(96H)X1K→150K
4	50	77	128	(EBH)X2K→470K
5	50	77X2	252	238X4K→952K
6	29	35	64	(77H)X1K→119K
7	29	70	128	(7BH)X2K→7246K

SECTION V

SYSTEM GENERATION

The SDOS configuration disk makes it possible for the user to configure a system for his specific memory and I/O requirements.

Place the SDOS configuration diskette (with the operating system in the system area) in the A drive and boot up. Place the disk which is to receive the operating system in the B drive.

Enter the following:

```
@ MAKSDOS
```

The computer will then perform several functions and request parameters. The following will be displayed on the console.

```
BATCH VERSION 01.00

[A] ERA F1A
    ERA F2B

[A]ERA SDOS.HEX

[A] ERA F11

[A] CRUN MAKF11
CRUN VER 1.02

MAKF11 V1.4
ENTER #K OF RAM: 56
ENTER SIGN-ON MESSAGE (20 CHARACTER MAX):  SBC-200 SERIAL PRINT
ENTER CONSOLE TYPE (0-VDB,1-RS232): 0
ENTER DEFAULT PRINTER TYPE (0-PARALLEL, 1-SERIAL): 1
SBC-100 OR SBC-200 (100/200):  200
ENTER SERIAL I/O BAUD RATE:
150,300,600,1200,2400,4800,9600:  9600
WHAT TYPE SERIAL OUTPUT DRIVER?
1 - INSERT NULLS AT END OF LINE
2 - USE EIA PIN 4 (RTS) FOR PRINTER READY
3 - USE EAI PIN 20 (DTR) FOR PRINTER READY
4 - CRT CONSOLE
ENTER DRIVER SELECTION: 3
NOTE:  MUST CONNECT X11-2 TO X11-3 ON SBC-100 OR SBC-200

DEFINE DISK DRIVE A
0 - SINGLE SIDED
1 - DOUBLE SIDED
ENTER DRIVE TYPE (0,1): 1
```

```
DEFINE DISK DRIVE B
0 - SINGLE SIDED
1 - DOUBLE SIDED
ENTER DRIVE TYPE (0,1): 1

DEFINE DISK DRIVE C
0 - SINGLE SIDED
1 - DOUBLE SIDED
ENTER DRIVE TYPE (0,1): 1

DEFINE DISK DRIVE D
0 - SINGLE SIDED
1 - DOUBLE SIDED
ENTER DRIVE TYPE (0,1): 1
[A]ZASM F1 /K
SD SYSTEMS Z80 ASSEMBLER V3.3

PASS 1 DONE

ERRORS=0000

[A]ZASM F2A /K
SD SYSTEMS Z80 ASSEMBLER V3.3

PASS 1 DONE

ERRORS=0000

[A]ZASM F11 /K
SD SYSTEMS Z80 ASSEMBLER V3.3

PASS 1 DONE

ERRORS=0000

[A]LINK F0,F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11 CU
SD SYSTEMS LINKER V3.1
F0      .OBJ
F1      .OBJ
F2      .OBJ
F2A     .OBJ
F3      .OBJ
F4      .OBJ
F5      .OBJ
F6      .OBJ
F7      .OBJ
F8      .OBJ
F9      .OBJ
F10     .OBJ
F11     .OBJ
UNDEFINED SYMBOLS 00
PASS 2
```

```
F0      .OBJ   ABS    BEG ADDR 0080    END ADDR 00F2
F1      .OBJ   ABS    BEG ADDR C4FF    END ADDR C4FF
F2      .OBJ   REL    BEG ADDR C500    END ADDR CCE8
F2A     .OBJ   REL    BEG ADDR CCE9    END ADDR CCFF
F3      .OBJ   REL    BEG ADDR CD00    END ADDR CF9F
F4      .OBJ   REL    BEG ADDR CFA0    END ADDR D153
F5      .OBJ   REL    BEG ADDR D154    END ADDR D194
F6      .OBJ   REL    BEG ADDR D195    END ADDR D415
F7      .OBJ   REL    BEG ADDR D416    END ADDR D6B1
F8      .OBJ   REL    BEG ADDR D6B2    END ADDR D911
F9      .OBJ   REL    BEG ADDR D912    END ADDR DC98
F10     .OBJ   REL    BEG ADDR DC99    END ADDR DD2D
F11     .OBJ   REL    BEG ADDR DD2E    END ADDR DE77

[A]REN SDOS.HEX=F0.HEX

[A]SDOSGEN * SDOS
SYSGEN REV 00.01
.SYS FILE COMPLETED

[A]
```

The user must now use WRTSYS to copy the new system to disk B:
Enter WRTSYS B:=SDOS.SYS (cr)

NOTES:

1. 56K is the standard configuration of SD SYSTEMS SD-200 computers. You must enter your configuration size.

2. The standard configuration of SD SYSTEMS computers includes a VDB console. However, if you are using an RS-232 terminal you must enter "1" for RS-232.

3. The standard SD SYSTEMS configuration includes either:

   A. The TI-800 printer with parallel and serial capability (serial baud rate is 9600).

   B. The NEC SPINWRITER™ 5100 and 5200 serial baud rate is 1200.

4. See your SYSTEM MANUAL.

January 1, 1980

DISK OPERATING SYSTEM
(SDOS)

ADDENDUM

Some of the utilities have been changed or corrected as follows:

1. @ (BATCH)

   The option switch is a '/' instead of an "=". Use only one '/' for all options used.

NOTE: Recursive batch files must have their work file ($$$$.CMD) located on a disk which is not being written to during the batch process.

2. DSKDUMP

   The operational commands have been increased as follows:

   CR - displays next block (retains interleave if set)
   N - displays next block (retains interleave if set)
   ↑ - displays previous block (retains interleave if set)
   X(N)-gets extent # N (Ø is first entry)
   Q - exits to SDOS (closes file if open)
   ^C - exits to SDOS (closes file if open)

3. XFER

   The following options have been added:

   M - conditional transfer N=No, Y=Yes, R=Copy Rest of Files
   E - does not exit on failure unless this option switch is set

4. MCOPY

   This is a new utility which is described below:

MCOPY (VERSION 1.1)

Copies all files from the disk in the designated source drive (A, B, C or D) to the disk in the designated destination drive (A, B, C or D).

CAUTION: Source drive should not be designated as a destination drive.

COMMAND LINE STRUCTURE

[X:] MCOPY [/YZF] (CR)

Where:

X: Specifies the disk drive location of MCOPY.COM file. This parameter is required only if the .COM file is not located on the currently selected drive (or drive A, if it is not the currently selected drive). Values are A, B, C and D. The SD-200 has only two drives (A&B) and therefore for C and D to be valid values, other drives must be connected to the system.

/YZF The '/' must be entered before the drive selections. Y is the source drive and Z is the destination drive. The optional 'F' after drive selections will format the disk in the destination drive the same as the disk in the source drive before copying to it. An error occurs if only one drive is specified.

When MCOPY (CR) is typed the program will go into the multiple copy mode automatically. The user will be prompted with:

COPY FROM DRIVE (A, B, C or D) or Q to quit

The source drive type is specified, then the user will be prompted with:

COPY TO DRIVE (A, B, C or D)

The destination drive is specified, then the user will be prompted with:

```
INSERT SOURCE DISKETTE IN DRIVE Y:
AND DESTINATION DISKETTE IN DRIVE Z:
AND TYPE
C - TO CONTINUE
F - TO FORMAT DISKETTE IN DRIVE Z:
R - TO RESET DRIVES
Q - TO QUIT AND RETURN TO SDOS
```

Where

Y: is drive previously selected as source (or from)

Z: is drive previously selected as destination (or to)

User types in the option he wishes, or hits 'C' to continue.

If 'F' is typed, the diskette in the drive specified as the destination drive is formatted and the same menu prompt is displayed again.

If 'R' is typed, the user may start over with the new diskette types.

If 'Q' is typed, the program returns to operating system (SDOS).

When copying starts it will display the progress on the CRT as follows:

```
0         1         2         3         4         5         6         7
012345678901234567890123456789012345678901234567890123456789012345678901 2
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^EEEEEEEEEE
DISKETTE COPIED AND VERIFIED
```

At this time 'COPY FROM DRIVE (A, B, C or D) or Q to QUIT' will be displayed. The operator can then resume copying after specifying new selections and inserting a new diskette in the destination and/or source drives.

During copying the user may abort by hitting any key. This returns the prompt:

OPERATOR ABORT

and allows the same selections to be entered or a new task to be started. In the above progress display, the numbers are the track read vertically. The '^' is representative of a successful track copied. The 'E' is representative of an empty track read but not copied. After 10 empty tracks read the task will abort. If a newly formatted diskette is inserted in the source drive, the program reads the first 10 tracks printing an 'E' prompt and aborts without copying to the destination drive. This prevents user from copying an empty diskette to a good diskette in the event that drives are specified improperly. However, the message will print:

DISKETTE COPIED AND VERIFIED

as if data had been copied. At this time user may simply insert proper diskettes, type in new selections, and resume the task.

CAUTION

The format option will only format the diskette in the drive specified as destination drive in menu prompt, so be sure the diskette in the drive showr in the prompt is the one you want formatted, otherwise, a good diskette will be formatted if it is in the drive specified as the destination (and the 'F' option is selected).

ERROR MESSAGES:

Wrong Disk Type Drive [A / B / C: / D] Disk being copied to (destination disk) must be reformatted.

INVALID DRIVE SELECTION - Disk drive selection was specified incorrectly

OPERATOR ABORT

DATA VERIFICATION ERROR DRIVE [X: / Y:]

DISK READ ERROR DRIVE [X: / Y:]

XDIR

Lists, in alphabetical order, in up to four columns of 21 rows, the disk files giving their size (in K bytes).

COMMAND LINE SYNTAX:

XDIR [{Y: / file.ref}] (cr)

Where:

Y: Specifies the disk drive whose directory of files is to be displayed. If a drive is not specified, the directory of the currently selected drive will be displayed.

file.ref Specifies the file(s) for which the size is to be displayed. This file reference may include the replacement characters ? and/or *.

e.g. Entering: XDIR FILE??.* (with files located on a single sided double/density disk) will cause the display of the following heading and files.

SD SYSTEMS EXTENDED DIRECTORY V1.2

FILE1.FAX 2K FILE22.FAX 2K

FILE2.FAX 2K FILE23.FAX 2K

FILE3.FAX 2K FILE24.FAX 2K

FILE4.FAX 2K FILE25.FAX 2K

FILE5.FAX 2K FILE31.FAZ 3K

FILE6.FAX 2K FILE41.FAZ 5K

FILE7.FAX 2K

FILE8.FAX 2K

FILE9.FAX 2K

FILE10.FAX 2K

FILE11.FAX 2K

FILE12.FAX 2K

FILE13.FAX 2K

FILE14.FAX 2K

FILE15.FAX 2K

FILE16.FAX 2K

FILE17.FAX 2K

FILE18.FAX 2K

FILE19.FAX 2K

FILE20.FAX 2K

FILE21.FAX 2K

27 entries listed, 58K disk space used.

NOTE: If there are more than 84 files on a disk the following message will appear:

Press RETURN to continue:

Entering XDIR FILE41.FAZ will cause the following to be displayed on the console:

```
SD SYSTEMS EXTENDED DIRECTORY V1.2
FILE41   FAZ   5K
1 entry listed, 5K disk space used.
```

Entering a CTRL S while the directory is being displayed will cause the display to pause until another key is pressed.

The size parameter denotes the rounded up number of K (1024 bytes) actually written on the diskette. However, the number of K allocated for that file may be different as indicated by XSTAT. Minimum file allocations for different diskette formats are as shown below:

TYPE	MIN FILE ALLOCATION
Single side Single density	1K
Double side Single density	2K
Single side Double density	2K
Double side Double density	4K

The SDOS configuration program (MAKSDOS) now responds differently. The following illustration shows the procedure for configurating a 56K system for the SD-200 using the serial port for the printer. default @ 9600 Baud. This system would be treated as single sided on an SD-100:

```
@ MAKSDOS
BATCH VERSION 01.00

[A]ERA F1A

[A]ERA SDOS.HEX

[A]ERA F11

[A]CRUNII MAKF11

CRUN VER 2.05
MAKF11 V1.3
ENTER #K OF RAM:  56
ENTER SIGN-ON MESSAGE (20 CHARACTERS MAX):  SERIAL PRINT SD200
ENTER CONSOLE TYPE (0-VDB,1-RS232):  0
ENTER DEFAULT PRINTER TYPE (0-PARALLEL, 1-SERIAL):  1
2.0,2.5,OR 4.0 MHZ (2.0/2.5/4.0):  4.0
ENTER SERIAL I/O BAUD RATE:
150,300,600,1200,2400,4800,9600:  9600
WHAT TYPE SERIAL OUTPUT DRIVER?
1 - INSERT NULLS AT END OF LINE
2 - USE EIA PIN 4 (RTS) FOR PRINTER READY
3 - USE EIA PIN 20 (DTR) FOR PRINTER READY
4 - CRT CONSOLE
ENTER DRIVER SELECTION:  (3)
NOTE:  MUST CONNECT X11-2 TO X11-3 ON SBC-100 OR SBC-200


DEFINE DISK DRIVE A
0 - SINGLE SIDED
1 - DOUBLE SIDED
ENTER DRIVE TYPE (0,1):  0


DEFINE DISK DRIVE B
0 - SINGLE SIDED
1 - DOUBLE SIDED
ENTER DRIVE TYPE (0,1):  0


DEFINE DISK DRIVE C
0 -SINGLE SIDED
1 - DOUBLE SIDED
ENTER DRIVE TYPE (0,1):  0
```

```
DEFINE DISK DRIVE D
0 -SINGLE SIDED
1 -DOUBLE SIDED
ENTER DRIVE TYPE (0,1):  0


[A]ZASM F1 /K
SD SYSTEM Z80 ASSEMBLER V3.3


PASS 1 DONE


ERRORS=0000


[A]ZASM F2A /K
SD SYSTEMS Z80 ASSEMBLER V3.3


PASS 1 DONE


ERRORS=0000


[A]ZASM F11 /K
SD SYSTEMS Z80 ASSEMBLER V3.3


PASS 1 DONE


ERRORS=0000


[A]LINK F0,F1,F2,F2A,F3,F4,F5,F6,F7,F8,F9,F10,F11 /CU
SD SYSTEMS LINKER V3.1
F0      .OBJ
F1      .OBJ
F2      .OBJ
F2A     .OBJ
F3      .OBJ
F4      .OBJ
F5      .OBJ
F6      .OBJ
F7      .OBJ
F8      .OBJ
F9      .OBJ
F10     .OBJ
F11     .OBJ
UNDEFINED SYMBOLS 00
PASS 2
```

```
F0      .OBJ    ABS     BEG ADDR 0080    END ADDR 00F2
F1      .OBJ    ABS     BEG ADDR C4FF    END ADDR C4FF
F2      .OBJ    REL     BEG ADDR C500    END ADDR CCE8
F2A     .OBJ    REL     BEG ADDR CCE9    END ADDR CCFF
F3      .OBJ    REL     BEG ADDR CD00    END ADDR CF9F
F4      .OBJ    REL     BEG ADDR CFA0    END ADDR D153
F5      .OBJ    REL     BEG ADDR D154    END ADDR D194
F6      .OBJ    REL     BEG ADDR D195    END ADDR D415
F7      .OBJ    REL     BEG ADDR D416    END ADDR D6AF
F8      .OBJ    REL     BEG ADDR D6B0    END ADDR D90F
F9      .OBJ    REL     BEG ADDR D910    END ADDR DC4E
F10     .OBJ    REL     BEG ADDR DC4F    END ADDR DCE3
F11     .OBJ    REL     BEG ADDR DCE4    END ADDR DE2E

[A]REN SDOS.HEX=F0.HEX


[A]SDOSGEN * SDOS
SYSGEN REV 00.01
.SYS FILE COMPLETED

[A] WRTSYS B: SDOS.SYS    (CP/M-SDOS NOTES)
```

```
@ MAKSDOS
SD SYSTEMS BATCH V1.2

[A]ERA F1A

[A]ERA SDOS.HEX

[A]ERA F11

[A]CRUNII MAKF11

CRUN VER 2.05
MAKF11 V1.3
ENTER #K OF RAM:  56
ENTER SIGN-ON MESSAGE (20 CHARACTERS MAX): 9600 RTS=20 4/26
ENTER CONSOLE TYPE (0-VDB,1-RS232):  0
ENTER DEFAULT PRINTER TYPE (0-PARALLEL,1-SERIAL):  1
2.0,2.5,OR 4.0 MHZ (2.0/2.5/4.0):  4
ENTER SERIAL I/O BAUD RATE:
150,300,600,1200,2400,4800,9600:  9600
WHAT TYPE SERIAL OUTPUT DRIVER?
1 - INSERT NULLS AT END OF LINE
2 - USE EIA PIN 4  (RTS) FOR PRINTER READY
3 - USE EIA PIN 20 (DTR) FOR PRINTER READY
4 - CRT CONSOLE
ENTER DRIVER SELECTION:  3
NOTE: MUST CONNECT X11-2 TO X11-3 ON SBC-100 OR SBC-200


DEFINE DISK DRIVE A
0 - SINGLE SIDED
1 - DOUBLE SIDED
ENTER DRIVE TYPE (0,1):  1

DEFINE DISK DRIVE B
0 - SINGLE SIDED
1 - DOUBLE SIDED
ENTER DRIVE TYPE (0,1):  1

DEFINE DISK DRIVE C
0 - SINGLE SIDED
1 - DOUBLE SIDED
ENTER DRIVE TYPE (0,1):  1

DEFINE DISK DRIVE D
0 - SINGLE SIDED
1 - DOUBLE SIDED
ENTER DRIVE TYPE (0,1):  1

[A]ZASM F1 /K
SD SYSTEMS Z80 ASSEMBLER V3.3

PASS 1 DONE

ERRORS=0000

[A]ZASM F2A /K
SD SYSTEMS Z80 ASSEMBLER V3.3

PASS 1 DONE
```

```
ERRORS=0000

[A]ZASM F11 /K
SD SYSTEMS Z80 ASSEMBLER V3.3

PASS 1 DONE

ERRORS=0000

[A]LINK F0,F1,F2,F2A,F3,F4,F5,F6,F7,F8,F9,F10,F11 /CU
SD SYSTEMS LINKER V3.1
F0        .OBJ
F1        .OBJ
F2        .OBJ
F2A       .OBJ
F3        .OBJ
F4        .OBJ
F5        .OBJ
F6        .OBJ
F7        .OBJ
F8        .OBJ
F9        .OBJ
F10       .OBJ
F11       .OBJ
UNDEFINED SYMBOLS 00
PASS 2
F0        .OBJ    ABS     BEG ADDR 0080    END ADDR 00F2
F1        .OBJ    ABS     BEG ADDR C4FF    END ADDR C4FF
F2        .OBJ    REL     BEG ADDR C500    END ADDR CCE8
F2A       .OBJ    REL     BEG ADDR CCE9    END ADDR CCFF
F3        .OBJ    REL     BEG ADDR CD00    END ADDR CF9F
F4        .OBJ    REL     BEG ADDR CFA0    END ADDR D153
F5        .OBJ    REL     BEG ADDR D154    END ADDR D194
F6        .OBJ    REL     BEG ADDR D195    END ADDR D415
F7        .OBJ    REL     BEG ADDR D416    END ADDR D6AF
F8        .OBJ    REL     BEG ADDR D6B0    END ADDR D90F
F9        .OBJ    REL     BEG ADDR D910    END ADDR DC4E
F10       .OBJ    REL     BEG ADDR DC4F    END ADDR DCE3
F11       .OBJ    REL     BEG ADDR DCE4    END ADDR DE2E

[A]REN SDOS.HEX=F0.HEX

[A]SDOSGEN * SDOS
SYSGEN REV 00.01
.SYS FILE COMPLETED

[A]WRTSYS B:=SDOS.SYS
SD SYSTEMS WRTSYS V1.8
WRTSYS DONE & VERIFIED
[A]
[A]
```

TECHNICAL BULLETIN NO. 101

MARCH 1980

SDOS RELEASE 1.5 AND 1.6

Release 1.6 of SDOS for the SD-100 and SD-200 is available for immediate distribution. Included in this revision of SDOS are necessary modifications for the efficient functioning of the program and several enhancements to the program. If you wish to receive either the SDOS version 1.6 documentation or the software, please fill out and return the enclosed form.

CHANGES IN RELEASE 1.5

- Syntax change to the BATCH command.
- Additional option on the DSKDMP utility.
- Additional option on the XFER utility.
- MCOPY — new utility allows copying all files on a disk to another disk.
- XDIR — new utility lists disk files and their sizes.
- MAKSDOS configuration utility responds differently.

CHANGES IN RELEASE 1.6

- DIR COMMAND — Displays correct file size for files larger than 256K.
- PIP properly handles multi-file copy commands.
- MCOPY utility operable in 32K system.
- Parallel printer driver operates in all memory configurations.
- DSKDIAG works properly with double sided disks.

SD SYSTEMS

A SUBSIDIARY OF THE MICHIGAN GENERAL CORPORATION

TECHNICAL BULLETIN NO. 102

MARCH 1980

SD-100 AND SD-200

SD-100 and SD-200 systems are originally shipped with a pair of Mod 4 MFE or Mod 5 MFE drives. Field usage has determined that system problems can occur when a Mod 4 MFE drive is mixed with a Mod 5 MFE drive. This is a potential problem only if a system has required a MFE drive replacement and that MFE drive Mod replacement does not match the original. **If a SD-100/200 contains mixed drives due to drive replacement, log on failures can occur.**

TO IDENTIFY PROBLEM:

Check the MFE drives for uniformity.

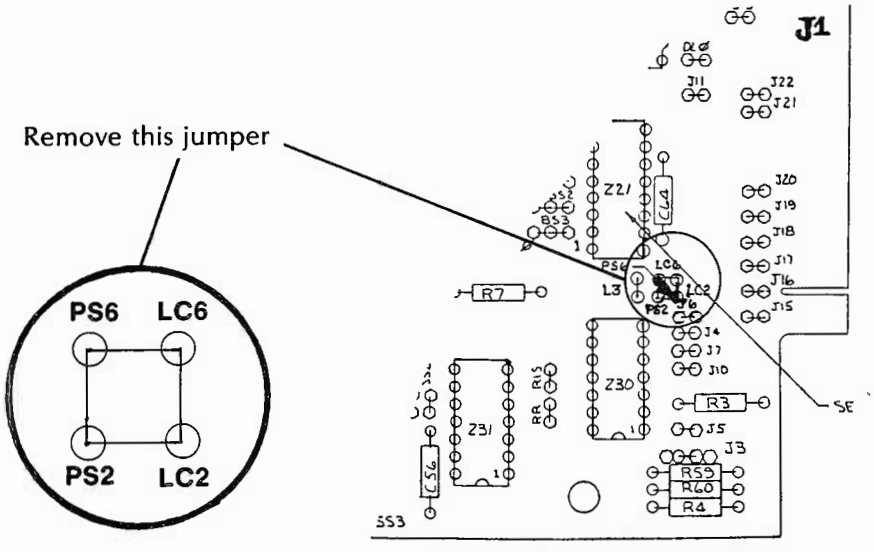
Mod 4 chassis is smooth, obviously machine casted and has no seam.

Mod 5 chassis is a darker cast and has a seam edge.

TO MODIFY SYSTEM:

(1.) Cut straps PS6 and LC2 on Mod 5 drives, as this solves the problem which can occur due to an option strap on this drive between PS2 and PS6 near J1. Note diagram below.

(2.) If assistance is required, call our Technical Assistance hot line number: 1-800-527-4121.



SD SYSTEMS
A SUBSIDIARY OF THE MICHIGAN GENERAL CORPORATION

TECHNICAL BULLETIN NO. 103

MARCH 1980

SD-200

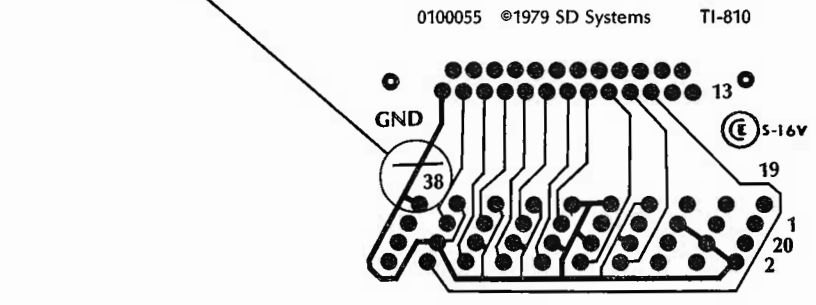
The following modification corrects a problem encountered when a TI-810 printer adapter cable is attached to the SD-200. Field usage has determined that occasional extra characters are printed due to the need for an additional ground.

TO IDENTIFY PROBLEM:

All TI-810 printer adapter cables require the following modification.

TO MODIFY SYSTEM:

Install jumper between GND run from pin 1 to pin 38 on the TI-810 parallel adapter cable. Note diagram below.



SD SYSTEMS
A SUBSIDIARY OF THE MICHIGAN GENERAL CORPORATION

TECHNICAL BULLETIN: No. 104

APRIL 1980

C-BASIC APPLICATION NOTE

When C-Basic reads or writes a record to or from a random file where the record being accessed does not already exist and is discontiguous from previously existing records, it will create an extent for that record. This extent will be seen as a separate entry for the same file under the DIR command.

For example, if records 1 thru 100 exist on a random file and a program erroneously writes record 50000, a separate extent will be created to handle that record. The extent that is created will show up as a separate entry to the same file under the DIR command.

SD SYSTEMS
A SUBSIDIARY OF THE MICHIGAN GENERAL CORPORATION