

**SD SYSTEMS
COMMUNICATION ORIENTED
MULTI-USER OPERATING SYSTEM
(COSMOS)
INSTRUCTIONAL PUBLICATION
VERSION 2.17**

**SD #7140075 REV. B
JUNE, 1981**

**SD
SYSTEMS**
P.O. Box 28810, Dallas, Texas 75228

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
SECTION I GENERAL DESCRIPTION		
1.0	<u>GENERAL INFORMATION</u>	1-1
SECTION II COSMOS SYSTEM ORGANIZATION		
2.1	<u>DISK ALLOCATION</u>	2-1
2.2	<u>MEMORY ALLOCATION</u>	2-1
2.2.1	RESIDENT COSMOS.....	2-1
2.2.2	USER VARIABLES & I/O BUFFER.....	2-2
2.2.3	USER MEMORY.....	2-2
2.2.4	LOW MEMORY.....	2-2
2.3	<u>FILES AND I/O DEVICES</u>	2-3
2.3.1	FILE NAMES.....	2-3
2.3.1	FILE REFERENCES (TWO TYPES).....	2-4
2.3.2	I/O DEVICE NAMES.....	2-5
SECTION III COSMOS OPERATION		
3.1	<u>INTRODUCTION</u>	3-1
3.2	<u>MOUNT/DISMOUNT PROCEDURE</u>	3-1
3.2.1	MOUNT/DISMOUNT WARNING.....	3-1
3.2.2	DISK FILE RECOVERY SUGGESTIONS.....	3-1
3.2.3	SUBDIRECTORY CONSIDERATIONS.....	3-2
3.3	<u>GLOBAL FILE CONSIDERATIONS</u>	3-2
3.4	<u>STARTUP</u>	3-7
3.5	<u>DISK DRIVE SELECTION</u>	3-7
3.6	<u>CHANGING DISKS AFTER COLD BOOT</u>	3-8
3.7	<u>CONSOLE CONTROL CHARACTERS</u>	3-8
3.7.1	CONSOLE CONTROL CHARACTERS.....	3-8
3.7.2	PRINTER CONTROL MESSAGES.....	3-9
3.7.3	PRINTER ERROR MESSAGES.....	3-10
3.7.4	DISK DRIVER ERROR MESSAGE AND RESPONSES.....	3-10
SECTION IV COSMOS CONSOLE PROCESSOR COMMANDS		
4.1	<u>COMMAND STRUCTURE AND SYNTAX</u>	4-1
4.2	<u>INTERNAL COMMANDS</u>	4-2
4.2.1	ATTRIB.....	4-3
4.2.2	ATTACH.....	4-4
4.2.3	DIR.....	4-4
4.2.4	DM.....	4-5
4.2.5	DSTAT.....	4-6
4.2.6	EOP.....	4-6
4.2.7	ERA.....	4-6
4.2.8	GET.....	4-7

TABLE OF CONTENTS (CONTINUED)

PARAGRAPH	TITLE	PAGE
4.2.9	IDENTIFY.....	4-7
4.2.10	MONITOR.....	4-8
4.2.11	MO.....	4-8
4.2.12	MSG.....	4-8
4.2.13	REN.....	4-9
4.2.14	SAVE.....	4-10
4.2.15	SET.....	4-10
4.2.16	START.....	4-11
4.2.17	STOP.....	4-11
4.2.18	TYPE.....	4-12
4.2.19	USER.....	4-12
4.3	<u>UTILITY COMMANDS</u>	4-12
4.3.1	@ (BATCH).....	4-14
4.3.2	CBASICII, CRUNII, AND XREF.....	4-16
4.3.3	CLOCK AND ECLOCK.....	4-16
4.3.4	DIRMOVE.....	4-17
4.3.5	DSKDIAG.....	4-17
4.3.6	DSKDUMP.....	4-18
4.3.7	DTYPE.....	4-20
4.3.8	DUMP.....	4-21
4.3.9	EDIT.....	4-21
4.3.10	ERASE.....	4-21
4.3.11	FCLUSTER.....	4-22
4.3.12	FORMAT.....	4-22
4.3.13	CDCOPY/WDCOPY.....	4-24
4.3.14	CDINIT/WDINIT.....	4-24
4.3.15	CDTEST/WDTEST.....	4-25
4.3.16	LINK.....	4-26
4.3.17	LOAD.....	4-27
4.3.18	MCOPY (VERSION 2.0).....	4-27
4.3.19	MOVE.....	4-30
4.3.20	SPOOL.....	4-31
4.3.21	WRTCOS.....	4-31
4.3.22	XDIR.....	4-32
4.3.23	XFER.....	4-34
4.3.24	XSTAT.....	4-36
4.3.25	ZASM.....	4-36

SECTION V
PROGRAMMERS GUIDE

5.1	<u>INTRODUCTION</u>	5-1
5.2	<u>SYSTEM CALLS</u>	5-2
5.3	<u>COSMOS NON-DISK DEVICE SYSTEM CALLS</u>	5-2
5.4	<u>COSMOS DISK SYSTEM CALLS</u>	5-7
5.5	<u>SPECIAL PURPOSE SYSTEM CALLS</u>	5-15
5.6	<u>SPECIAL COSMOS CALLS</u>	5-16
5.7	<u>NON-CP/M SYSTEM CALLS</u>	5-20

TABLE OF CONTENTS (CONTINUED)

PARAGRAPH	TITLE	PAGE
SECTION VI		
COSMOS INSTALLATION AND CONFIGURATION		
6.1	<u>INTRODUCTION</u>	6-1
6.2	<u>HARDWARE CONFIGURATION</u>	6-1
6.2.1	MEMORY SYSTEM CONFIGURATION.....	6-1
6.2.2	TWO AND FIVE USER CONFIGURATION.....	6-2
6.2.2.1	<u>Two User Configuration</u>	6-2
6.2.2.2	<u>Five User Configuration</u>	6-3
6.2.3	JUMPER OPTION CONFIGURATION.....	6-4
6.2.3.1	<u>SBC-200 Configuration</u>	6-5
6.2.3.2	<u>VERSAFLOPPY II Configuration</u>	6-5
6.2.3.3	<u>EXPANDORAM II Configuration</u>	6-5
6.2.3.4	<u>VDB-8024 Configuration</u>	6-5
6.3	<u>SOFTWARE CONFIGURATION</u>	6-6
6.3.1	BOOTING THE COSMOS DISTRIBUTION DISKETTE.....	6-6
6.3.2	COSMOS CONFIGURATION PROCEDURE.....	6-6
6.3.2.1	<u>System Disk Definition</u>	6-6
6.3.2.2	<u>Sign-On Message</u>	6-6
6.3.2.3	<u>Partition Size</u>	6-6
6.3.2.4	<u>Default Printer Type</u>	6-7
6.3.2.5	<u>SBC Serial I/O Baud Rate Selection</u>	6-7
6.3.2.6	<u>System Type Definition</u>	6-7
6.3.2.7	<u>MPC-4 Channel Baud Rate Selection</u>	6-7
6.3.2.8	<u>Printer Driver Alterations</u>	6-7
6.3.2.9	<u>Using The Configured System</u>	6-7
6.3.2.10	<u>Booting The Configured COSMOS</u>	6-8
APPENDIX A	SD SYSTEM DISK CONFIGURATION.....	A-1
APPENDIX B	COSMOS CONFIGURATION EXAMPLE.....	B-1

LIST OF TABLES

TABLE#	TITLE	PAGE
2-1	User Memory Page Low Memory Description.....	2-3
2-2	COSMOS I/O Device Mnemonics.....	2-5
3-1	Console Control Characters.....	3-9
3-2	Printer Control Characters.....	3-9
4-1	Internal Commands Summary.....	4-2
4-2	Minimum File Allocation.....	4-5
4-3	Utility Command Summary.....	4-13
4-4	File Allocations.....	4-34
5-1	Detailed Low Memory Utilization Map.....	5-2
5-2	I/O Bytes Description.....	5-2
5-3	File Control Blocks Bytes Description.....	5-7
5-4	E Register Drive Numbers.....	5-8
5-5	Register A Error Codes.....	5-18
5-6	COSMOS Call Summary.....	5-21
6-1	ExpandoRAM II Dip Switch (S3) Settings.....	6-5
6-2	Disk Numbers Used for Boot-Up.....	6-8

LIST OF ILLUSTRATIONS

FIGURE NUMBER	TITLE	PAGE
2-1	COSMOS Memory Configuration.....	2-2
6-1	192K Memory System For Five 32K Users.....	6-1
6-2	256K Memory System For Five 48K Users.....	6-2
6-3	Parallel Interface.....	6-3
6-4	MPC-4 RS-232 Ports Interface.....	6-4
6-5	SD SYSTEMS Microcomputer Connection.....	6-4

SECTION I

GENERAL DESCRIPTION

1.1 GENERAL INFORMATION

COSMOS is a multi-user operating system which allows each of up to eight users to run independent jobs concurrently. Time is allocated to all active users on an equal priority basis. When a user's job is held up by an I/O device or a section of semaphored code, no more time is allocated to that job until the wait condition is gone. Programs written to run under SD-OS or CP/M* will also run under COSMOS.

COSMOS provides each user with a separate file directory with two basic file types supported, local and global. Local files may only be accessed by the user in whose directory the file exists. Global files may be accessed by any user and are especially useful for system utilities and common data base access.

*Trademark of Digital Research

SECTION II

COSMOS SYSTEM ORGANIZATION

2.1 DISK ALLOCATION

Disks (media) used under COSMOS contain four basic areas: COSMOS system image, file directory, BAD.MAP and user file area.

The COSMOS system image area starts on track 0 of the disk and is recorded in non-interleaved format on the first section of the disk.

The file directory contains entries for all users who have files on the disk. The directory size depends on the disk media being used. The disk allocation bitmap is on disk for hard disk. Each platter has its own bitmap, called MAP.DIR. CDINIT/WDINIT has an option to (re-) initialize this file.

Subdirectory (really multi-directory) capability is implemented for Hard Disk Platters, with the following properties:

- * Up to 9 directories are allowed on each platter
- * Each user independently selects for each platter which subdirectory to activate.
- * BAD.MAP is used to pre-allocate bad clusters to avoid their usage by normal files.

The user file area takes up all the remaining space on the disk.

2.2 MEMORY ALLOCATION

COSMOS is a page-mode multi-user operating system. Page 0 is 64K with the top 16K containing the operating system. Each user is allocated a 32K or 48K user memory page, the majority is available for use by the application software.

The console processor (Operator Interface to O.S.) is not always memory resident. It is reloaded at the end of each user program's execution and remains in memory during the execution of all Intrinsic Functions. It is reloaded from the BOOT DISK on the occurrence of a user program doing a JP 0 or LD C,0.
CALL 5.

Figure 2-1 illustrates the memory configuration of a COSMOS system.

64K	RESIDENT COSMOS		16K
31/47K	USER 0 VARIABLES & I/O BUFFER	USER N VARIABLES & I/O BUFFER	
0K	USER 0 or CONPROC MEMORY OVERLAY (31K or 47K)	USER N or CONPROC MEMORY OVERLAY (31K or 47K)	48K 32K
	LOW MEMORY	LOW MEMORY	
	PAGE 0	PAGES 1-N	

Figure 2-1. COSMOS Memory Configuration

2.2.1 RESIDENT COSMOS

Resident COSMOS contains the disk operating system, floppy disk allocation maps, non-disk device drivers and disk device drivers. Note that only page 0 contains a copy of resident COSMOS. Most of this code is fully re-entrant which means that all active users are running in the same physical memory during system calls made by application programs or the console processor.

2.2.2 USER VARIABLES AND I/O BUFFER

The top bytes of the user's 48K memory page is used by COSMOS to store all user-specific information. This includes CPU register values, stacks and a disk I/O buffer for disks using sectors larger than 128 bytes.

2.2.3 USER MEMORY

Approximately 47K of the 48K user memory page is available for use by the application software being run by that user.

2.2.4 LOW MEMORY

The low memory (0-FFH) in each user memory page is assigned in the same way as SDOS and CP/M*.

Table 2-1. User Memory Page Low Memory Description

<u>HEX ADDRESS</u>	<u>DESCRIPTION</u>
0	WARM RESTART ENTRY ADDRESS ON USER PROGRAM EXIT
5	DOS ENTRY ADDRESS FOR SYSTEM CALLS
6-7	LOWEST MEMORY ADDRESS USED BY COSMOS
8-3F	RESERVED FOR RST INTERRUPT VECTORS
38	ILLEGAL ADDRESS TRAP (RST 7)
40-5B	SYSTEM VARIABLES
5C-7F	STANDARD FCB'S
80-FF	STANDARD I/O BUFFER

2.3 FILES AND I/O DEVICES

There are two basic types of files supported by COSMOS, local and global. A user's directory contains all local files for that user and all files with the global attribute on the disk. A user may give a file the global attribute (paragraph 4.2.1) only if the file name does not exist in any other user's directory. This is checked by the 'ATTRIB' command and the user is notified if the file cannot be globalized.

COSMOS files may be accessed either sequentially or randomly (section 5).

2.3.1 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). The file name 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.

*Trademark of Digital Research

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 - *SD Systems relocatable, linkable hex object

If when operating within the console processor (section 4) an executive command file (.COM) is referenced to without specifying a disk drive for the file, the current drive is searched first. If not found (and the current drive is not the "system" drive) the "system" drive is then searched for the file. The "system" drive is specified during the system generation procedure (sec. 6). For floppy disk based system it is usually drive A, for fixed/removable, it may be assigned to E-J.

2.3.2 FILE REFERENCES (TWO TYPES)

1. The first type of file reference is called a specific (individual) file reference. This 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 file reference and contains either a ? or an * seen by the system as "don't care" specifies. This is used in certain CONPROC functions and utility programs to refer to a group of files.

The ? specifies to the directory search routine that any character in that position is acceptable when comparing. The "*" 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 '.', with a '?'.

For example, if a disk contained 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 are found.

*ASCII - American Standard Code for Information Interchange

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.

2.3.3 I/O DEVICE NAMES

When referring to the I/O devices under COSMOS, use the mnemonics listed in table 2-2.

Table 2-2. COSMOS I/O Device Mnemonics

<u>DEVICE</u>	<u>NAME</u>	<u>DATA TRANSFER</u>
Console	CON:	Input & Output
Card Reader	RDR:	Input
Paper-tape punch	PUN:	Output
Line Printer	LST:	Output
Floppy Disk	A:,B:,C:,D:	Input & Output
Removable Disk Cartridge	E:	Input & Output
Fixed Disk Platters	F:,G:,H:,I:,J:	Input & Output

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

AAA:

where AAA is a three character name.

The device names listed above are those used by COSMOS, however, only the console, printer and disk are active when shipped from SD Systems.

SECTION III

COSMOS OPERATION

3.1 INTRODUCTION

COSMOS maintains system compatibility with SD-OS and CP/M*. However, due to the nature of a multi-user system it is important to consider that when more than one user is operating on shared facilities, no single user can change the shared environment without considering the effect on the other users. This shared environment includes the disks, printers and other devices connected to the system.

3.2 MOUNT/DISMOUNT PROCEDURE

When a user wishes to mount a disk (diskette or disk cartridge) in the system, user 0 (integral console) must issue a MOUNT (MO) command, which checks to determine if a disk is already mounted in the drive. If there is one, a DISMOUNT (DM) command must be issued (paragraph 4.2.4). If any users are logged on that disk, an error message prints and the system does not allow the dismount operation. The 'DSTAT' command (paragraph 4.2.5) displays the currently logged on users for each mounted disk. All users must log onto another disk and enter control-C (^ C) in order to allow the DISMOUNT to occur. The MOUNT may then take place. The actual MOUNT operation reads the disk directory and creates an allocation map in memory. This map is used for any allocation and de-allocation of disk space.

3.2.1 MOUNT/DISMOUNT WARNING

It is imperative that the MOUNT/DISMOUNT procedure be observed in order to maintain disk file integrity. Failure to observe this procedure causes destruction of disk files.

Control -C (^C) does not read in the directory and create the allocation map and is not a substitute for the MOUNT/DISMOUNT procedure. Door locks on the floppy disk drives are utilized to prevent inadvertent removal of media. The doors are only "unlocked" after a successful DISMOUNT or a 'STOP *' command.

3.2.2 DISK FILE RECOVERY SUGGESTIONS

- A. Backups should be taken on a periodic basis.
- B. If suspicious of problems on a currently mounted disk, run XSTAT.

* Trademark of Digital Research

- C. If unable to mount a disk because of BAD CLUSTER # message, use DSKDUMP to put an E5 in the first byte of directory entry.
- D. Re-run CDINIT BITMAP initialization on offending platters to bring the bitmap back into sync with cluster allocation.
- E. If the disk is unrecoverable, CDINIT (NOTE: HD requires a label and bitmap initialization in order to be useable).
- F. The message: CHECKSUM ERROR IN BITMAP DISK X: indicates a problem that requires re-boot and application of the above recovery procedures.
- G. The search option of DSKDUMP may be used to manually locate portions of lost data files.

3.2.3 SUBDIRECTORY CONSIDERATIONS

Up to 8 additional subdirectories may be provided per platter. Each subdirectory occupies 16K of disk storage. Each has the standard 254 entries.

The following operational rules apply:

1. In running on a platter, each user independently selects an active subdirectory via the ATTACH intrinsic. When STARTed and when a disk platter is MOUNTed, a user is initially using subdirectory 0 (the master directory) on the platter. Only one subdirectory per platter per user is available at any one time.
2. Most system calls and utilities accessing a directory (usually via FCB parameters) are implicitly referring to the current subdirectory thereof. The only exceptions are CDINIT, WDINIT, XSTAT, DIRMOVE, FCLUSTER, ATTACH, and IDENTIFY.
3. Moving files from one subdirectory to another on the same platter is done one at a time via DIRMOVE. An alternative is to use 2 ambiguous XFERs through an intermediate platter, with an ATTACH in between.
4. The only means to change subdirectories is via the Conproc intrinsic ATTACH or via an ATTACH system call.

3.3 GLOBAL FILE CONSIDERATIONS

There are two primary reasons for using global files under COSMOS: system utilities and common access to a data base.

Global system utilities may be run by all users without having separate copies in each directory. There is no difference in running these programs between SD-OS and COSMOS.

However, access and modification of a global data base file should receive special consideration by the applications programmer.

A shared data base file must be created and fully allocated or extended before multi-user access. The application software should create the file and write as many sequential blank records needed by the data base. Thus, disk space for the data base is fully allocated and subsequent multi-user access does not cause any new disk space to be allocated to the file.

A standard text editor should not be used on a global file because files with this attribute cannot be erased or renamed.

In shared data base access applications where more than one user is to update the file at the same time, the COSMOS record locking system should be used (see paragraph 5.6). To make high level language access to this facility easy, an entry is provided which performs the appropriate system calls. The following BASIC program is an example of how this is done:

*** C-BASIC II RECORD LOCKING EXAMPLE ***

```

REM*****EXAMPLE OF RECORD LOCKING USAGE*****
REM
REM FNSYLK% IS THE BARE BONES SYSTEM CALL
REM FNLKMSG% PRINTS APPROPRIATE MESSAGE FOR LOCK ERRORS
REM FNLOCK% IS A FILTER WHICH INTERPRETS AND ACTS
REM      ON ERROR CODES BASED ON A PARTICULAR
REM      SET OF APPLICATION REQUIREMENTS
REM THE REST IS A MINIMAL FILE ACCESS PROGRAM
REM
REM*****

```

```

DEF FNSYLK%(OPCODE%,FILENO%,DISKNO%,RSIZE%,RECNO)
REM ***STEP 1: POINT TO LOCK PARMS
LOKADDR  = FLOAT(PEEK 7))*256.0 + FLOAT(PEEK(6)) + 9.0
LOKPADDR = LOKADDR + 3
REM ***STEP 2: BREAK UP LOCK PARMS INTO 1 BYTE PIECES
RSIZE1%  = RSIZE% / 256
RSIZE2%  = RSIZE% - RSIZE1% * 256
RBYTE1%  = INT%(RECNO / 256)
REMAIN   = RECNO - RBYTE1% * 256
RBYTE2%  = INT%(REMAIN / 256)
REMAIN   = REMAIN - RBYTE2% * 256
RBYTE3%  = INT%(REMAIN)
REM ***STEP 3: INITIALIZE LOCK PARMS
POKE LOKPADDR+0,OPCODE%
POKE LOKPADDR+1,FILENO%
POKE LOKPADDR+2,DISKNO%
POKE LOKPADDR+3,RSIZE1%
POKE LOKPADDR+4,RSIZE2%
POKE LOKPADDR+5,RBYTE1%
POKE LOKPADDR+6,RBYTE2%
POKE LOKPADDR+7,RBYTE3%
POKE LOKPADDR+8,0
REM ***STEP 4: CALL RECORD LOCKING
CALL LOKADDR
REM ***STEP 5: EXAMINE RESULTS
RETCODE% = PEEK(LOKPADDR+8)
IF RETCODE% > 9 THEN RETCODE% = RETCODE% - 256
FNSYLK% = RETCODE%
RETURN
FEND

```

```

DEF FNLKMSG%(RETCODE%)
IF RETCODE% = 0 THEN PRINT "***CALL COMPLETED"
IF RETCODE% = -1 THEN PRINT "***UNKNOWN LOCK OPCODE"
IF RETCODE% = -2 THEN PRINT "***LOCK TABLE FULL"
IF RETCODE% = -3 THEN PRINT "***INVALID PARMS"
IF RETCODE% = -4 THEN PRINT "***ALREADY PRESENT"
IF RETCODE% = -5 THEN PRINT "***FILE NOT IN TABLE"
IF RETCODE% = -6 THEN PRINT "***NOT ACTIVE ON INDICATED FILE(S)"
IF RETCODE% > 0 THEN PRINT "***SORRY-LOCKED BY USER#",RETCODE% -1
IF RETCODE% < -6 THEN PRINT "***UNKNOWN ERROR CODE:",RETCODE%
FNLKMSG% = 0

```

```
RETURN
FEND
```

```
DEF FNLOCK%(OPCODE%,FILENO%,DISKNO%,RSIZE%,RECNO,LOOPCNT)
REM *** ASSUMPTION IS THAT ALL ERRORS ARE CONSIDERED
REM     FATAL EXCEPT THE FOLLOWING:
REM     INIT : ALREADY PRESENT IS O.K.
REM     LOCK : LOCKED BY OTHER USER--NOTIFY USER AT
REM           TICK INTERVALS
REM     FLUSH: IGNORE ALL ERRORS EXCEPT -3
05  LOKCNT = 1
10  LOOPER = 1
20  RET% = FNSYLK%(OPCODE%,FILENO%,DISKNO%,RSIZE%,RECNO)
30  IF RET% = 0 THEN GOTO 100
40  IF (OPCODE%=3)AND(RET% NE -3) THEN GOTO 100
50  IF (OPCODE% = 1)AND(RET%=-4) THEN GOTO 100
60  IF OPCODE% NE 2 THEN GOTO 200
70  IF (FILENO% = 0)OR(RECNO=0) THEN GOTO 100
72  IF RET% < 0 THEN GOTO 200
80  LOOPER = LOOPER + 1
90  IF LOOPER < 10.0 THEN GOTO 20
92  LOKCNT = LOKCNT + 1
93  IF LOKCNT <= LOOPCNT THEN GOTO 10
95  WAIT% = FNLKMSG%(RET%)
96  GOTO 05
100 FNLOCK% = RET%
110 RETURN
200 FATAL% = FNLKMSG%(RET%)
210 PRINT "***FATAL ERROR:ABORTING***"
220 STOP
230 FEND
```

```
REM *****SAMPLE PROGRAM IN C-BASIC 2*****
```

```
500 REM ***SELECT COMMAND
PRINT "WHICH CMD TO EXECUTE?"
PRINT " 1 = INIT"
PRINT " 2 = GAIN ACCESS TO RECORD"
PRINT " 3 = RELEASE RECORD"
PRINT " 4 = FLUSH FILE"
PRINT " 5 = SELECT FILE"
PRINT " 6 = EXIT PROGRAM"
INPUT "YOUR SELECTION:";CMD%
IF CMD% = 1 THEN 1000
IF CMD% = 2 THEN 2000
IF CMD% = 3 THEN 3000
IF CMD% = 4 THEN 4000
IF CMD% = 5 THEN 5000
IF CMD% = 6 THEN 6000
PRINT "***BAD SELECTION--TRY AGAIN"
GOTO 500
1000 REM *** INIT
DSKN% = ASC(FINAME$) - ASC("A") + 1
RETLK% = FNLOCK%(1,FIL%,DSKN%,RSIZ%,0,0)
OPEN FINAME$ RECL RSIZ% AS FIL%
```



```

GOTO 500
2000 REM *** GAIN ACCESS TO A RECORD
      INPUT "RECORD#:";REC
      RETLK% = FNLOCK%(2,FIL%,0,0,REC,THRESH)
      READ # FIL%,REC; VALU$
      INPUT "DISPLAY OR MODIFY(D/M):";C$
      IF C$ = "D" THEN 2200
      INPUT "NEW VALUE;";VALU$
      PRINT # FIL%,REC; VALU$
      GOTO 500
2200 PRINT VALU$
      GOTO 500
3000 REM *** RELEASE RECORD
      REM NOTE THAT LOCKING NEXT RECORD WOULD DO IT ALSO
      RETLK% = FNLOCK%(2,FIL%,0,0,0,0)
      GOTO 500
4000 REM *** FLUSH USER FROM ACTIVE LIST ON FILE
      REM      POSSIBLY REMOVING FILE FROM TABLE
      RETLK% = FNLOCK%(3,FIL%,0,0,0,0)
      CLOSE FIL%
      GOTO 500
5000 REM *** SELECT A FILE
      INPUT "DISK AND FILE NAME:";FINAME$
      INPUT "DESIRED FILE#";FIL%
      INPUT "RECORD SIZE:";RSIZ%
      INPUT "WAIT THRESHOLD:";THRESH
      GOTO 500
6000 REM *** EXIT PROGRAM
      PRINT "NORMAL TERMINATION"
      STOP

```

3.4 STARTUP

See Section 6 for installation instructions.

Turn on the SD computer and the terminals. Place a disk with the COSMOS System in the "system drive"* and press the RESET switch once.

The computer responds with a prompt "." Enter a "C" (upper case) and a carriage RETURN (CR). This boots up the system, which is now ready [A:0].

This procedure is called a cold boot and loads the operating system into memory. Upon completion of the cold boot the "system" drive is the currently selected drive and the disk in the "system" drive is automatically mounted.

If a batch command file named STARTUP.CMD is on the system disk it is automatically executed. The commands stored in this file are run sequentially (see Batch Utility Command-paragraph 4.3.1). If for example, when creating the STARTUP.CMD file, a password request program is referenced as the first command, a request for a password is displayed as shown below.

```
[A:0] @ STARTUP
      BATCH VERSION 01.00
[A:0] SIGNON
      ENTER PASSWORD
```

This program is written to prevent unauthorized access to the system.

3.5 DISK DRIVE SELECTION (when in the console processor)

Once a disk is mounted (paragraph 3.2), a disk drive other than the currently selected drive may be selected by entering the drive name (A,B,....J) followed by a colon (:) and carriage RETURN key (cr).

For example: If the currently selected disk drive is A and access to disk drive B is required; enter the following: B: (cr). The console displays [B:0] to indicate that drive B is now the currently selected disk drive. If drive B is not mounted, the following message displays:

```
"ATTEMPTING TO ACCESS UNMOUNTED DISK B:"
```

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

*On floppy disks shipped from the factory, disk A: is the system disk.

3.6 CHANGING DISKS AFTER COLD BOOT

Before a disk (diskette or disk cartridge) may be removed from a drive, user 0 (integral console) must issue a DISMOUNT command (see paragraph 3.2). The new disk may then be MOUNTED to read the disk allocation map for that disk, into memory for subsequent disk allocation and de-allocation.

The BOOT DRIVE or PLATTER must always contain a (not necessarily mounted) disk with the operating system, SAME REV. AND SIZE, as boot. This is required to reload the CONPROC OVERLAY upon return from a user program. COSMOS prompts for insertion of an appropriate diskette if necessary. A corollary restriction is that with a floppy boot disk, only C or D format diskettes can be in the boot drive, except for a short time to do an MCOPY or DTYPE at the system console. Also, do not WRTCOS to the boot disk.

3.7 CONSOLE CONTROL CHARACTERS

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

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).

3.7.1 CONSOLE CONTROL CHARACTERS

Once the system is running, basic user interaction is with the console. The CONsole PROCcessor (CONPROC) program supports file access by means of the INTERNAL commands and the utility programs described in Section 4. 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. Table 3-1 provides a list of control characters and their functions.

Table 3-1. Console Control Characters

CHARACTER	FUNCTION
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 input/output. 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 (CTR only). Moves cursor to the left margin without a linefeed.
^X	Deletes current line (goes to next line).

3.7.2 PRINTER CONTROL CHARACTERS

There are four control characters used to control output to the printer described in table 3-2.

Table 3-2. Printer Control Characters

CHARACTER	FUNCTION
^P	Send all console output to the printer and console. This is a toggle switch action. If the printer is off, it is turned on by entering CTRL-P. After a CTRL-P is entered, entering another CTRL-P cancels the first one.
^\ (lCH)	Turns off all the output to the printer. This control character can be output by a user program but has no effect if issued from the console.*
^I (lDH)	Send all output to the printer and console. This control character can be output by a user program but has no effect if issued from the console.*

*These characters are not sent to VDB-8024 under SD-OS.

Table 3-2. Printer Control Characters (Continued)

CHARACTER	FUNCTION
^Z (1AH)	Release printer for other users. All reports using the printer(s) should send a ^Z (1AH) to the printer at the end. This causes a form feed to occur and releases the printer for the other users. The EOP (paragraph 4.2.6) command in the console processor may be used to manually release the printer.

3.7.3 PRINTER ERROR MESSAGES

PRINTER NOT READY - The printer is not 'on-line'.

PRINTER IN USE - The printer is being used by another user.

Note that ^S^C can be used to abort in these situations.

3.7.4 DISK DRIVER ERROR MESSAGE AND RESPONSES

ERROR MESSAGE FORMAT:

DISK ERROR xx yy-Z:BLOCKaaaTRACKbbbSECTORccccdddd

Where:

- xx = disk driver command
- yy = disk driver error
- z = disk drive
- aaa specifies block location of error
- bbb specifies track location of error
- ccc specifies sector location of error
- dddd defines a descriptive error message

Response:

- R representing retry
- ^C abort program
- (cr) to ignore or act as though no error occurred.

- xx = 0 representing home
- 1 representing lifthead
- 2 representing mount
- 3 representing dismount
- 4 representing read physical
- 5 representing read interleaved
- 6 representing write physical
- 7 representing write interleaved
- 8 representing free buffer
- 9 representing dump buffer
- 10 representing log in disk

yy = 0 representing no error
1 representing read error
2 representing illegal command
3 representing illegal block
4 representing software write protect

z = A-J for drive identification

dddd = SOFTWARE WRITE PROTECT (only on CP/M Operating System)
WRITE PROTECT
NOT READY
READ ERROR
HARDWARE ERRORS
ILLEGAL PARAMETER - SOFTWARE ERROR
(This error indicates COSMOS or Driver bug)
BAD BLOCK
ILLEGAL DISK NBR
(This causes the SD SYSTEMS Microcomputer to halt)

SECTION IV

COSMOS CONSOLE PROCESSOR COMMANDS

4.1 COMMAND STRUCTURE AND SYNTAX

The Command (.COM) files can be executed anytime the system prompt is displayed by entering the primary file name of the command without the secondary file name (.COM is assumed), and the remaining elements of the command line. The console processor then determines if it is an internal command. If it is not an internal command, the currently logged-in disk drive (and the "system" drive if it is not the currently logged-in drive) is searched until the system finds the utility or user command file. When the program is found it is loaded into memory starting at 100H, with the remainder of the command line passed to the program as control information. Execution of the command starts at 100H. If the command file is not found the following message is displayed on the console: PROGRAM NOT FOUND.

Each command line can start with an optional disk drive specifier. The command (.COM assumed) is then entered. The remaining content of the command line is determined by the individual command. The following conventions are observed in the structuring of the command lines:

1. Any combination of upper and lower case characters may be entered in a command line. They are automatically translated by the system into upper case characters.
2. The REN, WRTCOS, CDCOPY and 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 are used to separate file names when concatenating files.
4. Options are preceded by a space and a slash [/].

NOTE

In explaining the structure of command lines the following symbols and combinations are been used.

- [] Brackets are used to indicate an optional parameter. 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.
- [{ }] Brackets enclosing braces indicates an optional choice of items.

- Character underline is used to show which characters must be entered by the user.

4.2 INTERNAL COMMANDS

The Internal Commands reside within COSMOS once the system has been loaded. (See table 4-1).

Table 4-1. Internal Commands Summary

NAME	DESCRIPTION	PARAGRAPH
AT <u>R</u> IB	SET FILE ATTRIBUTES	4.2.1
ATTACH	ATTACH SUBDIRECTORY	4.2.2
DIR	DISPLAY DISK FILE DIRECTORY	4.2.3
DM*	DISMOUNT DISK	4.2.4
DSTAT	DISPLAY USER STATUS ON MOUNTED DRIVE	4.2.5
EOP	RELEASE PRINTER	4.2.6
ERA	ERASE FILE(S)	4.2.7
GET	LOAD HEX OBJECT FILE	4.2.8
IDENTIFY	IDENTIFY SUBDIRECTORY	4.2.9
MO*	MOUNT DISK	4.2.10
MON*	GO TO SD MONITOR	4.2.11
MSG	SEND MESSAGE TO OTHER USER(S)	4.2.12
REN	RENAME FILE(S)	4.2.13
SAVE	CREATE FILE FROM MEMORY IMAGE	4.2.14
SET	SELECT PRINTER (PARALLEL OR SERIAL)	4.2.15
START*	START USER	4.2.16
STOP*	STOP USER(S)	4.2.17
TYPE	DISPLAY FILE	4.2.18
USER*	SELECT USER'S DIRECTORY	4.2.19

*Only operative from user 0.

4.2.1 ATRIB

Enables the user to set, change or delete file attributes which protect a file from being written to or erased, or make it global.

COMMAND LINE SYNTAX:

```
ATTRIB file.ref [+ ] [p...] (cr)
```

Where:

file.ref	is the file (which may be ambiguous) being referenced.
+	This parameter is required only when assigning additional protective parameters to a file.
[p...]	are the protective parameters assigned to the file.
P	protects a file from being erased, renamed, or written.
G	Global attribute erase and rename protects file and makes it accessible to all users. This operation does not occur if the file exists in any users directory other than the one issuing the command. (Error - 'CONFLICT' EXISTS') Note that if an ambiguous reference is made which matches in another directory, none of the files are made global.

Attributes already assigned to a file are deleted by assigning new attributes without using the + sign. All attributes assigned to a file are 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    P
FILE22.FAX  5K    1    PG
FILE44.FAZ  3K    1
```

To make FILE11.FAX global simply enter:
ATTRIB FILE11.FAX +G (cr)

To remove protection from FILE22.FAX simply enter:
ATTRIB FILE22.FAX (cr)

To assign protection to FILE44.FAZ simply enter:
ATTRIB FILE44.FAX P (cr)

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

FILE11.FAX	2K	1	GP
FILE22.FAX	5K	1	
FILE44.FAZ	3K	1	P

4.2.2 ATTACH

COMMAND LINE SYNTAX:

ATTACH X:Y

Where X = specific disk drive E-J
Y = subdir# 0-8

Subdirectory Y on disk X is made active for this user. If the disk is not mounted or the subdirectory does not exist, ILLEGAL ATTACH is displayed.

4.2.3 DIR

Lists the disk files giving their size (in K bytes), number of extents and the file attributes. All files in the users directory and all global files are listed.

COMMAND LINE SYNTAX:

[Y:]
DIR [file.ref] (cr)

Where:

Y:	Specifies the disk drive whose directory of file(s) to display. If a drive is not specified, the directory of the currently selected drive is displayed.
file.ref	Specifies the file(s) size, number of extents and attributes to display. This file reference may include the "don't care" characters ? and/or *.

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

<u>NAME</u>	<u>EXT</u>	<u>SIZE</u>	<u>EXTENTS</u>	<u>ATTRIBUTES</u>	<u>USER #</u>
FILE11	FAX	1K	1	G	0
FILE21	FAX	1K	1	P	0
FILE31	FAX	1K	1		0
FILE41	FAZ	1K	1		0
Totals					
4 Files		4K	4		

Entering: DIR FILE41.FAZ causes the following to display on the console.

<u>NAME</u>	<u>EXT</u>	<u>SIZE</u>	<u>EXTENTS</u>	<u>ATTRIBUTES</u>	<u>USER #</u>
FILE41	FAZ	1K	1		0
Totals					
1 File		1K	1		

NOTE

Pressing any key during the displaying of the directory causes it to abort and display the totals for the files displayed up to that point.

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

The size parameter denotes the round 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 given in table 4-2.

Table 4-2. Minimum File Allocations

<u>TYPE</u>	<u>MIN FILE ALLOCATION</u>
Single side Single density	1K
Double side Single density	2K
Single side Double density	2K
Double side Double density	4K
Hard Disk	1K

4.2.4 DM

The DISMOUNT command may only be issued by user 0. Refer to paragraph 3.2 and 3.2.1 for further information on the dismount procedures.

COMMAND LINE SYNTAX:

DM Y: (cr)

Where Y: is a disk device name (A-J)

The user log-in table is checked, and if the specified disk is logged on by any users the following message is displayed:

"CANNOT DISMOUNT, DISK IN USE"

Otherwise, the specified disk is flagged as dismounted in the mounted disk table, and another disk may then be MOUNTED in the drive. Successful execution of the 'DISMOUNT' command unlocks the door of the specified drive (if floppy disk).

4.2.5 DSTAT

COMMAND LINE SYNTAX:

```
DSTAT (cr)
```

The DSTAT command displays the user log-on status of all MOUNTED disks.

For example, if disks A, B and F are mounted:

```
A: 0 1 3
B: 0 2
F:
```

This would indicate that disk A is being used by users 0, 1, and 3. Disk B is being used by users 0 and 2. Disk F is mounted but not currently in use.

In order to dismount a disk, there cannot be any users currently using that disk.

4.2.6 EOP

COMMAND LINE SYNTAX:

```
EOP (cr)
```

Once a printer listing is complete, a ^Z (1AH) must be sent to release the printer to other users. If the application program does not send a ^Z, the EOP command is used to release the printer.

4.2.7 ERA

ERA is an internal command to delete the file(s) from the user's disk directory.

COMMAND LINE SYNTAX:

```
ERA file.ref (cr)
```

Where:

file.ref	Specifies the file(s) to erase from the disk directory. This file reference may include the "don't care" characters ? and/or *. The disk space previously used by the erased files is then available.
----------	---

NOTE

Caution is recommended when "don't care" characters are used, as it is possible to delete a large number of files with one entry.

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

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

4.2.8 GET

GET is an internal command that enables the user to load into memory (and optionally run) any .HEX absolute object file. The file is loaded into the address specified by the Intel hex object format.

COMMAND LINE SYNTAX:

```
GET file.ref [ /G] (cr)
```

Where:

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

4.2.9 IDENTIFY

IDENTIFY is an internal command used to list which subdirectory is currently active on a given disk.

COMMAND LINE SYNTAX:

IDENTIFY X:

Where X = disk drive E-J

Disk X must be mounted. Console processor responds with the message NOT MOUNTED or SUBDIR# : XX indicating which subdirectory is active.

4.2.10 MONITOR

To re-enter the SD Monitor depress the 'RESET' switch. Care should be taken not to depress 'RESET' while other users are active.

4.2.11 MO

MO is an internal command to enable user 0 to mount a disk. The MO command may only be issued by user 0. Refer to paragraph 3.2 and 3.2.1 for further information on the mount procedure.

COMMAND LINE SYNTAX:

MO Y: (cr)

Where:

Y: is a disk device name (A-J)

If the specified disk is already flagged as mounted, the following message is displayed:

"CANNOT MOUNT, ALREADY MOUNTED"

Otherwise, the specified disk directory is read and an allocation map is created in memory for subsequent use in allocation or de-allocation of disk space.

If a disk error is encountered during the mount operation, one of the following messages is displayed:

"CANNOT MOUNT, BAD DISK"
"BAD CLUSTER #XXXX"

4.2.12 MSG

MSG is an internal command to provide a means of inter-user message communications.

COMMAND LINE SYNTAX:

MSG {n,*} message text (cr)

where n is a user number from 0-7

Options: a. n - send message text to user n.
 b. * - send message text to all active users.

If the specified user(s) is not active, the following message is displayed:

USER NOT ACTIVE

Otherwise the message text is displayed on the specified user's console and that user must respond by entering a (cr) to resume processing.

If a previous message sent to a user is not displayed, the following message is displayed on the sending user's console:

PREVIOUS MSG PENDING, TRY AGAIN (Y/N)?

A response of 'Y' causes another attempt at sending the message to the same user. If a previous message is still pending after several tries, it probably means that the specified user is running an applications program which does not make system calls. Should the users application program return to CONPROC, the pending message would then be displayed.

4.2.13 REN

REN is an internal command to enable the user to rename a file (either the primary name, secondary name or both).

COMMAND LINE SYNTAX:

REN file.ref1=file.ref2 (cr)

Where:

file.ref1 Is the new file reference. This file reference may include the "don't care" characters ? and/or *. NOTE: All characters from the primary and secondary file names (referred to by the old file.ref) are substitutes for the ambiguous characters of the new file reference.

file.ref2 Is the old file reference. This file reference may include the "don't care" characters ? and/or *.

NOTE

Initially, the user's directory 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 because it limits access to the first file directory name.

e.g. With the following files

```
FILE.ABC
FI12.FAX
FI13.FAX
FI14.FAX
```

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

```
FILE.ABC
SAME.TAG
SAME.TAG
SAME.TAG
```

4.2.14 SAVE

SAVE is an internal command to save the specified number of 256 byte pages of the user area starting at address 100H.

COMMAND LINE SYNTAX:

SAVE file.ref N (cr)

Where:

file.ref Specifies the file name to give to the contents saved from the user area. If the secondary file name is entered as ".COM" the file is saved as a command file.

N Is the number (decimal) of pages to save.

4.2.15 SET

SET is an internal command that provides a means of re-assigning the listing channel to the non-default device.

COMMAND LINE SYNTAX:


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

4.2.16 START

START is an internal command to enable terminals (1-7). The START command may only be issued by user 0.

COMMAND LINE SYNTAX:

```
      {*      }
START N [{COMMAND LINE}] (cr)
```

Where:

N is a user number from 1-7.

NOTE

Command Line Option - start up the specified user in the console processor with the specified command line to use first before prompting the console.

* - Restart a previous halted job using the STOP command.

No option specified - start up the specified user in the console processor.

4.2.17 STOP

STOP is an internal command to disable terminals (1-7).

COMMAND LINE SYNTAX:

```
      {*      }
STOP {n} (cr)
```

Where:

N is a user number from 1-7.

N - stops the specified user's job. Any user may stop his own job. User 0 may stop any user's job except 0.

* - Stops all users (0-7) jobs. This may only be issued by user 0. All floppy disk doors are unlocked and the following message is displayed:

"SYSTEM STOPPED, ENTER 'RETURN' TO RESTART:"

4.2.18 TYPE

TYPE is an internal command to display the contents of the ASCII file referenced on the console.

COMMAND LINE SYNTAX:

TYPE file.ref (cr)

Where:

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

NOTE

Pressing any key terminates the command. The display pauses when the CTRL-S is entered. Pressing any other key resumes the display.

4.2.19 USER

The USER command may only be issued by user 0.

COMMAND LINE SYNTAX:

USER N (cr)

Where:

N is a decimal user number from 0-7.

The user command allows user 0 to examine the contents of any user's directory.

4.3 UTILITY COMMANDS

Utility Commands are resident on the disk as .COM files and are loaded into the user area as needed. Execution of Utility Programs alters the user area. (See table 4-2)

Table 4-3. Utility Command Summary

NAME	DESCRIPTION	PARAGRAPH
@	BATCH PROCESSOR	4.3.1
CBASICII	CBASICII COMPILER	4.3.2
CLOCK	SET/READ TIME CLOCK (US)	4.3.3
CRUNII	CBASICII RUN-TIME INTERPRETER	4.3.2
DIRMOVE	COPY FILE TO ANOTHER SUBDIRECTORY	4.3.4
DSKDIAG*	FLOPPY DISK DIAGNOSTIC	4.3.5
DSKDUMP	FILE DISPLAY/MODIFY (HEXADECIMAL)	4.3.6
DTYPE*	DISPLAY FORMAT OF DISKETTES	4.3.7
DUMP	HEXADECIMAL FILE DUMP	4.3.8
ECLOCK	SET/READ TIME CLOCK (EUROPEAN)	4.3.3
EDIT	TEXT EDITOR	4.3.9
ERASE	CONDITIONAL ERASE	4.3.10
FCLUSTER	FIND ALL FILES WITH A GIVEN CLUSTER	4.3.11
FORMAT*	FORMAT DISKETTES AND CARTRIDGES	4.3.12
CDCOPY/WDCOPY*	HARD DISK COPY	4.3.13
CDINIT/WDINIT*	INITIALIZE HARD DISK	4.3.14
CDTEST/WDTEST*	HARD DISK TEST	4.3.15
LINK	LINK OBJECT MODULES	4.3.16
LOAD	CONVERT HEX OBJECT FILE TO BINARY	4.3.17
MCOPY*	FLOPPY DISK COPY	4.3.18
MOVE	COPY LOCAL FILE TO OTHER USER'S DIRECTORY	4.3.19
SPOOL	PRINT FILE IN BACKGROUND	4.3.20
WRTCOS	COPY OPERATING SYSTEM	4.3.21
XDIR	ALPHABETICALLY SORTED DIRECTORY	4.3.22
XFER	COPY FILES	4.3.23
XREF	CBASICII CROSS REFERENCE LISTER	4.3.2
XSTAT	DISK STATUS	4.3.24
ZASM	Z-80 GLOBAL ASSEMBLER	4.3.25

* Only operative from user 0, with no other users started.

4.3.1 @ (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 where 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 where a permanent file is created (using the Text Editor) and whose commands can be executed at any time following a system prompt.

COMMAND LINE SYNTAX:

ONE TIME EXECUTION:

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

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, the "system" drive. Values are A, B,...J.
R	Is an optional nesting function switch which enables a batch command (.CMD) file to call another batch command (.CMD) file and return to the first .CMD file when execution of the second .CMD file is completed. If this switch is not set the system does not return to the first file. The level of nesting is equal to the minimum number of commands pending at one time (128).
Y	Specifies where the disk drive on which the batch work file (\$\$\$\$.CMD) is located. Defaults to the 'system' drive.
file.ref	Defines the batch command file (CMD) to execute.
P1,P2...9	Provides optional parameters to pass to the .CMD file upon entering the @ (BATCH) Command line.
(CR)	Carriage RETURN.

@ (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 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 COSMOS prompt ([A]) are valid. e.g.

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

ONE TIME EXECUTION MODE

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 pass to the file commands from the @ (BATCH) Command line, an up carat (^) followed by the number of the parameter (e.g. ^1).

```
e.g.  EDIT  SAMPLE.CMD
      TYPE ^0   (CR)
      DIR  ^1   (CR)
      REN  ^2   (CR)
      TYPE ^3   (CR)
      TYPE ^1   (CR)
      @  ^4   (CR)
      Q      (CR)  ends editing
```

If an ^0 is entered after a file command, upon execution of the batch file, the batch file itself is referenced. e.g. The command, TYPE ^0, in the example above causes the contents of the above file (SAMPLE.CMD) to display 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 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 a parameter. Complex parameters (ones which include character space + = 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 above the following takes place.

The system displays on the console the contents of SAMPLE.CMD
followed by:

```
DIR FILE.CMD
```

(displaying the directory listing along with the size)

Followed by:

```
REN PROGRAM.CMD = BLOCK.CMD
TYPE ITEMS.CMD      (ITEMS.CMD is displayed)
TYPE FILE.CMD       (FILE.CMD is displayed)
@ RECORDS.CMD       (This batch file is processed)
```

NOTE

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
physical write protected diskette, or the batch file disk being
full (either space or directory).

4.3.2 CBASICII, CRUNII AND XREF

See "CBASICII" manual for descriptions.

4.3.3 CLOCK AND ECLOCK

COMMAND LINE SYNTAX:

[X:] <u>CLOCK</u> [mm/dd/yy hh:mm:ss] (cr) <u>US FORMAT</u>	
[X:] <u>ECLOCK</u> [dd/mm/yy hh:mm:ss] (cr) <u>EUROPEAN FORMAT</u>	
Where:	
mm is month (1-12)	hh is hour (0-23)
dd is day (1-31)	mm is minute (0-59)
yy is year (last 2 digits)	ss is second (0-59)

The clock utility is used to set and read the real time clock.
If the optional date and time parameters are entered, the real
time clock is set to the specified values.

If no parameters are entered, the clock is read and the date and time are displayed.

NOTE

The CLOCK and ECLOCK utilities are only functional on systems containing an MPC-4 board (See Section 6).

4.3.4 DIRMOVE

COMMAND LINE SYNTAX:

DIRMOVE [X:]Y N

Where X optional disk drive E-J
Y filename with extent
N receiving subdir#

The file Y is copied within the same disk from the current subdirectory to subdir# N. Only global files and files under the current user# are accessible. The current user# and filename are retained in the receiving directory. Error messages are essentially as for the MOVE utility.

4.3.5 DSKDIAG

Used with the SD-100/200 to test the floppy disk drives for proper operation.

COMMAND LINE SYNTAX:

[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 the "system" drive. Values are A-J.

Entry of the command line causes 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)
- FF00 - Exits the program to location specified by user
- [.] - Exits back to monitor
- (cr) - Exits back to COSMOS.

Where:

T 1-5, 10, 11 or FF
 DD (or XX) is the drive specification as follows:

	#SIDES SIZE	1		2		1		2		1		2	
		FULL	FULL	MINI	MINI	FULL	FULL	MINI	MINI	FULL	FULL	MINI	MINI
DRIVE DENSITY	SECTOR SIZE	SNGL	SNGL	SNGL	SNGL	DBL	DBL	DBL	DBL	DBL	DBL	DBL	DBL
A		128	128	128	128	128	128	128	128	128	128	256	256
B		00	10	20	30	40	50	60	70	C0	D0		
C		01	11	21	31	41	51	61	71	C1	D1		
D		02	12	22	32	42	52	62	72	C2	D2		
		03	13	23	33	43	53	63	73	C3	D3		

4.3.6 DSKDUMP

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

COMMAND LINE SYNTAX:

[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 the "system" drive. Values are A-J.
- Y Specifies the disk drive whose disk is to be dumped.

file.ref	If this field is not entered, then disk [Y] is accessed by or interleaved block addresses. If a file name is entered then the block read designates the block number of that file rather than the absolute block number.
E	Extent (16K block) of file to access (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) is 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, is read and displayed; rather than the next physical block on the disk. Pressing any key stops the display.

If one of the command letters (W, M, N, S, X, Q, or ^) are entered instead of the hex value, the corresponding operation is performed.

MODIFY

MA, value [,value,...value] (cr)

A is the first hex location in the block to change.

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), locations 20H-24H of the buffer block are changed to 20H, 23H, 41H, 42H and 43H.

SHOW

S (cr) When the character S is entered followed by the carriage RETURN (cr) the buffer block, including any modifications, are displayed.

WRITE

W (cr)

WXXXX (cr)

WXXXXI (cr)

Where XXXX is the block number
I specifies an interleaved write

W-writes the buffer block to the read disk origin.

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

W35I-writes the buffer block to block 35 interleaved (absolute block number if no file was entered)

NEXT

N Advances to next block. (cr) has same effect.

CHANGE EXTENT

Xn Change extents where n is the extent to be opened.

QUIT

Q (cr) Return to console processor. ^C has same effect.

LIST MATCHES

Lxxx,valuelist (cr)
Lxxx,/startbyte,stopbyte/,valuelist (cr)

This searches in a forward direction in the active file extent or disk for xxx occurrences of a block with valuelist present at positions between startbyte and stopbyte. Here:

xxx =
occurrences to find and list
valuelist =
same as in Modify option
startbyte, stopbyte =
start, stop positions each in the range 0-7FH. Default is /0,127/

All numbers are given in hex.

4.3.7 DTYPE

DTYPE is a utility command to display the system and drive configuration.

COMMAND LINE SYNTAX:

[X:] <u>DTYPE</u> (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 the "system" drive. Values are A-J.

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)

4.3.8 DUMP

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

COMMAND LINE SYNTAX:

```
[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 the "system" drive. Values are A-J.

file.ref Specifies the file to dump.

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.

4.3.9 EDIT

Enables the user to create and modify text files. See the Text Editor, Z-80 Global Assembler, Linker Manual for details.

4.3.10 ERASE

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

COMMAND LINE SYNTAX:

[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 the "system" drive. Values are A-J.

file.ref Specifies the ambiguous file(s) to erase.

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

CONDITIONAL ERASE VERSION 1.1
FILE.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 are listed.

4.3.11 FCLUSTER

COMMAND LINE SYNTAX:

FCLUSTER X: N (cr)

Where X = disk drive E-J
N = cluster# in hex

All subdirectories on disk X are searched and the names of all files having cluster# N are listed.

4.3.12 FORMAT

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

COMMAND LINE SYNTAX:

[X:]FORMAT (cr) (multiple disk formatting-DRIVE B:)

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 the "system" drive. Values are A-J.

Entry of the Command Line Syntax of this program initially prompts the user as follows:

FORMAT <== DRIVE B: ==>, MULTIPLE FORMAT OPTION
TO ESCAPE ENTER CARRIAGE RETURN
INSERT DISKETTE TO BE FORMATTED IN DRIVE B:
AND SPECIFY TYPE (0-7, C OR D)

0 - 8" - 1 sided-single density, 128 Bytes/Sector
1 - 8" - 2 sided-single density, 128 Bytes/Sector
2 - 5" - 1 sided-single density, 128 Bytes/Sector
3 - 5" - 2 sided-single density, 128 Bytes/Sector
4 - 8" - 1 sided-double density, 128 Bytes/Sector
5 - 8" - 2 sided double density, 128 Bytes/Sector
6 - 5" - 1 sided-double density, 128 Bytes/Sector
7 - 5" - 2 sided-double density, 128 Bytes/Sector
C - 8" - 1 sided-double density, 256 Bytes/Sector
D - 8" - 2 sided-double density, 256 Bytes/Sector

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)

NOTE

Only user 0 can be active and Drive B must be dismounted.

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. Entering any other key causes aborting of the program.

If in multiple format mode when the first diskette is formatted, the type options are redisplayed automatically. After inserting a new disk in the B drive only insert the type numbers. This is a multiple format option only.

When an error message of "Drive Not Ready" appears, you may enter an R causing the system to repeat its attempt to access the disk in drive B. Return to Console Processor by entering another key.

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

4.3.13 CDCOPY/WDCOPY

The purpose of XXCOPY Utility Command is to make a copy of platter Y on platter Z.

COMMAND LINE SYNTAX:

```
[X:]XXCOPY Y:=Z: (cr)
```

Where: XX = CD represents Cartridge Module Drive
 = WD represents Winchester Disk
 Y = Output Platter
 Z = Input Platter

CDCOPY/WDCOPY assumes both platters are initialized using CDINIT/WDCOPY.

Once the CDCOPY/WDCOPY is complete, CDINIT/WDINIT is run on the receiving input platter for function verification.

4.3.14 CDINIT/WDINIT

The purpose of CDINIT/WDINIT Utility Command is to allow formatting, verification, labelling and bitmap initialization of hard disk platters.

COMMAND LINE SYNTAX:

```
[X:]XXINIT (cr)
```

Where: CD represents Cartridge Module Drive
 WD represents Winchester Drive

XXINIT interaction is as follows:

CDINIT prompts the user with two menus. The first is a selection area of Hard Disk to operate as follows:

A = all of fixed area (F thru J)
B = all of both fixed and removable (E thru J)
E,..J = all of indicated single platter
X:xxx = track xxx on indicated platter

NOTE

- a. A,B are not allowable if BITMAP initialization is to be selected in function menu
- b. X:xxx - xxx portion is ignored by functions where this is not relevant

An invalid selection invokes a retry or abort program message.

The second menu is to select the function(s) to perform are as follows:

- 1 = format
- 2 = label and create directories
- 3 = verify (BADTRACK.MAP is produced automatically)
- 4 = reassign bad tracks (is not done unless verify is selected)
- 5 = initialize bitmap on a single platter

Option 2 creates all directories on a platter. It asks for a label and the number of additional subdirectories (0-8). The master directory of each platter is directory 0. The label entry in this directory points to all of the other subdirectories. Labels and empty entries (E5's) are written throughout each directory. A subdirectory cannot simply be added to those already existing; instead, the complete directory structure on a platter must be re-initialized from scratch.

NOTE

BADTRACK.MAP is for information purposes only. It is not used by the operating system in any way.

NOTE

During execution of bad track reassignment, results of verification are displayed, and additional bad tracks. These are added to the bad track map.

4.3.15 CDTEST/WDTEST

The purpose of CDTEST/WDTEST Utility Command is to seek and read at random on a hard disk drive, cartridge module disk drive or Winchester Disk Drive.

COMMAND LINE SYNTAX:

[X:]XXTEST

Where:

- XX CD for Cartridge Module Disk Drive
- WD for Winchester Disk Drive
- Example: CDTEST: (cr)
- X Specifies the disk which contains the CDTEST.COM file or WDTEST.COM file.

The user is prompted with:

1. FORMAT, VERIFY and WRITE a data pattern to drive.
2. Perform random read/seek on selected platter.
3. Perform incremental read/seek on selected platter.
4. Perform continuous read/write seeks on selected platter.
5. Perform RAM buffer test on the XCOMP controller.

After making your selection the user is prompted with:

Depress RETURN when ready to STOP test:

The test selected is running at this time and can be stopped by just depressing the RETURN key.

If test 2 is selected then the Cartridge Module Drive stops seeking when an error is found. However, the error found is not displayed. At this time the user may simply depress RETURN and reselect the test or quit at his option.

If an invalid test is selected the user is prompted with:

```
INVALID TEST SELECTION====TRY AGAIN  :
(1)  RANDOM SEEK/READ
(2)  RANDOM SEEK/READ W/STOP ON ERROR
(3)  END TEST ROUTINE
YOUR SELECTION?                       :
```

4.3.16 LINK

LINK IS A Utility Command that concatenates modules together and resolves global symbol references to provide communication between modules. See the Text Editor, Z-80 Global Assembler, Linker Manual for details. This command performs the following functions:

- * Links object input files (secondary filename = OBJ)
- * Reads a LOAD module to produce a memory image file (secondary filename = COM)
- * This command has an option to create a cross reference file (secondary filename = CRS)

COMMAND LINE SYNTAX:

```
A>LINK filename1,filename2,...filenameN/options (cr)
```

OPTIONS:

- C - Produces an output file containing a global cross reference table and a load map
- U - Lists all undefined global symbols.
- A - Specifies where the relocatable module is positioned.

Example: Link the relocatable object modules MAIN.OBJ, SUB1.OBJ, SUB2.OBJ, SUB3.OBJ together starting at 100H producing the LOAD module MAIN.HEX. Also generate a global cross reference table and a load map in the file MAIN.CRS.

```
A>LINK MAIN,SUB1,SUB2,SUB3 /C A=100 (CR)
```


NOTE

With LINK (Version 3.1) Linker options can be entered on the command line following the last file name in the list of files to link. This makes it possible to batch up several links to do without operator interaction. If the slash (/) is omitted, the options are requested as shown below.

OPTIONS? A C (CR)

ENTER STARTING LINK ADDRESS> 100 (CR)

MAIN .OBJ
SUB1 .OBJ
SUB2 .OBJ
SUB3 .OBJ
UNDEFINED SYMBOLS 00
PASS 2

MAIN	.OBJ	REL	BEG ADDR 0100	END ADDR 0125
SUB1	.OBJ	REL	BEG ADDR 0126	END ADDR 01CD
SUB2	.OBJ	REL	BEG ADDR 01CE	END ADDR 01E8
SUB3	.OBJ	REL	BEG ADDR 01E9	END ADDR 0212

A>

4.3.17 LOAD

The purpose of LOAD Utility Command is to load the Hex object file referenced and converts the file to a COM file (binary). The hex object file is left intact.

COMMAND LINE SYNTAX:

[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 the "system" drive. Values are A-J.
file.ref	Specifies the file to load. This must be a file with the secondary name HEX.

4.3.18 MCOPY (VERSION 2.0)

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 SYNTAX

[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 other drives must be connected to the system for C and D values.

/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 formats 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 converts into the multiple copy mode automatically. The user is prompted with:

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

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

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

The destination drive is specified, then the user is 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 COSMOS

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 depresses '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 the COSMOS operating system.

When copying starts it displays the progress on the CRT as follows:

```
o          1          2          3          4          5          6
012345678901234567890123456789012345678901234567890123
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
EEEEEEEEEEEEEEEE
```

DISKETTE COPIED AND VERIFIED

At this time 'COPY FROM DRIVE (A, B, C or D) or Q to QUIT' is 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 depressing any key. This returns the prompt:

OPERATOR ABORT

and allows the same selections to enter or a new task to start. 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 aborts. 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 the user from copying an empty diskette to a good diskette in the event that drives are specified improperly. However, the following message prints:

DISKETTE COPIED AND VERIFIED

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

CAUTION

The format option only formats the diskette in the drive specified as destination drive in menu prompt, so be sure the diskette in the drive shown in the prompt is the one you want formatted, otherwise, a good diskette is formatted if it is the

drive specified as the destination (and the 'F' option is selected).

ERROR MESSAGES:

Wrong Disk Type Drive A Disk being copied to (destination
B must be reformatted.
C
D

INVALID DRIVE SELECTION - Disk drive selection was specified incorrectly

OPERATOR ABORT

DATA VERIFICATION ERROR DRIVE X:
Y:

DISK READ ERROR DRIVE X:
Y:

4.3.19 MOVE

The MOVE utility allows moving a copy of a local file to another user's directory.

COMMAND LINE SYNTAX:

```
[X:]MOVE [Y:] file.ref N (cr)
```

Where:

X Specifies the disk drive location of MOVE.COM. This parameter is required only if the COM file is not located on the currently selected drive or the "system" drive.

Y Specifies the desired disk drive location. This defaults to the currently selected drive.

file.ref is an unambiguous file name

N is a user number (0-7)

If the specified file exists in the current users directory and is not global, a copy of the file is made in the specified user's directory on the current drive. If the file exists in the specified user's directory no copy is done.

4.3.20 SPOOL

COMMAND LINE SYNTAX:

```
[X:] SPOOL [Y] file.ref (cr)
```

```
[X:] SPOOL (*) (cr)
```

Where

- X Specifies the disk drive location of the SPOOL.COM file. This parameter is required only if the COM file is not located on the currently selected drive or the "system" drive. Values are A-J.
- Y: Specifies the disk location of the file. The default is the "system" drive.

The SPOOL command is used to print a file as a background operation. After issuing the command, the specified file is printed. When the printing is complete, a form feed is issued and the printer is released.

The spooling operation may be terminated by entering "SPOOL * (cr)" with no file name specified.

4.3.21 WRTCOS

Provides a means of moving COSMOS from one disk to another. Note that when copying COSMOS to a floppy diskette, the 256 byte formats (C or D) must be used.

COMMAND LINE SYNTAX:

```
[X:] WRTCOS {Y: } {Z: }  
{file.ref1}={file.ref2} (cr)
```

Where:

- X Specifies the disk drive location of the WRTCOS.COM file. This parameter is required only if the COM file is not located on the currently selected drive or the "system" drive. Values are A - J.
- Y Specifies the disk drive to write to. NOTE: If only the drive designation is given COSMOS is written to the system area.

file.ref1	Specifies the file to write to. .SYS must be included in the secondary file if no disk drive is specified before the file name.
	NOTE: When using a file.ref, precede it with the disk drive designation of the file to find.
Z	Specifies the source disk drive.
	NOTE: If only the drive designation is given the system is read from the system area.
file.ref2	Specifies the file to read from and must include the secondary file name of SYS.

NOTE

The memory resident COSMOS is changed only upon booting the system.

Examples: WRTCOS B:=A: (cr)

copies COSMOS from the disk system area on drive A to the disk system area on drive B.

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

copies the COSMOS from OLD.SYS file on the disk in drive B to MOD.SYS file on the disk in drive A.

4.3.22 XDIR

Lists the disk directory in alphabetical sequence. All files in the users directory and all global files are listed.

COMMAND LINE SYNTAX:

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

Where:

Y: Specifies the disk drive file directory to display. If a drive is not specified, the directory of the currently selected drive is displayed.

file.ref Causes the specified file(s) size(s) to display. This file reference may include the replacement characters ? and/or *.

e.g. Entering: XDIR FILE??.* (with file located on a single sided double/density disk) causes the display of the following headings 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 causes the following to display on the console.

```
FILE41.FAZ 5K
```

1 Entry listed, 5K disk space used.

Entering a CTRL S while the directory is being displayed causes 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 listed in table 4-4.

Table 4-4. File Allocations

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

4.3.23 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 SYNTAX
 REPETITIVE EXECUTION:
`[X:]XFER (cr)`
 ONE TIME EXECUTION:
`[X:]XFER[/p1 /p2...{file.ref1}=file.ref2 [,file.ref3...] (cr)`

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 the "system" drive. Values are A-J.
- /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 as the same.
 - F designates to drop illegal ASCII characters.
 - I Designates that the ASCII end of file should be ignored. (CTRL-Z)
 - S Strips all rub-outs and nulls 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 verify after transfer.
Z	Designates that the size statistics are not to print upon completion of the XFER.
file.ref1	Specifies the destination file reference. This may be any file name or non-disk device such as LST: or CON:.
file.ref2,3	Specifies the source file references. If more than one file ref is given the files concatenate (link). If ASCII files are linked, the A parameter must be used to remove the end of file marks.

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 paragraph 4.3.1).

NOTE

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

Ex.: 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) concatenate and transfer to FILE31.FAX on disk drive A. Any illegal characters drop in any of the transferred files (/F). After the transfer the size statistics are not displayed (Z).

NOTE

When concatenating 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 concatenate.

4.3.24 XSTAT

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

COMMAND LINE SYNTAX:

```
[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 the "system" drive. Values are A-J.
- 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.

```
SD SYSTEMS DISK STATUS V3.3
```

```
User Memory Size      :      XX kbytes
```

```
Total Disk Space     :      XX kbytes
```

```
Disk Space Used       :      XX kbytes
```

```
Disk Space Remaining:      XX kbytes
```

```
Subdir 0 - Directory Entries Used :    XX
```

```
..  
Disk Status Completed
```

If an erroneous status is detected, one of the following is displayed:

```
Checksum error in bitmap block# XX
```

```
Linked or bad cluster# XX : File YYY
```

Note that XDIR, DIR give information for a single user, whereas XSTAT is for all users. Also the disk utilization totals displayed by XDIR, DIR do not include the label, directory, and bitmap allocation, whereas XSTAT does include these. XDIR and DIR only include the file space used by global files and the local user. XSTAT is an accurate indication of total disk space utilization.

4.3.25 ZASM

Used to assemble Z-80 source programs (Mostek and Zilog mnemonics). 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 V

PROGRAMMERS GUIDE

5.1 INTRODUCTION

COSMOS is a multi-user 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. All standard* programs load and run at 100H. (see detailed Low Memory Utilization Map in table 5-1). Once loaded, a program may use all memory up to the bottom of the user variable area. This address is contained in memory locations 6,7 (LSB first). All DOS requests are made with a CALL to location 5.

2. When a program is started from CONPROC, the command line is parsed starting with the first character after the program name and up to two (2) File Control Blocks (FCBs) are created (FCB-1 and FCB-2). FCB-1 starts at 5CH and receives the first file-reference, if any, in the command line.

FCB-2 starts at 6CH and will receive the second file-reference, if any, in the command line. 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 FCBs are given in paragraph 5.4.2.

3. COSMOS uses a byte (address 0003) for specifying which physical device is assigned to each non-disk logical device. This byte is utilized by the internal SET command (table 5-2), and DOS calls 7 and 8 (paragraph 5.3.1). Only the printer is presently implemented.

NOTE

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

Table 5-1. Detailed Low Memory Utilization Map

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	COSMOS Re-Entry (warm start)

Table 5-2. I/O Bytes Description

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

5.2 SYSTEM CALLS

A system call is made to the operating system to initiate a function, usually involving I/O. These functions are divided into the following four (4) sections:

- A. I/O functions for all non-disk devices (including the console, printer, etc.)
- B. I/O functions for disk files (including creation, search, renaming, deleting, opening and closing.)
- C. Special purpose functions (including integer multiplication, division, system abort and program links).
- D. Special COSMOS calls.

A system call is accomplished by loading the C register with the function number and loading the entry parameters (if any) into the specified registers. When a CALL 5 instruction is executed COSMOS performs the requested function. When finished, the user program returns by entering a RET instruction. The user's registers are returned unchanged, except those specified in table 5-3.

5.3 COSMOS NON-DISK DEVICE SYSTEM CALLS

The system calls in this section are for non-disk device I/O. The number preceding each COSMOS function is the number 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 5-3.

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). COSMOS returns control to the user program when a character is read and echoed back to the CRT. Only the value in the C register is required as an entry parameter.

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. COSMOS waits until the console is ready to receive the character and then displays it. No parameters are returned by the call.

If a CTRL-P (^P) is entered while COSMOS 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-\ (^\) also sends subsequent characters to the console and the printer. CTRL-] (^]) cancels the effect of either the CTRL-\ or the CTRL-P and sends characters only to the console. CTRL-\ and CTRL-] may be sent programatically to the console or placed in a file so that when the file is displayed on the console, the file can start and stop the printer at the required 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 (09H).

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) are set up initially with the console as a reader for this system call.

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 punch is placed in the E register before the call. COSMOS 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 print is placed in the E register before the call, and COSMOS 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.

6-DIRECT CONSOLE I/O

Register E either contains FF, indicating a console input request, or contains an ASCII character to output. If E = FF, then, then sys call 6 returns A := 0 if no character is ready, otherwise A := next console input character. If E was not FF, then the character is output to the console.

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 in table 5-2. 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 in table 5-2.

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 COSMOS. If the printer toggle is on, the line is also 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 COSMOS. The first byte of the buffer must contain the maximum buffer length. Upon return from this call the second byte of the buffer contains the actual length entered. The input line is stored beginning at the third byte. If the buffer is not full, the byte at the end of the line is zero filled.

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

CTRL-C (^C)	Warm boot back to COSMOS.
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 the console screen.
CTRL-P (^P)	Toggles printer list software switch. Causes console output to also print. Alternate on/off action.
CTRL-R (^R)	Repeats what is 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 (0BH)- TEST CONSOLE READY

The console is tested for a typed character. If a character is typed, FFH is returned in the A register. If no character is typed, 0 is returned in the A register.

128 (80H)-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.

<u>Mnemonic</u>	<u>Function</u>	<u>Parameters for Register D</u>	<u>Parameters for Register E</u>
Address	Address cursor on screen	1-80	1-24
Clear	Clear CRT screen	0	0

When addressing the cursor, the D register should contain the column address (1 through 80) 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.

For reference, the cursor location (1,1) is in the upper left-hand corner and the cursor location (80,24) is in the lower right-hand corner of the screen.

NOTE
Hex values must be entered.

5.4 COSMOS DISK SYSTEM CALLS

COSMOS 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:

Table 5-3. File Control Blocks Bytes Description

DESCRIPTION	BYTE #	
FCBDK Disk descriptor	0	(0=current disk, 1-drive A, 2-B, 3-C, 4-D, 5-E, 6-F, 7-G, 8-H, 9-I, 10-J)
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	12,13	(initially=0; is incremented by 1 in every new entry of 16K bytes)
FCBDE Directory Entry	14	(big directory disks initially = 0)
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 0 through 127)
FCBRR Random record	33-35	(required only for sys calls 33-37 for random I/O. Record number with least significant byte on left.)

Several system calls assign a directory code in the A register as a returned parameter. This is a value in the range 0-3 which is useful only for system calls 17,18.

All system calls dealing with the directory structure of COSMOS, i.e., those having an FCB as a parameter, are implicitly referring to the currently active subdirectory of the disk indicated in byte 0 of the FCB.

12 (0CH)-LIFT HEAD and specify CP/M version compatibility

Register B is disk# (0=current,1=A,etc.). If B is invalid, the current disk is used. HL is returned with 02FH, indicating CP/M 2.X compatibility.

13 (0DH)-RESET COSMOS AND SELECT SYSTEM DISK

COSMOS is initialized, all disks are logged-off, and the system disk as defined during system generation is selected as the current drive. The other disks are logged-on as needed.

14 (0EH)-SELECT DISK DRIVE

The disk drive number in the E register is selected as the current disk. The drive numbers in the E register are listed in table 5-4.

Table 5-4. E Register Drive Numbers

<u>DRIVE NAME</u>	<u>DRIVE NUMBER</u>
A	0
B	1
C	2
D	3
E	4
F	5
G	6
H	7
I	8
J	9

15 (0FH)-OPEN DISK FILE

The file extent 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 extent is not found, or a directory code if the file extent is found. Block numbers start at 0 and there are four directory entries per block on small directory disks and two entries per block on big directory disks. The DE register pair returns pointing to the directory entry in memory.

16 (10H)-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 code if the file is found on the drive. The file described by the FCB must be previously opened or created. A file must be closed using this function or the entire last entry (or extent) is unreadable because no cluster information is present 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 directory code 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 picks up the directory entry whether it has been erased or not. Files are erased by placing a 0E5H 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 the 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.

20 (14H)-READ NEXT RECORD

The DE register pair points to a successfully opened FCB. The next block 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:

- 0 - 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 write the next record. One of the following codes is returned in the A register:

- 0 - write completed
- 1 - entry error (attempted to close an unopened entry)
- 2 - out of disk space
- 1 - (or FFH) out of directory space (see Appendix 1)

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 directory code of the directory entry (see 0FH - 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 defines the FCB to rename. 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 matches with any character. The A register returns -1 if error else a directory code.

24 (18H)-GET MOUNTED DISK VECTOR

The HL register pair is returned specifying the disks that are mounted. Each bit represents one disk drive; if the bit is a one, it is mounted. The least significant bit is the A drive, next most significant (Bit 1) is drive B, etc. For compatibility with SD-OS the A register also contains the contents of register L.

25 (19H)-GET CURRENT DISK

The number of the current disk drive is returned in the A register.

0 = drive A, 1 = drive B, 2 = drive C, 3 = drive D, 4 = drive E,
5 = drive F, 6 = drive G, 7 = drive H, 8 = drive I, 9 = drive J.

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 80H.

27 (1BH)-GET DISK CLUSTER ALLOCATION MAP

The BC register pair returns pointing to a bit map of the disk allocated clusters.

28 (1CH)-WRITE PROTECT CURRENT DISK

The current disk is software write-protected for this user. This protection is temporary, lasting only until the termination of the current program's execution.

29 (1DH)-GET READ/ONLY VECTOR

The HL register pair is returned specifying the disks that are software write-protected by this user. Each bit represents one disk drive; if the bit is a one, it is write-protected. The least significant bit is the A drive, next most significant (Bit 1) is drive B, etc. For compatibility with CP/M, the A register also contains the contents of register L.

Note that each user has his own R/O vector and cannot influence the R/O vector of another user.

30 (1EH)-SET FILE ATTRIBUTES

The only CP/M-style attribute supported by COSMOS is read-only, which is interpreted as COSMOS permanent attribute. The DE register pair points to an FCB, whose high-order bit in FCBEX indicates:

1 = make this file permanent

0 = make this file non-permanent

The current user/subdirectory is searched and the found entry is changed as indicated. A directory code is returned in A register, or -1 if no match is found.

31 (1FH)-GET DISK PARM ADDR

NOT SUPPORTED

32 (20H)-SET/GET USER DIRECTORY NUMBER

This call is used to identify or change the current user directory number from an application program. The Console Processor USER intrinsic does the same thing, but only allows it at the system console. The system call allows any terminal to perform this function.

Register E either contains FF, indicating a request to find the user dir#, or contains a value which (modulo 16) is an assignment request. If E=FF, then sys call 32 returns A:=current user#, otherwise the user dir# is changed and nothing is returned.

33 (21H)-READ RANDOM RECORD

The DE register pair points to an extended FCB with FCBRR filled in. Some extent, probably extent 0, must be open (FCBEX and FCBNR reflect last record read or written). COSMOS reads the record indicated by FCBRR, changing extents if necessary, and updates FCBEX, FCBNR based on FCBRR. The A register is assigned a status code as follows:

- 0 = O.K.
- 1 = reading unwritten data
- 3 = cannot close extent
- 4 = seek to unwritten extent
- 6 = seek past end of disk

There is no automatic incrementing by 1 as on sequential read.

34 (22H)-WRITE RANDOM RECORD

The DE register pair points to an extended FCB with FCBRR filled in. Some extent, probably extent 0, must be open (FCBEX and FCBNR reflect last record read or written). COSMOS writes the record indicated by FCBRR, changing extents if necessary, and updates FCBEX, FCBNR based on FCBRR. The A register is assigned a status code, same as for system call 33, or:

- 5 = can't create extent because dir full

There is no automatic incrementing by 1 as on sequential write.

35 (23H)-COMPUTE FILE SIZE

The DE register pair points to an extended FCB. COSMOS does a directory search and assigns FCBRR a value just past end of file. If the file does not exist, FCBRR:=0.

Data can be appended to an existing file by executing the following sequence of system calls:

```
open extent 0
compute file size
write random
add 1 to FCBRR
write random
add 1 to FCBRR
...
```

36 (24H)-SET RANDOM RECORD

The DE register pair points to an extended FCB. COSMOS examines FCBEX and FCBNR and computes the corresponding FCBRR value.

131 (83H)-READ LOGICAL BLOCK

This call reads a logical block from the disk without any attention to the files it may contain (no FCB is specified). A block is defined as one record of 128 bytes. When this function is called, the ADE registers should contain the block number and the B register should contain the disk number (0 for current drive, 1-10 for A-J). The high bit of the B register contains a 1 for an interleaved and a 0 for a non-interleaved read. Interleaved means the block is read and found in the order COSMOS stores it. Non-interleaved means the block is read and 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:

- 0 - OK
- 1 - I/O error
- 2 - illegal interrupt
- 3 - illegal block
- 4 - software write-protect

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 for the Read Logical Block (131) call.

133 (85H)-SPOOLER CONTROL

The spooler allows background printing of a disk file. This call is used to initialize, start, stop, and get the status of the spooler. The DE register pair is used to define the operation:

DE = FCB ADDRESS	TO START FILE SPOOLING
DE = 0	TO DISABLE SPOOLING
DE = 1	TO READ SPOOLER STATUS

Spooler status is returned in the A register and if 0, the spooler is inactive (either because previous file printing is complete or the spooler was disabled (DE = 0) by program). If status is FF, the spooler is actively printing a file.

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 built. The line entered is structured:

x:primary filename.secondary filename

where x is the drive, 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 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.

139 (8BH)-HOME DRIVE HEAD

The disk drive specified in the B register (0 for current drive and 1-10 for drives A-J) is sent a command to HOME the head (the disk drive head returns to track 0)

140 (8CH)-UNLOCK DISKETTE

This call unlocks the disk drive door whose number is given in the E register (0 for current drive and 1-4 for drives A-D, respectively).

148 (94H)-SET COSMOS FILE ATTRIBUTES

The DE register pair points to an FCB. Register B contains the control information for changing attributes as follows, with a value of 1 in a bit position indicating a request for the specified action:

bit 7 = Permanent
bit 6 = Global
bit 0 = +

The effects are as indicated under the Conproc intrinsic ATRIB. Register A is returned with -1 if no match, -2 if a conflict occurs, otherwise a directory code.

5.5 SPECIAL PURPOSE SYSTEM CALLS

Special purpose system calls are explained in this section.

0 (00H) PROGRAM ABORT

This call aborts the current program and returns control to COSMOS. This call has the same effect as jumping to location 0.

129 (81H)-GET USER REGISTER POINTERS

The BC register pair returns pointing to UCB (USER CONTROL BLOCK). HL contains address of active user table. A contains current active user and E contains directory type.

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

When CTRL-C (^C) is typed, the system normally aborts and returns control to COSMOS. 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 0 at any time still causes a return to COSMOS, 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 FCBS for the new program must be set up prior to this call if that program expects to use them. The DE register pair should contain the new program FCB address (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 80 bytes (from 100H to 17FH) are destroyed because this is used in reading the directory. If the program is found, execution begins at 100H 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).

5.6 SPECIAL COSMOS CALLS

The following four calls provide a means of setting and reading the clock/calendar. The clock is 24 hour.

143 (8FH)-SET DATE*

Set date from registers. Enter with B = day, D = month, E = year.

144 (90H)-READ DATE*

Read date into registers. Returns with A = day, B = month, C = year.

145 (91H)-SET TIME*

Set time from registers. Enter with B = seconds, D = minutes, E = hours.

146 (92H)-READ TIME*

Read time into registers. Returns with A = seconds, B =minutes, C = year.

192 (C0H)-START USER

This call is used by the console processor to start-up a user. Application programs should not issue this request.

193 (C1H)-STOP USER

This call is used by the console processor to stop a user's job. Application programs should not issue this request.

194 (C2H)-MOUNT DISK

This call is used by console processor to mount disks. Upon entry, the E-register contains the disk number (A=0, B= 1, etc.) to mount. If the system determines that a previous disk is not dismounted, the MOUNT request is not executed and the A-register returns with -1 (FF) in it. If a disk error occurs during the mount a -2 (FE) is returned. If the mount is successful A=0.

195 (C3H)-DISMOUNT DISK

This call is used by console processor to dismount disks. The disk to be dismounted (A=0, B=1, etc.) is loaded in the E register. Upon return, if A=0, then DISMOUNT is performed. If A does not = 0, then the DISMOUNT is not done because the disk is being used. (A=log-on status user)

196 (C4H)-GET DISK USER STATUS

This call is used to determine which users are currently logged on a specified disk. Upon entry, E - register contains the disk number (A=0, etc.). Upon return, the A-register contains the log-in status of each user on the specified disk. Bit 0 is user 0 status, bit 1, user 1 status, etc. A 1 indicates that the user is logged on the specified drive.

197 (C5H)-DISPATCH MESSAGE

This call is used to send a message to other users. Upon entry the A register should contain the destination user number. If A=15, the message is sent to all active users. The DE pair must point to the message text with the first byte defined as the message length. Maximum message length is 80 characters. The first 6 bytes of the message are overwritten and should be left blank.

198 (C6H)-RECORD LOCKING

ENTRY PARMS: DE = ADDR OF LOCK PARMS
 C = 0C6H
EXIT: A = ERROR CODE

Layout of the 8-byte parameter block pointed to by DE is as follows:

<u>MNEMONIC</u>	<u>MEANING</u>	<u>LENGTH</u>	<u>VALUES</u>
LKOP	Lock opcode	1 byte	1,2,3
LKFIL	File#	1 byte	<> 0
LKDSK	Disk#	1 byte	0=current ,1=A, etc. (as in FCB's)
LKRECZ	Record size	2 bytes	>=16
LKRECN	Record number	3 bytes	

*All Date and Time parameters are in packed BCD format. This means that two BCD numbers are contained within each eight bit register.

Not all parameters are used on all calls, but storage for same must still be present in the parameter block. COSMOS does not change the values of any parameters. LKRECZ and LKRECN are stored left-to-right, with most significant bit on left.

The error code in register A is 0 if operation is successful; -1 if invalid code, otherwise as indicated in table 5-5.

Table 5-5. Register A Error Codes

<u>OPCODE</u>	<u>PARMS REQUIRED</u>	<u>ERROR CODES/MEANING</u>
1 = init lock for a file	LKFIL LKDSK LKRECZ	-2 Table full -3 Invalid parm: File# = 0 Disk# invalid, Disk not mounted, or Record size < 16 bytes -4 Already present
2 = set a lock or zero it	LKFIL LKRECN	>0 implies already locked by user# = Errcode -1 -5 File not in table -6 Not active on indicated file
3 = flush file	LKFIL LKRECN	-3 Invalid parm: Either file# or rd# must be zero -5 File not in table -6 Not active on indicated file

The lock table stores information on up to 8 files. For each, the current record # (or zero if none) locked by each user is indicated. Only 1 record # can be locked at a time by a given user on a given file. Note that these record #'s start at 1. COSMOS does not wait for an error condition to clear but always return immediately to the user caller. The rules governing locking a file are as follows:

1. Each user must INIT prior to LOCKing; the first one in puts the file in the table.
2. If LOCK fails, current record # is zeroed anyway; if it succeeds, the current locked record # is updated and the prior record# is released.
3. LOCK with record # 0 is a request to zero the record #, releasing any locked record without removing the file from the lock table.
4. LOCK with file #0 is a request to zero all record #'s for this user.

5. FLUSH is the reverse of INIT; the last one out removes the file from the lock table.
6. FLUSH with file# 0 is a request to flush all files for this user; i.e. FLUSH ALL.
7. FLUSH ALL is done automatically by COSMOS at the end of a user program, i.e.


```

      JP 0 or LD C,0
      CALL 5
      
```

For the convenience of BASIC programmers, a parameter block and subroutine to call from BASIC is available as illustrated in paragraph 3.3 of this manual. The following is a brief explanation thereof:

Location 6-7 has the address of the "user area". At offset 9 in user area are the locking parameters laid out as follows:

X:	JP	LOCKER	3 bytes
LKPRM:	DEFS	8	8-byte parm block
LKRET:	DEFS	1	1-byte return code

The BASIC example does a PEEK to establish the location of locking parms, POKES them in, calls X, and PEEKS for results.

Files to access in this shared mode should be allocated (or extended) in an off-line mode first and globalized. Programs accessing them must be designed to detect empty records and/or last-valid-record via other than the normal end-of-file. This may involve locking an index or an EOF-Counter record stored in the same or different file.

199 (C7H)-ATTACH SUBDIRECTORY

Register B indicates a disk drive, 0=A, etc., register B indicates a subdirectory number. The indicated subdirectory is made active for this user. The A register is returned -1 if an error occurs, which means:

```

disk is not mounted
disk number is invalid
subdirectory does not exist
disk number is not a Hard Disk
  
```

200 (C8H)-IDENTIFY SUBDIRECTORY

Register E indicates a disk drive, 0=A, etc. The A register returns -1 if an error occurs (same as for ATTACH except floppy is valid, A: = 0), otherwise A is returned as the number of the currently active subdirectory on the specified disk for this user.

5.7 NON-CP/M SYSTEM CALLS

The following is an example of required sequence of usage on non CP/M System Calls.

SET DISK BUFFER	26 (1AH)
GET USER STATUS	196 (C4H)
LIFT HEAD/SPECIFY VERSION	12 (02FH)
DISMOUNT DISK	195 (C3H)
HOME DRIVE	139 (8BH)
DISK CLUSTER ALLOCATION MAP	27 (1BH)
READ/WRITE LOGICAL BLOCK	131 (83H)/132 (84H)

Table 5-6 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).

Table 5-6. COSMOS Call Summary

NUMBER	FUNCTION	REGISTERS AND ENTRY PARAMETERS	REGISTERS AND RETURN PARAMETERS
0	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	DIRECT CONSOLE I/O	E=0FFH requests input or =anything else is char to output	A=0 implies console not ready A<> 0 is character read
7	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 (0AH)	INPUT BUFFERED LINE	DE-buffer address	none
11 (0BH)	TEST CONSOLE READY	none	A= -1 (FFH) if ready A=0 if not ready
(02FH)	LIFT HEAD/ SPECIFY VERSION	B=disk drive #	HL=02FH indicating CP/M 2.X compatability
13 (0DH)	RESET COSMOS AND SELECT SYSTEM DRIVE	none	none
14 (0EH)	SELECT CURRENT DISK	E-disk drive no.	none
15 (0FH)	OPEN DISK FILE	DE-FCB address	A=directory code A=-1 (FFH) if not found
16 (10H)	CLOSE DISK FILE	DE-FCB address	A=directory code A= -1 (FFH) if not found
17 (11H)	SEARCH DIRECTORY FOR FILENAME	DE-FCB address	A=directory code A= -1 (FFH) if not found
18 (12H)	FIND NEXT ENTRY IN DIRECTORY	DE-FCB address	A=directory code A= -1 (FFH)if not found
19 (13H)	DELETE FILE	DE-FCB address	A= -1 implies not found = anything else, O.K.
20 (14H)	READ NEXT RECORD	DE-FCB address	A=0 if OK A=1 if end of file A=2 if tried to read unwritten records

Table 5-6. COSMOS Call Summary (Continued)

NUMBER	FUNCTION	REGISTERS AND ENTRY PARAMETERS	REGISTERS AND RETURN PARAMETERS
21 (15H)	WRITE NEXT RECORD	DE-FCB address	A=0 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 code A= -1 (FFH) if out of directory space
23 (17H)	RENAME FILE	DE-FCB address	A= -1 implies not found or already exists = anything else, O.K.
24 (18H)	GET DISK MOUNTED VECTOR	none	HL=those disks currently mounted A= L register A=disk drive number
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 HL=addr of device tables A=sectors/cluster
28 (1CH)	WRITE PROTECT CURRENT DISK	none	none
29 (1DH)	GET READ/ONLY VECTOR	none	HL=R/O vector value
30 (1EH)	SET FILE ATTRIBUTES	DE=FCB address	A= -1 if error (no such file)
31 (1FH)	GET DISK PARM ADDR	NOT SUPPORTED	
32 (20H)	SET/GET USER DIR NUMBER	E=0FFH indicates a request to find out user dir# E=anything else is a request to assign user dir# modulo 16	A=current user# if E=0FFH; none otherwise
33 (21H)	READ RANDOM	DE=Extended FCB addr	FCBEX,FCBNR in FCB are updated; A=status code as follows: 0=o.k. 1=reading unwritten data 3=cannot close extent 4=seek to unwritten extent 6=seek past end of disk
34 (22H)	WRITE RANDOM	DE=Extended FCB addr	FCBEX,FCBNR in FCB are updated; A=status code as for sys call 33 with also 5=can't create extent because dir full

Table 5-6. COSMOS Call Summary (Continued)

NUMBER	FUNCTION	REGISTERS AND ENTRY PARAMETERS	REGISTERS AND RETURN PARAMETERS
35 (23H)	COMPUTE FILE SZ	DE=Extended FCB addr	FCBRR assigned a value
36	SET RANDOM RECORD	DE=Extended FCB addr	FCBRR assigned a value
128 (80H)	READ CONSOLE (with no echo)	none	A=character
129 (81H)	GET USER REGISTER POINTER	none	BC=address of user registers HL=address of active user table A=current active user number E=directory type
130 (82H)	SET USER CTRL-C EXIT (ABORT)	DE=address of C handler (0 to reset; -1 to disable)	none
131 (83H)	READ LOGICAL BLOCK	ADE=block number B-drive number B top bit -1 if interleaved	A=0 if OK A=1 if I/O error A=2 if illegal request A=3 if illegal block
132 (84H)	WRITE LOGICAL BLOCK	ADE=block number B-drive number B top bit -1 if interleaved	A=0 if OK A=1 if I/O error A=2 if illegal request A=3 if illegal block A=4 if software write protect
133 (85H)	SPOOLER CONTROL	DE=FCB	See Details
134 (86H)	FORMAT NAME TO FILE CONTROL BLOCK	HL=address of string	HL=address of terminator
136 (88H)	LINK TO PROGRAM	DE=FCB address DE=FCB address	DE=FCB address A=-1 (FFH) if error; else execute at 100H
137 (89H)	MULTIPLY INTEGERS	DE-factor 1 HL-factor 2	DE=product
138 (8AH)	DIVIDE INTEGERS	DE-divisor HL-divident	BC, HL=quotient DE=remainder
139 (8BH)	HOME DRIVE	B-drive number	none
140 (8CH)	UNLOCK DISKETTE	E-drive number	none
141 (8DH)	GET VERSION OF OPERATING SYSTEM	none	B=version-number C=version-number A=OSTYPE E=SYSTEM DISK #
142 (8EH)	SET SPECIAL CRT FUNCTION	D-column address/ special function E-row address/0	none
143 (8FH)	SET DATE*	B = DAY (DD) D = MONTH (MM) E = YEAR (YY)	none
144 (90H)	READ DATE*	none	A = DAY (DD) B = MONTH (MM) C = YEAR (YY)

Table 5-6. COSMOS Call Summary (Continued)

NUMBER	FUNCTION	REGISTERS AND ENTRY PARAMETERS	REGISTERS AND RETURN PARAMETERS
145 (91H)	SET TIME*	B = SECOND (SS) D = MINUTE (MM) E = HOUR (HH)	none
146 (92H)	READ TIME*	none	A = SEC (SS) B = MIN (MM) C = HOUR (HH)
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	A=-1 implies not found =-2 implies conflict =anything else, O.K. DE-FCB address
149 (95H)	READ DISK LABEL	none	
192 (C0H)	START USER		
193 (C1H)	STOP USER		
194 (C2H)	MOUNT DISK	E = disk number	A = 0 if ok A = -1 if disk already mounted A = -2 if bad disk A = -3 bad cluster
195 (C3H)	DISMOUNT DISK	E = disk number	A = 0 if ok A < > 0 cannot dismount A = user's logged on disk
196 (C4H)	GET USER STATUS	E = disk number	A = user's logged on specified disk. Bit 0 for user 0, bit 1 for user 1, up to bit 7
197 (C5H)	DISPATCH MESSAGE	A = 0-7 destination user A = 15, send to all users DE = address of message with first byte = number of bytes in message	none
198 (C6H)	RECORD LOCKING	DE=addr of lock parms	A=status code
199 (C7H)	ATTACH	E=disk number	A=status code
200 (C8H)	IDENTIFY	B=disk number	A=subdir number

*All parameters for Time and Date are in packed BCD format.

SECTION VI

INSTALLATION AND CONFIGURATION

6.1 INTRODUCTION

The SD SYSTEMS microcomputer hardware must be configured for operation with COSMOS. After that, a software configuration procedure must be executed to specify all options for an installation.

6.2 HARDWARE CONFIGURATION

COSMOS can be run with 32K or 48K user memory partitions in a two user or (up to) five user hardware configuration. There are also a number of configuration jumpers to install on the basic four computer boards.

6.2.1 MEMORY SYSTEM CONFIGURATION

COSMOS supports two basic memory configurations: 32K user partition and 48K user partition. Each configuration supports up to five (5) users. Figure 6-1 shows the 32K memory configuration using three (3) Expandoram II boards. Figure 6-2 shows the 48K memory configuration using four (4) Expandoram II boards.

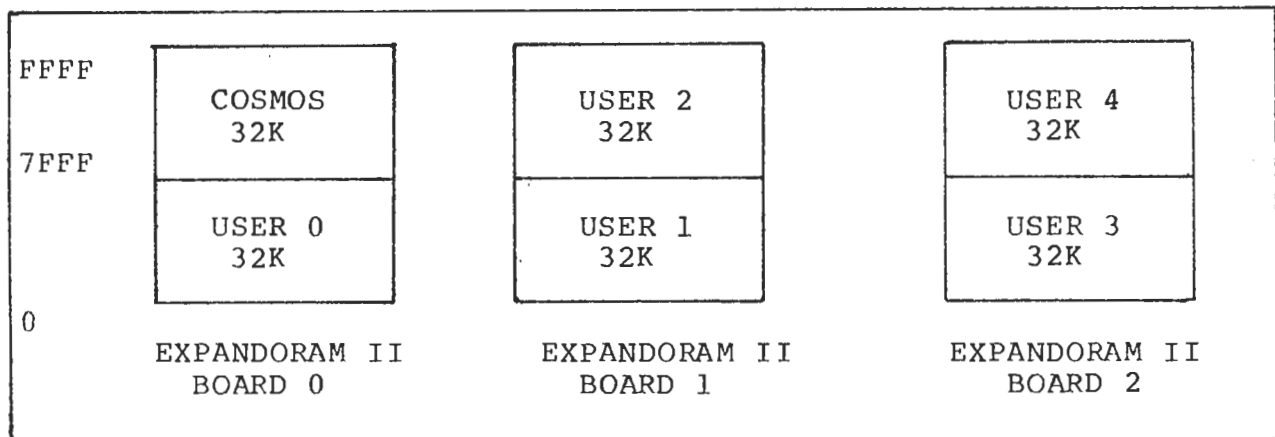


Figure 6-1. 192K Memory System For Five 32K Users

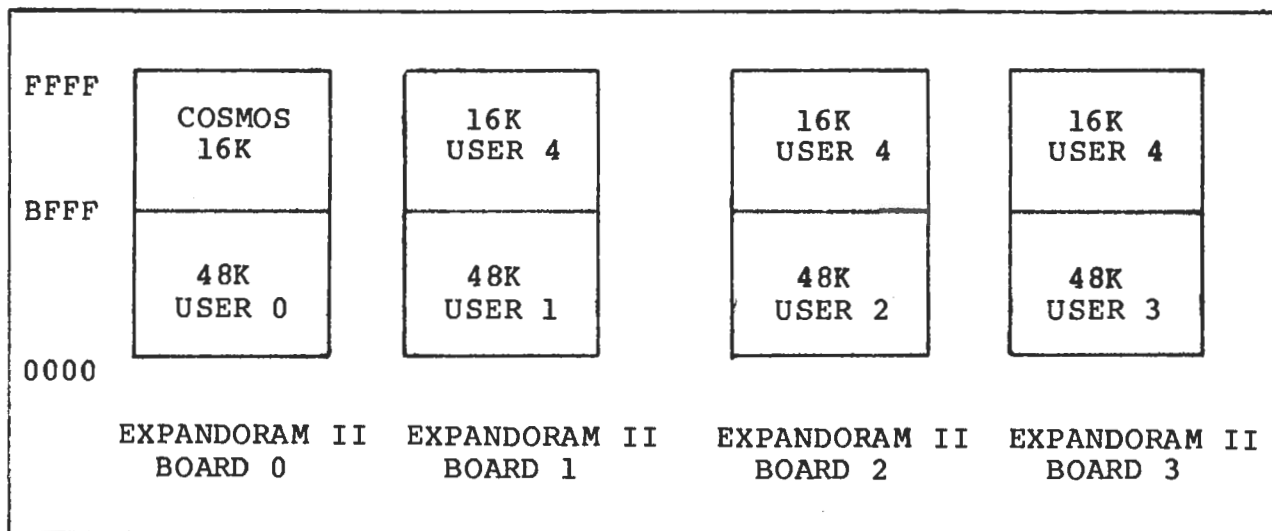


Figure 6-2. 256K Memory System For Five 48K Users

Each user's memory page in a 32K partition system contains logical addresses 0000 to 7FFFH and each page in a 48K partition system contains logical addresses 0000 to BFFFH. In both systems the top 1280 (500H) bytes are used by COSMOS for user variables and buffers.

The top of available memory is always communicated to system and application software via the DOS entry at location 0005 as in any CP/M system.

NOTE

When installing additional Expandoram II boards in the card cage no specific board sequence is required.

6.2.2 TWO AND FIVE USER CONFIGURATIONS

COSMOS can be configured to run with two users with no additional I/O ports added to the basic SD-200 microcomputer. If more than two users are required the MPC-4 board must be installed.

6.2.2.1 Two User Configuration

In a minimum two user configuration, the existing RS-232 port is used to interface with the one external terminal. The parallel interface is used to drive a printer (see figure 6-3).

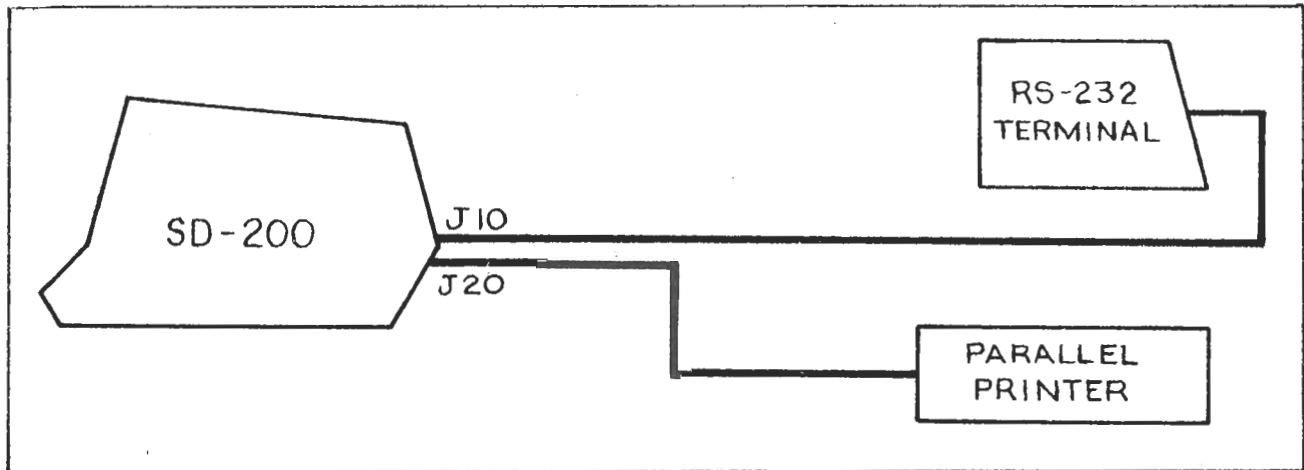


Figure 6-3. Parallel Interface

The only additional hardware required is an ExpandoRAM II for the second user.

6.2.2.2 Five User Configuration

If more than two users are required, an MPC-4 Multi-Port Communicator is needed. The MPC-4 board should be installed close to the rear of the card cage to allow the four mass terminated cables and six-connector panel to be mounted properly. This six-connector panel replaces the two-connector panel which is installed in single user microcomputers.

The top two connectors (J10 and J20) must be mounted with the RS-232 (from J2 on the SBC-100) cable on top, then the parallel interface cable (from J3 on the SBC-200). The CRT brightness control must be moved to the new panel. The lower four connectors are connected to the four MPC-4 RS-232 ports (see figure 6-4).

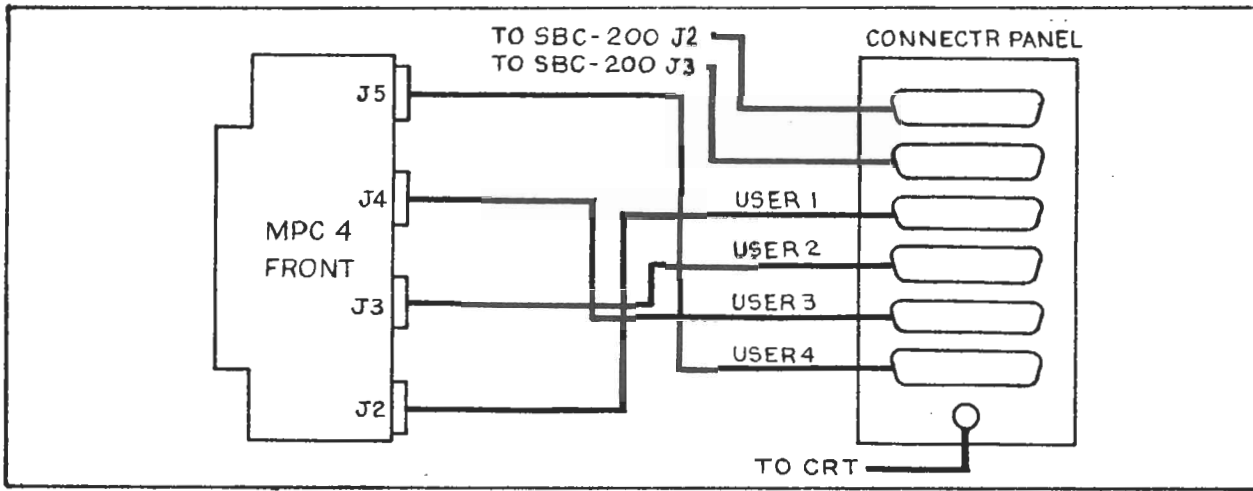


Figure 6-4. MPC-4 RS-232 Ports Interface

Once the MPC-4 and 6 six-connector panel have been installed, up to four external terminals may be connected using RS-232 DB25P connectors and 4-conductor cables with pins 1, 2, 3 and 7 connected (pin to pin) at both ends. Figure 6-5 shows the system connection.

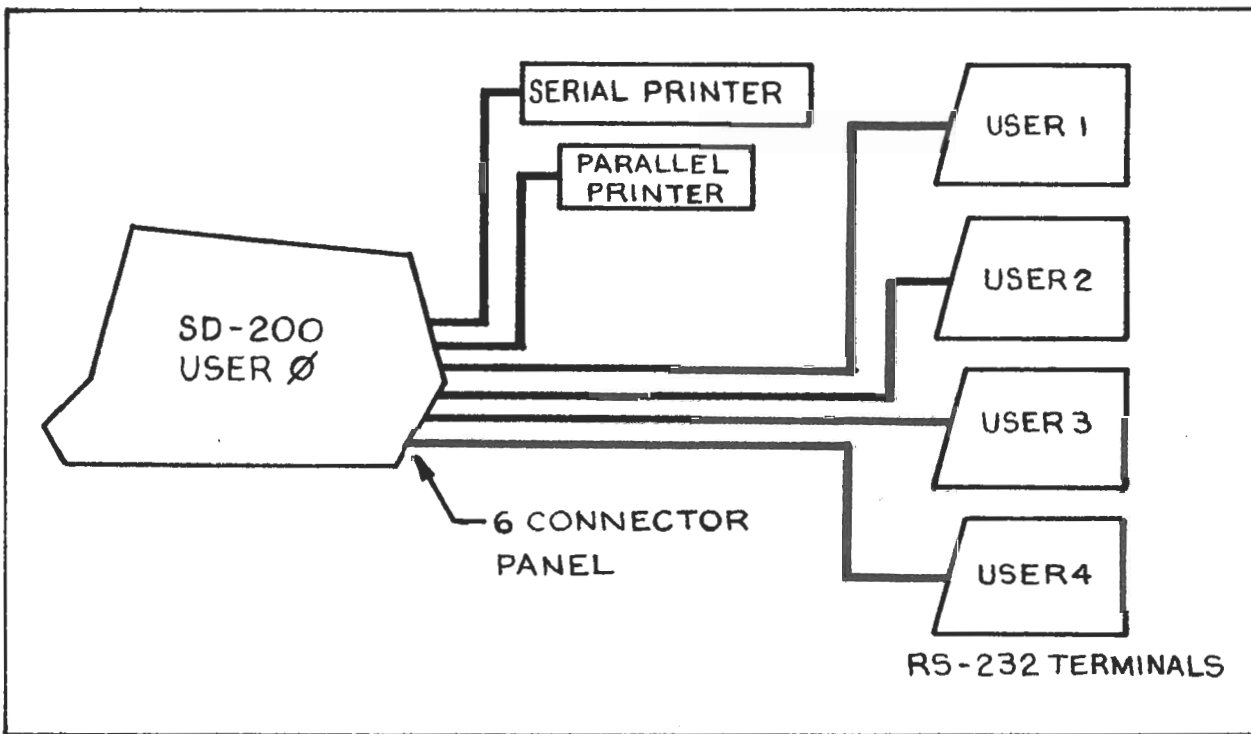


Figure 6-5. SD SYSTEMS Microcomputer Connection

6.2.3 JUMPER OPTION CONFIGURATION

Several jumpers must be installed on the SBC-200, Versafloppy II, ExpandoRAM II and VDB-8024 boards.

6.2.3.1 SBC-200 Configuration

1. Remove Pin 13 of U8 from its socket.
2. Remove jumper from X3-15 to X3-16.
3. Add a jumper from X13-2 to X13-3.
4. Add a jumper from X14-3 to X14-4.
5. The DDBIOS PROM must be V 2.3 or later.
6. Install HDBIOS PROM for fixed/removable hard disk in U37 socket.

6.2.3.2 VERSAFLOPPY II Configuration

1. Add a jumper from E8 to E9.
2. Add a jumper from E11 to E12.

6.2.3.3 EXPANDORAM II Configuration









All ExpandoRAM II boards in the system should have the following:

1. Add a jumper from E9 to E10.
2. Verify that the boards are either manufacturing level #9 or have had the modification specified on Technical Bulletin number 105. *old wait state problem*
3. All boards must have the correctly programmed 82S130 bipolar PROM in U8 for the required user partition size. For 32K user partition use the PROMS marked EX-32 (SD# 7010392). For 48K user partition use the PROMS marked EX-48 (SD# 7010393).
4. Set DIP switch (S3) as specified in Table 6-1.

6.2.3.4 VDB-8024 Configuration

1. Add a jumper from E14 to E17.

Table 6-1. ExpandoRAM II Dip Switch (S3) Settings
(Darkened Switches Are On)

		32K PARTITION	48K PARTITION
BOARD	0		
	1		
	2		
	3		

6.3 SOFTWARE CONFIGURATION

Now that the hardware has been set up for the required operating environment, COSMOS must be configured.

6.3.1 BOOTING THE COSMOS DISTRIBUTION DISKETTE

COSMOS is distributed on a double density 8 inch diskette which is formatted with 256 byte sectors. The distribution diskette will boot on any of the hardware configurations described in section 6.1 by inserting the diskette in drive A and entering "C" followed by "RETURN". COSMOS will then sign on and the prompt [A:0] will appear on the screen.

6.3.2 COSMOS CONFIGURATION PROCEDURE

A batch procedure called MCOSMOS.COM is then to be started by entering:

```
[A:0]@ MCOSMOS (cr)
```

If the configured system is to be booted from fixed removable hard disk, the following should be entered:

```
[A:0]@ MCOSMOS HD (cr)
```

The batch will ask a series of questions (see Appendix II) which must be responded to and creates a system image file called COSMOS.SYS.

6.3.2.1 System Disk Definition

This first question will be to define the system disk. Normally in a floppy disk based system this would be assigned to drive A. When the fixed/removable hard disk is in the system, the system disk may be assigned to E thru J.

6.3.2.2 Sign-On Message

A sign-on message is then requested. This message will be displayed when COSMOS is booted up. This is used to specify customer name, baud rates or any installation related information.

6.3.2.3 Partition Size

The partition size specified must match the hardware configuration as defined in section 6.2.1.

6.3.2.4 Default Printer Type

The default printer allows specifying whether a serial (J10) or parallel (J20) printer is selected after booting up. (Once the configured system is booted, the "SET" command (section 4.2.15) allows selecting either printer type.)

6.3.2.5 SBC Serial I/O Baud Rate Selection

This specifies the baud rate for the RS-232 (J10) interface on the SBC-200. This interface is used to connect with the second user's terminal in a two-user system as defined in section 6.2.2.1. In five user systems (section 6.2.2.2.) this interface connects to the serial printer.

6.3.2.6 System Type Definition

"IS MPC-4 IN SYSTEM (Y/N)" is the next question displayed. A yes (Y) response specifies that the system will support up to five users (section 6.2.2.2). A no (N) response specifies that the system will support only two users (section 6.2.2.1)

6.3.2.7 MPC-4 Channel Baud Rate Selection

If the MPC-4 board is in the system, the baud rate must be selected for each of the four serial channels. Sixteen baud rates from 50 to 19,200 are available. When the configured COSMOS is booted up, the software programmable baud rate generators on the MPC-4 are set to these selected rates.

6.3.2.8 Printer Driver Alterations

The standard printer drivers are contained in the file F9A on the distribution diskette and are written to drive a TI-810 printer. These drivers may be altered to meet special printer requirements. Once F9A is edited, the next MCOSMOS run will incorporate the changes. In parallel mode the SD internal cable (SD #38018) for the TI-810 is required. In serial mode the 'PRINTER READY' signal must be on RS-232 pin 20.

6.3.2.9 Using The Configured System

Once the MCOSMOS.COMD sequence has completed, the created COSMOS.SYS file must be written to the diskette and/or fixed/removable disk which is to be used for normal operation. Floppy disks must have been formatted using 'FORMAT' (format C or D). Hard disks must have been initialized using HDINIT.

WRTCOS is the utility used to write the configured version of COSMOS on a disk. If, for example, the removable hard disk cartridge (E:) is to be used to boot up:

[A:0] WRTCOS E:=COSMOS.SYS (cr)

The COSMOS system would then reside on the system tracks of the hard disk cartridge.

6.3.2.10 Booting The Configured COSMOS

Once the user configured COSMOS resides on the system area of the disk it can be booted and run. The standard booting procedure for floppy disk (C) is used with an additional parameter specifying the disk to boot from. Table 6-2 contains the possible disk numbers. If for example, COSMOS resides on hard disk cartridge E: the following would boot it up:

.C 4 (cr)

The "4" specifies disk E:. If the disk number is omitted, it defaults to 0.

Table 6-2. Disk Numbers Used for Boot-Up

0	A:	FLOPPY DISK
1	B:	FLOPPY DISK
2	C:	FLOPPY DISK
3	D:	FLOPPY DISK
4	E:	DISK CARTRIDGE
5	F:	FIXED PLATTER
6	G:	FIXED PLATTER
7	H:	FIXED PLATTER
8	I:	FIXED PLATTER
9	J:	FIXED PLATTER

APPENDIX A

SD SYSTEMS DISK CONFIGURATION

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	35X2	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	35X2	128	(7BH)X2K 724K
	(256 Bytes)			
C	26	77X1	128	(F7H)X2K 494K
D	26	77X2	252	(F7H)X4K 988K

APPENDIX B

COSMOS CONFIGURATION EXAMPLE

[F:0]@ /F MCOSMOS HD
SD SYSTEMS BATCH V2.0

[F:0]ERA COSBOOT.HEX

[F:0]CRUNII MAKF9

CRUN VER 2.36
MAKF9A V2.01

INITIALIZATION OF CONFIG FILES IN PROGRESS

DEFINE SYSTEM DISK (A-J) : F
ENTER SIGN-ON MESSAGE (20 CHARACTERS MAX) : SD DEMO
ENTER USER PARTITION SIZE (32,48) : 32
ENTER DEFAULT PRINTER TYPE (0-PAR,1-SER) : 1
ENTER SBC SERIAL I/O BAUD RATE:
150,300,600,1200,2400,4800,9600 : 9600
IS MPC-4 IN SYSTEM (Y/N) : Y
SELECT BAUD RATE FOR MPC-4 CHANNELS
50 - 0
75 - 1
110 - 2
134.5 - 3
150 - 4
300 - 5
600 - 6
1200 - 7
1800 - 8
2000 - 9
2400 -10
3600 -11
4800 -12
7200 -13
9600 -14
19200 -15

SELECT BAUD RATE (0-15) FOR CHANNEL 1 : 14
SELECT BAUD RATE (0-15) FOR CHANNEL 2 : 14
SELECT BAUD RATE (0-15) FOR CHANNEL 3 : 14
SELECT BAUD RATE (0-15) FOR CHANNEL 4 : 14

[F:0]ZASM B3 /K
SD SYSTEMS Z80 ASSEMBLER V3.4

PASS 1 DONE

ERRORS=0000

[F:0]ZASM B4 /K
SD SYSTEMS Z80 ASSEMBLER V3.4

PASS 1 DONE

ERRORS=0000

[F:0]ZASM F0A /K
SD SYSTEMS Z80 ASSEMBLER V3.4

PASS 1 DONE

ERRORS=0000

[F:0]ZASM F1 /K
SD SYSTEMS Z80 ASSEMBLER V3.4

PASS 1 DONE

ERRORS=0000

[F:0]ZASM F2A /K
SD SYSTEMS Z80 ASSEMBLER V3.4

PASS 1 DONE

ERRORS=0000

[F:0]ZASM F9 /K
SD SYSTEMS Z80 ASSEMBLER V3.4

PASS 1 DONE

ERRORS=0000

[F:0]LINK B2HD,B3 /CU
SD SYSTEMS LINKER V3.1

B2HD .OBJ

B3 .OBJ

UNDEFINED SYMBOLS 00

PASS 2

B2HD .OBJ ABS BEG ADDR 0100 END ADDR 0166

B3 .OBJ ABS BEG ADDR 0180 END ADDR 0181

[F:0]REN COSBOOT.HEX=B2HD.HEX

[F:0]LOAD COSBOOT
SD SYSTEMS LOADER V1.0
COM File Completed

[F:0]LINK B4,F0A,F12,OPI,F1,F11,F2A,F3,F4,F5,F6A,F6B,F7,F7B,F8,
F9,F9B,F9C,F10,F14 /CU

SD SYSTEMS LINKER V3.1

B4 .OBJ

F0A .OBJ

F12 .OBJ

OPI .OBJ

F1 .OBJ

```

F11      .OBJ
F2A      .OBJ
F3       .OBJ
F4       .OBJ
F5       .OBJ
F6A      .OBJ
F6B      .OBJ
F7       .OBJ
F7B      .OBJ
F8       .OBJ
F9       .OBJ
F9B      .OBJ
F9C      .OBJ
F10      .OBJ
F14      .OBJ

```

UNDEFINED SYMBOLS 00

PASS 2

B4	.OBJ	ABS	BEG ADDR 6E00	END ADDR 6E02
F0A	.OBJ	ABS	BEG ADDR 6E80	END ADDR 6E80
F12	.OBJ	REL	BEG ADDR 6E81	END ADDR 6EFC
OPI	.OBJ	REL	BEG ADDR 6EFD	END ADDR 7A7B
F1	.OBJ	ABS	BEG ADDR BFFF	END ADDR BFFF
F11	.OBJ	REL	BEG ADDR C000	END ADDR C0C8
F2A	.OBJ	REL	BEG ADDR C0F9	END ADDR C10F
F3	.OBJ	REL	BEG ADDR C110	END ADDR C701
F4	.OBJ	REL	BEG ADDR C702	END ADDR CE49
F5	.OBJ	REL	BEG ADDR CE4A	END ADDR D101
F6A	.OBJ	REL	BEG ADDR D102	END ADDR D67D
F6B	.OBJ	REL	BEG ADDR D67E	END ADDR DE3F
F7	.OBJ	REL	BEG ADDR DE40	END ADDR E6DE
F7B	.OBJ	REL	BEG ADDR E6DF	END ADDR EA64
F8	.OBJ	REL	BEG ADDR EA65	END ADDR EB42
F9	.OBJ	REL	BEG ADDR EB43	END ADDR EDA6
F9B	.OBJ	REL	BEG ADDR EDA7	END ADDR EE53
F9C	.OBJ	REL	BEG ADDR EE5F	END ADDR F1B2
F10	.OBJ	REL	BEG ADDR F1B3	END ADDR F405
F14	.OBJ	REL	BEG ADDR F406	END ADDR F6FC

```

[F:0]LOAD B4
SD SYSTEMS LOADER V1.0
COM File Completed

```

```

[F:0]COSGEN
SD SYSTEMS COSGEN V1.0
EOF ENCOUNTERED
SYSTEM SUCCESSFULLY GENERATED

```

[F:0]