# SD MONITOR INSTRUCTIONAL PUBLICATION

#SD 7140011
APRIL, 1981
REVISION B

SDSYSTEMS
A SYNTECH COMPANY

# SD MONITOR

# INSTRUCTIONAL

# PUBLICATION

COPYRIGHT © 1981

BY SD SYSTEMS

REVISION B
MARCH 1981

## LIMITED WARRANTY

This unit is warranted for a period of one hundred twenty (120) days from the date of shipment to be free of material or workmanship defects. This warranty is invalid if product has been misused or modified. Warranty is limited to replacement of defective parts and no responsibility is assumed for damage to other equipment.

**THERE ARE NO UNDERSTANDINGS, AGREEMENTS, REPRESENTATIONS, OR WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING ANY REGARDING FITNESS FOR A PARTICULAR PURPOSE, NOT SPECIFIED IN THE WARRANTY.**

TABLE OF CONTENTS

# SECTION I

# GENERAL INFORMATION

## 1.1 INTRODUCTION

The SD MONITOR PROGRAM was developed by SD Systems to assist the SBC-100/200 user in software and hardware debug. Commands are included to provide control and testing of memory and input/output ports. Software debug is aided with breakpoint, single step, and register examine commands. Additionally, when using the VERSAFLOPPY II disk controller board and DDBIOS software, commands are provided to support formatting diskettes, reading and writing to disk from memory, and booting up SD-OS and COSMOS.

## 1.2 SD MONITOR START UP

1. Depress: RESET
   This command starts up the SD MONITOR.
2. Response: .
3. If you system has RS-232 monitor PROM, then reply:
   CR (carriage return)
   This command causes the terminal baud rate to match the software programmable baud rate generator on the SBC-100.
4. Response: .
5. Reply:
   The operator may proceed to use any or all of the SD MONITOR commands as described in this document.

## 1.3 COMMAND SYNTAX

Each SD MONITOR command is initiated by entering an alpha letter (see table 1-1) followed by optional hexadecimal operands. The number of operands varies and when there are more than one, each must be separated by a space. Note, however, that a space is automatically inserted when the command letter is keyed in and the user must not enter a space before the first operand.

When entering an operand, only hexadecimal characters (0-9, A-F) are accepted. If a non hex character is entered, a question mark (?) is printed and the command is aborted. If, while entering an operand, a wrong number is entered, the user may start over, as only the last four digits of an operand are used. Leading zeroes are assumed if less than four digits are entered. All alpha characters must be capitals.

Examples:

```
.B 6A defaults to 006A (Hex).
.B 123456 defaults to 3456 (Hex).
```

All command lines are terminated with a carriage return (CR) which causes the command to execute. If you wish to abort a command before the carriage return (CR), then enter a period (.).

Table 1-1. SD Monitor Command Summary

| COMMAND | DESCRIPTION |
|---|---|
| B aaaa | Set breakpoint at aaaa |
| C | Boots up operating system |
| D aaaa bbbb | Display memory from aaaa to bbbb |
| E aaaa | Examine memory at aaaa |
| F aaaa bbbb dd | Fill memory from aaaa to bbbb with dd |
| G aaaa | Go to program at aaaa |
| H aaaa bbbb | Hex sum and difference |
| I pp nn | Input from port pp, nn times |
| L aaaa bbbb $cc_0$ $cc_1$ $cc_n$ | Locate string in memory from aaaa to bbbb |
| M aaaa bbbb cccc | Move memory block from aaaa to bbbb to block starting at cccc |
| O pp dd nn | Output dd to port pp, nn times |
| P pp | Examine port pp |
| R aaaa d tt ss nn | Read from disk to memory at aaaa |
| S aaaa | Single step at aaaa |
| T aaaa bbbb | Test RAM from aaaa to bbbb |
| V aaaa bbbb cccc | Verify (Compare) memory blocks |
| W aaaa d tt ss nn | Write data to disk from memory |
| X a | Examine CPU Registers (map) |
| Z f d | Format a diskette (IBM 3740) |
| Q f d | Read a diskette (IBM 3740) |

## SECTION II

## MEMORY COMMANDS

### 2.1 INTRODUCTION

The SD MONITOR contains seven (7) system memory commands providing: display, modification, block move, block fill, string location, block comparison and Random Access Memory (RAM) test.

### 2.2 MEMORY DISPLAY - "D" COMMAND

SYNTAX .D aaaa bbbb (CR)

Where aaaa is the hexadecimal value of the memory address to start displaying from and bbbb is the optional end address. If bbbb is omitted, 256 bytes are displayed starting at aaaa.

When the specified memory is displayed, the monitor waits for the user to enter a 'space' to display the next 256 bytes or a period (.) to return to the SD MONITOR command mode.

The memory is displayed 16 bytes per line, hexadecimal and ASCII form with the starting address (hex) at the beginning of each line. The ASCII output displays periods (.) for all non-printable characters.

If the user wishes to terminate a display before completion, a period (.)is entered to return to the SD MONITOR command mode.

### 2.3 EXAMINE MEMORY - "E" COMMAND

SYNTAX - .E aaaa (CR)

Where aaaa is the hexadecimal memory address at which the user wishes to begin examination and/or modification.

The purpose of the "E" command is to print the memory address and data for user examination. The user may then enter new data and a carriage return (CR) to change the data in that memory location. The monitor will then advance to the next memory location and display the contents. A period is used at any time to return to the SD MONITOR command mode.

The "up carrot" (^) character is used in lieu of the carriage return after data entry to re-examine the same address allowing verification of the change. If the "up carrot" (^) is used without data entry, the previous memory address is examined.

Note that the memory is not altered unless an operand (hex data) is entered before the carriage return or "up carrot" (^). If no hex data is entered before the carriage return, the next location is examined. If no data is entered before the "up carrot" (^), the previous location is examined.

Note also that if a period (.) is entered after a hex data operand, memory is not altered before returning to the SD MONITOR command mode.

## 2.4 FILL MEMORY - "F" COMMAND

SYNTAX - .F aaaa bbbb cc (CR)

Where aaaa is the starting address, bbbb is the ending address and cc is the fill data.

The purpose of the "F" Command is to cause the hex data (cc) to write to every Random Access Memory (RAM) location from aaaa through bbbb.

## 2.5 MOVE MEMORY - "M" COMMAND

SYNTAX - .M aaaa bbbb cccc

Where aaaa, bbbb and cccc are hexadecimal memory addresses.

The purpose of the "M" Command is to cause the memory block starting at address aaaa and ending at address bbbb to copy to the memory block starting at cccc. bbbb must be greater than aaaa.

## 2.6 LOCATE STRING - "L" COMMAND

SYNTAX - .L aaaa bbbb $cc_0$ $cc_1$ $cc_2$....$cc_n$ (CR)

Where aaaa and bbbb are hex memory addresses and $cc_0$ through $cc_n$ are single byte hex data values.

The purpose of the "L" Command is to search memory starting at aaaa and ending at bbbb for the string (up to 6 bytes) specified by the hex values $cc_0$ through $cc_n$.

Each time the string is located, the address of the first byte is printed on the console.

The search is terminated at any time by entering a period (.) on the console.

## 2.7 MEMORY TEST - "T" COMMAND

SYNTAX - .T aaaa bbbb (CR)

Where aaaa and bbbb are hexadecimal memory addresses.

The purpose of the "T" Command is to perform a modified incrementing memory address test on the Random Access Memory (RAM) starting at aaaa and ending at bbbb.

All errors are reported on the console with the memory address, data written and data read back being displayed for all bad locations.

At the end of each complete pass on the specified memory block, a "P" is displayed on the console.

A period (.) must be entered to terminate the memory test.

## 2.8 VERIFY MEMORY - "V" COMMAND

SYNTAX - .V aaaa bbbb cccc (CR)

Where aaaa, bbbb and cccc are valid hexadecimal memory addresses and bbbb is greater than aaaa.

The purpose of the "V" Command is to compare the memory block starting at aaaa and ending at bbbb with the memory block starting at cccc.

Any differences in the memory blocks are reported on the console with the first block address with data followed by the second block address with data.

The verification is terminated by entering a period (.) at any time.

## SECTION III

## INPUT/OUTPUT COMMANDS

### 3.1 INTRODUCTION

The SD MONITOR contains three commands for reading and writing to input and output ports: "I" Command for input from ports, "O" Command for output to ports and "P" Command for port examination.

### 3.2 INPUT FROM PORT - "I" COMMAND

SYNTAX - .I pp nn (CR)

Where pp is the input port address, nn is the number of times the port is read.

The purpose of the "I" Command is to read the input port "pp" and display the data nn times. If "nn" is omitted, the default is one (1) read. If "nn"=0, the port is continuously read and displayed until a period (.) is entered to terminate the operation.

### 3.3 OUTPUT FROM PORT - "O" COMMAND

SYNTAX - .O pp dd nn (CR)

Where pp is the port address, dd is the data to output, nn is the number of times to output the data.

The purpose of the "O" Command is to write the data defined by dd to port pp. The data is written the number of times defined by nn. If nn is omitted, the default is one (1). If "nn"=0, the data is continuously written until a period (.) is entered to terminate the operation.

### 3.4 PORT EXAMINE - "P" COMMAND

SYNTAX - .P pp (CR)

Where pp is the hexadecimal port number (address) to begin examining and/or modification.

After the command is entered, the specified input port address and data is printed. The user may then enter new data and a carriage return (CR) to change the data in that input/output port. The monitor will then advance to the next input/output port number and display the contents. A period can be entered at any time to return to the SD MONITOR command mode.

The "up carrot" (^) character may be used in lieu of the carriage return after data entry to re-examine the same input/output port. If the "up carrot" is used without data entry, the previous input/output port is examined.

Note that the port is not altered unless an operand (hex data) is entered before the carriage return or "up carrot" (^). If no hex data is entered before the carriage return, the next input/output port is examined. If no hex data is entered before the "up carrot" (^), the previous input/output port is examined.

Note also that if a period (.) is entered after a hex data operand, the input/output port is not altered before returning to the SD MONITOR command mode.

## SECTION IV

## PROGRAM CONTROL COMMANDS

### 4.1 INTRODUCTION

The SD MONITOR provides several commands designed to facilitate rapid Z80 software development.

The user has total control of all the Z80 Central Processing Unit (CPU) Registers via a register map in memory. This map is loaded into the Z80's registers each time a program is executed from the monitor. Conversely, each time a breakpoint is encountered, the Z80 CPU Registers are saved in the register map and displayed on the console.

During single step operation, the registers are loaded and saved in between each instruction step.

The user may also set up the register map using the "E" Command. Table 4-1 contains the memory addresses of each of the Z80 registers in the map.

### 4.2 BREAKPOINT - "B" COMMAND

| SYNTAX - .B aaaa (CR)<br><br>Where aaaa is the breakpoint address to insert. |
|---|

The purpose of the "B" Command is to insert a software breakpoint in the user's RAM based program. The breakpoint consists of a three byte jump instruction replacing the user's code at the specified address. When the breakpoint is encountered, the user's code is restored and the registers are displayed on the console.

If the user inserts a breakpoint while another one is still in memory, the previous one is automatically removed before inserting the new one.

If "aaaa" is omitted, any existing breakpoint is removed. When the user inserts a breakpoint, executes a program, but doesn't encounter the breakpoint, the system reset is used to return to the SD MONITOR. In that case, the user must execute at the breakpoint address in order to remove it.

When the program encounters a breakpoint, the CPU registers are displayed. After that, the monitor enters the single step mode (see paragraph 4.4). The user may then enter carriage return to single step or period (.) to return to the SD MONITOR command mode.

## 4.3 "GO" TO PROGRAM - "G" COMMAND

SYNTAX - .G aaaa (CR)

Where aaaa is the hexadecimal address at which the user wishes to begin execution.

The purpose of the "G" Command is to begin execution of a program anywhere in memory (aaaa).

When the "G" Command is entered, the Random Access Memory (RAM) register map is loaded into the Z80's registers before executing the address aaaa.

A program is resumed after a breakpoint (or single step) by omitting the "aaaa". This causes execution to begin at the address defined by "PC" in the register map (as displayed by X Command).

## 4.4 SINGLE STEP - "S" COMMAND

SYNTAX - .S aaaa nn (CR)

Where aaaa is the address to begin stepping and nn is the number of steps to trace.

The purpose of the "S" Command is to single step through Random Access Memory (RAM) based programs, displaying all of the CPU registers after each step. If the nn is omitted, the default value is 1. If the aaaa is omitted, the default value is the previous "PC" loaded into the Random Access Memory (RAM) register map either by a breakpoint, the memory examine "E" Command, or a previous single stepping sequence.

When the "S" Command is entered, the Z80 registers are displayed with the registers labeled over each column. The user may then step one instruction by entering carriage return. If the user enters a "space" the monitor steps eleven steps along with a heading and register display. The user must enter period (.) to return to the SD MONITOR command mode.

Once back in the SD MONITOR command mode, all commands may be used. Single stepping is resumed at the address left off by entering "S" (CR) with no operands.

## 4.5 REGISTER EXAMINE - "X" COMMAND

SYNTAX - .X a b (CR)

Where "a" is the heading print option and "b" is the length option.

The purpose of the "X" Command is to examine the Random Access Memory (RAM) register map at any time.

If "a" is omitted, the registers are displayed without a label heading.

If "a" is non-zero, a label heading is printed above the registers.

If "b" is entered, it sets the register display length.

If b=0, it sets the display to short mode; if b=1, it sets the display to the long mode (short mode only displays PC and AF). This effects breakpoint and single step in addition to the X Command.

The display mode remains set until the X Command is used to change it.

Note that if "b" is omitted, the display mode is not altered.

The user must remember also that the Random Access Memory (RAM) register map is only altered by a breakpoint, single step or manual changes via the "E" Command.

## 4.6 HEX ARITHMETIC - "H" COMMAND

SYNTAX - .H aaaa bbbb (CR)

Where aaaa and bbbb are any two hexadecimal numbers.

The purpose of the "H" Command is to display the hexadecimal sum and difference of aaaa and bbbb. The sum aaaa + bbbb is displayed first preceeded by a "+". The difference (aaaa - bbbb) is displayed second, preceeded by a "-".

Table 4-1. SD Monitor Register Map

| MEMORY ADDRESS (HEX) | REGISTER | DESCRIPTION |
|---|---|---|
| FFE6 | SP (LSB) | Stack Pointer Lower Half |
| FFE7 | SP (MSB) | Stack Pointer Upper Half |
| FFE8 | IY (LSB) | Index Reg IY Lower Half |
| FFE9 | IY (MSB) | Index Reg IY Upper Half |
| FFEA | IX (LSB) | Index Reg IX Lower Half |
| FFEB | IX (MSB) | Index Reg IX Upper Half |
| FFEC | L' | Alternate L |
| FFED | H' | Alternate H |
| FFEE | E' | Alternate E |
| FFEF | D' | Alternate D |
| FFFO | C' | Alternate C |
| FFF1 | B' | Alternate B |
| FFF2 | F' | Alternate Flags |
| FFF3 | A' | Alternate A |
| FFF4 | L | L Register |
| FFF5 | H | H Register |
| FFF6 | E | E Register |
| FFF7 | D | D Register |
| FFF8 | C | C Register |
| FFF9 | B | B Register |
| FFFA | IF | Interrupt enable flag (04=enabled) |
| FFFB | I | I Register |
| FFFC | F | Flags |
| FFFD | A | A Register |
| FFFE | PC (LSB) | Program Counter Lower Half |
| FFFF | PC (MSB) | Program Counter Upper Half |

## SECTION V

## DISK UTILITY COMMANDS

### 5.1 INTRODUCTION

The SD MONITOR provides several commands that are useful in a disk based system. These commands are only operative when a VERSAFLOPPY II disk controller board and the SBC-200 DDBIOS PROM are present in the SD SYSTEMS Microcomputer.

These commands operate on both mini and standard drives. All parameters relating to drive type are contained in the BIOS PROM.

### 5.2 SD-OS BOOT UP - "C" COMMAND

SYNTAX - .C aa (CR)

Where aa is the drive number 0 through 9.

| Drive Number | Drive Letter |
|---|---|
| 0 | A |
| 1 | B |
| 2 | C |
| 3 | D |
| 4 | E |
| 5 | F |
| 6 | G |
| 7 | H |
| 8 | I |
| 9 | J |

The purpose of the "C" Command is to boot up the SD SYSTEMS Operating System from the specified drive. If the drive number is not specified in the SD MONITOR .C Command, it defaults to 0 (drive A).

NOTE

The SDOS Disk Operating System Diskette must be in the defined drive before entering this command.

Once the SDOS boot up is complete, the SD MONITOR is exited and the SD SYSTEMS Disk Operating System (SDOS or COSMOS) is activated causing "[A]" to display on the console.

## 5.3 READ FROM DISK - "R" COMMAND

SYNTAX - .R aaaa fd tt ss nn

Where aaaa is a valid RAM memory address, f is the format type, d is a valid drive number, tt a valid track number, ss a valid sector number and nn is the number of 128 byte sectors to read (up to FF).

The purpose of the "R" Command is to provide a means of reading data starting at the specified drive, track and sector into the RAM buffer (starting at aaaa). The number of bytes is determined by nn times 128.

## 5.4 WRITE TO DISK - "W" COMMAND

SYNTAX - .W aaaa fd tt ss nn

Where aaaa is a valid memory address, f is the format type, d is a valid drive number (0 to 3), tt is a valid track number, ss is a valid sector number, and nn is the number of 128 byte sectors to be written from memory (up to FF).

The purpose of the "W" Command is to provide a means of writing data starting at the specified drive, track and sector from the memory starting at aaaa. The number of bytes written is determined by 128 times nn.

## 5.5 FORMAT A DISK - "Z" COMMAND

```
SYNTAX - .Z f d (CR)

Where "f" specifies the format type represented as follows:

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

Where "d" specifies a drive from 0 to 3.
```

The purpose of the "Z" Command is to initialize a diskette with the IBM 3740 format. If the double sided jumper or the VERSAFLOPPY II is inserted, both sides of the diskette are formatted.

## 5.6 READS A DISK - "Q" COMMAND

```
SYNTAX - .Q f d (CR)

Where "f" specifies the format type represented as follows:

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

Where "d" specifies a drive from 0 to 3.
```

The purpose of the "Q" Command is to read IBM 3740 formatted diskette data. It is important to note that this command doesn't work with the Versafloppy II board.