SSM Z-80 MONITOR
VERSION 1.10

INSTRUCTION MANUAL

January 19, 1981

SSM Microcomputer Products
2190 Paragon Drive
San Jose, California 95131

PHONE: (408) 946-7400
TWX: 910-338-2077
TELEX: 171171
DDD: (408) 946-3644 (110 BAUD)

TABLE OF CONTENTS

SECTION PAGE

Written By: Malcolm T. Wright
Edited By: Dan Fischler

1.0 INTRODUCTION

The SSM Z-80 MONITOR constitutes a powerful central operating system for a Z-80 based microcomputer system. The Z-80 MONITOR provides many of the features of the original Monitor V1.0 (SSM's 8080 based monitor) with the addition of two new commands for testing I/O and full Z-80 register display/modify capability. The monitor will support a minimum system of 2K bytes of RAM, an I/O board, and a Z-80 CPU, but can be easily expanded to a disk-based system running CP/M with up to 48K of RAM.

The SSM Z-80 Montior is contained within 2048 bytes of ROM and requires less than 96 bytes of scratchpad RAM during operation. The monitor will automatically allocate the scratchpad RAM at the top of a minimum of 2K bytes of RAM, and will change this allocation as more memory is added to the user's system.

The monitor provides an interface to system peripheral devices through externally referenced subroutines and logical I/O assignments. Operator commands are available for examining and changing memory, executing user programs with or without breakpoints, examining or changing CPU register contents, input or output data from/to any I/O port, assigning physical devices to logical devices, etc.

CP/M is a registered trademark of Digital Research, P.O. Box 579, Pacific Grove, California 93950

Z-80 is a registered trademark of Zilog Inc., 10340 Bubb Road, Cupertino, California 95014

8080 is a registered trademark of Intel Corp., 3065 Bowers Avenue, Santa Clara, California 95051

2.0 HARDWARE SET-UP

2.1 General Hardware

The SSM Z-80 MONITOR is provided in one 2048 x 8 bit ROM (2716). The starting address for the Z-80 monitor is F000 Hex.

The monitor needs a minimum of 2048 bytes of contiguous RAM memory starting at address 0000. On initialization of the monitor program, up to 96 bytes of scratchpad data are written into the top of RAM. The Z-80 CPU board or ROM board must have a vector jump circuit unless a front control panel is used so that reset or power-on clear operations will force the CPU to F000 for proper start-up. If a front panel is used then examine F000 and press RUN.

The monitor supports any one of four possible console configurations.

I---Console I/O device on ports 0 & 1.

```
    Status port = 0, Data port = 1
    Status bits:
       D0 = Data-available, low=data received
       D7 = Data-acknowledge, low=not busy
```

Console configuration I uses the most common I/O port addresses established by MICROSOFT for the S-100 bus.

II---Console I/O device on ports 2 & 3.

```
    Status port = 2, Data port = 3
    Status bits:
       same as configuration I.
```

Console configuration II is set-up to take advantage of the serial port addressing capability of SSM's IO4 board. The listing device (line printer) is preset to this port address too, unless redefined by the user with the "AL" (Assign List device) command.

III---Console input on ports 6 & 7.

```
    Status port = 6, Data port = 7
    Status bits:
       D0 = Data-available. low = data received
       Console output is the VB1B or VB1C
       VB1 address = E000
```

Console configurations III & IV are set up to support SSM's video boards, either the VB1B/C or VB3.

IV---Console input on ports E0 & E1.

Status port = E0, Data port = E1
Status bits:
same as configuration III.
Console output is the VB3.
VB3 address = C000
CRT control port = D0 to DF

On power-up, the console is automatically determined by the monitor with no special hardware required.

2.2 IO4 Settings (References from SSM IO4 Manual)

The IO4 board can be used to support three of the four possible console configurations. Consoles I & II become SERIAL terminals and console III uses the IO4's PARALLEL port for a parallel keyboard.

Serial A (Console I)

a) Set switch S3 so all positions are closed (on).
b) Set switch S2, either all open(off) or per your terminal's requirements.
c) Jumper W2, pin 4 to pin 9
d) Jumper W2, pin 5 to pin 16.
e) Install a 74LS368 (or 74368) into U18.
f) Jumper W3 (baud rate), pins 12 & 11 to the rate used by your terminal(pins 1 up to pin 10, only one).
g) Connect RS232 or Current-loop interface to J2.

Serial B (Console II)

a) Set switch S3 so all positions are closed (on).
b) Set switch S1, either all open(off) or per your terminals needs.
c) Jumper W1, pin 4 to pin 9.
d) Jumper W1, pin 5 to pin 16.
e) Install a 74LS368 (or 74368) into U16.
f) Jumper W3 (baud rate), pins 14 & 13 to the rate used by your terminal(pins 1 up to pin 10, only one).
g) Connect RS232 or Current-loop interface to J1.

Parallel A & B (Console III)

a) Connect a 1K ohm resistor from J6,pin 1 to J6,pin 14.
b) Feed positive keyboard strobe to J4,pin 1 (TTL level only).
c) Connect J4,pin 2 to J6, pin 9.

d) Connect 7 or 8 data lines from keyboard to J4.

```
D0(LSB)=J4,pin 9
D1     =J4,pin 6
D2     =J4,pin 10
D3     =J4,pin 5
D4     =J4,pin 11
D5     =J4,pin 4
D6     =J4,pin 12
D7(MSB)=J4,pin 3  (sometimes not used)
```

e) Set Switch S4:
positions A3 thru A7 closed (on)
positions A1 and A2 open (off).

2.3 VB1B/VB1C Settings (References from VB1B/VB1C Manual)

The VB1 board is used in console configuration III. The console input for this configuration must be supported by a parallel output type of keyboard (ASCII) into an I/O board set for ports 6 & 7.

a) Set switch S1:
A10 thru A12 set closed (on).
A13 thru A15 set open (off).
Size position should be closed (on) for 64 characters.
Graphics position, user's choice.
b) VB1C only
Wirewrap E2 to E3.

2.4 VB2 Settings (References from VB2 Manual)

The VB2 is used in console configuration I. The board will be addressed at ports 0 and 1.

a) Set switch S1 all positions closed (on).
b) Set switches S2b and S2c with the desired vertical position.
c) Set switch S2d (position "k") to the correct strobe polarity for your keyboard:

Closed (on)	Negative-going strobe
Open (off)	Positive-going strobe

d) Jumper the status header (SH) for the proper status bits:

Jumper pin 1 to pin 16
Jumper pin 8 to pin 10

e) Set switch S2a to the desired video output (black-on-white or white-on-black).

2.5 VB3 Settings (References from the VB3 Manual)

The VB3 board is used in console configuration IV. The console input for this configuration uses the keyboard input on the VB3 at address E0 & E1.

a) Set switch S1 all positions open (off), unless you are using external sync capability.
b) Set switch S2:
   positions 2 thru 4 open (off).
   position 1 closed (on).
c) Set switch S3:
   positions 1,2,3 & 8 open (off).
   positions 4 thru 7 closed (on).
d) Set switch S4:
   positions 1,2,4,5,7 open (off).
   positions 3,6,8 closed (on).
e) Wirewrap E5 to E6 for the CB2.

2.6 CB2 Settings (References from the CB2 Manual)

The following switches and wirewrap options should be set for power-on/reset jump operation into the SSM Z-80 Monitor on the CB2.

a) Vector Jump Option (See CB2, section 4.1.4)
   1) Wirewrap E18 to E19
   2) Wirewrap E24 to E25

b) Addressing On-board ROM
   1) Set switch SC
      positions 1,4 thru 7 set open (off)
      positions 2 & 3 set closed (on)
   2) Set switch SD
      All positions closed (on)

c) Select ROM Type
   1) Set switch SE
      position 1 & 2 set open (off)
   2) NOTE: Some 2716 EPROM's draw more current than the 680 ohm resistors (R7 and R9) can provide. To provide more current drive to U16, solder a short wire across R9 on the back of the CB2. Be sure that switch SE, positions 1 and 2 are open.

d) Install the ROM monitor into socket U16.

3.0 MONITOR START-UP

The CPU must jump to F000 to start the SSM Z-80 MONITOR. The monitor then scans all four console status ports to see if one has data. If the data found is a carriage-return, then that console configuration is selected. Even if all four possible consoles exist on your Z-80 system, only the one with a carriage-return will be selected. Be sure to type several carriage-returns on power-up of your microcomputer.

When a console is selected, the monitor will print out:

```
SSM Z80 MONITOR V1.10
```

followed by a decimal-point. An asterisk is printed, if an illegal command or character is typed.

The Z-80 system must have a minimum of 2K of RAM memory starting at location 0000 (HEX). If more than 2K of RAM is present in the system for the monitor to use, it must be contiguous starting at location 0000 (HEX). If only 1K of RAM is available, it can be addressed to start at 0400 (HEX) as a special case and the Z-80 monitor will operate.

4.0 MONITOR COMMANDS

Note: Spaces have been inserted into the "format:" examples for clarification of the elements of the command, and should not be typed. <CR> = carriage-return.

4.1 'A'-----Assign I/O devices

```
Format: A  N1 = N2      (no carriage-return)
        N1 = Logical device (C,R,P,L)
             C = Console
             R = Reader
             P = Punch
             L = Listing device
        N2 = Physical device (0,1,2,3)
             Picked from I/O table (IOTAB)
```

A physical device (N2) is assigned to a logical device (N1). N1 may be any of the four system devices like console, reader, punch or listing device. Only the first character of the device name is used. N2 may be 0,1,2 or 3. The actual physical device assigned has been preset by the IOTAB table in the system configuration section of the monitor. IOTAB consists of 6 subtables of two byte addresses pointing to the I/O subroutines. The console configuration uses 3 of these subtables for defining status, input and output routine addresses.

EXAMPLE:

```
     AC=0    Select console I.
     AR=1    Select console input II as a reader.
     AL=3    Select VB1 as a listing device.
```

4.2 'D'-----Display memory

```
Format: D N1 , N2 <CR>
        N1 = Starting address in hex.
        N2 = Ending address in hex.
```

Memory from the starting address to the ending address is displayed on the logical console output device. Data bytes are displayed in hexadecimal, 16 bytes per line. The beginning address of each line is displayed before the data, on the left, as four characters in hex. Typing just N1 followed by a comma or space and a carriage-return will display one location only. N1 and N2 can be separated by a comma or a space. Input of leading zeros is not necessary.

The display function can be halted and continued by hitting any keyboard character except a capital 'S'. The capital S character is used to exit the display operation before it has completed the area (N1 to N2) requested by the user.

EXAMPLE:

```
     D0,F <CR>        Display from 0000 to 000F
     D100,3FF <CR>    Display from 0100 to 03FF
     D8000, <CR>      Display byte at 8000
```

4.3 'F'-----Fill memory

```
Format: F N1 , N2 , N3 <CR>
          N1 = Starting address in hex
          N2 = Ending address
          N3 = Data used in fill
```

Memory from the starting address (N1) to the ending address (N2) is filled with data (N3). If the ending address is smaller than the starting address then only the one starting location is filled. Input of leading zeros is not necessary.

EXAMPLE:

```
     F0,FFF,0 <CR>   Fill memory from 0000 to 0FFF with 00.
     F100,,8  <CR>   Fill location 0100 with 08.
```

4.4 'H'-----Hexadecimal arithmetic

```
Format: H N1 , N2 <CR>
```

The sum (N1+N2) and difference (N1-N2) of N1 and N2 are displayed in hexadecimal on the logical console output device. The sum will be the left four digits, the difference will be the right.

EXAMPLE:

```
     H3 4 <CR>   Enter numbers.

     0007 FFFF   Results.
```

4.5 'I'-----Input port data

```
Format: I N1 ,         (no carriage-return)
          N1 = Port address (00 to FF)
```

Input from port N1 the data present. Displays 8 bits of input data in hex from any port address.

EXAMPLE:

```
     I20,       Input from port 20 Hex.

     FF         Data found by the monitor was FF.
```

4.6 'M'-----Move memory data

```
Format: M N1 , N2 , N3 <CR>
          N1 = Starting address of data to be moved.
          N2 = Ending address of data.
          N3 = Destination address for data.
```

Data from the starting address (N1) to the ending address (N2) are moved to memory beginning at the destination address(N3). If N2 is equal to or smaller than N1 only the byte at N1 is moved. If N3 is between N1 and N2, the data from N1 to N3 will of course be repeated until the destination field is filled. Input of leading zeros is not necessary.

EXAMPLE:

```
     M0,FF,100 <CR>  Duplicate 0000 to 00FF from 0100 to 01FF
```

4.7 'O'-----Output data to port

```
Format: O N1  , N2 <CR>
              N1 = Port address (00 to FF)
              N2 = Data to be sent
```

Output data to the port N1. Input of leading zeros is not necssary.

EXAMPLE:

```
     O1,41 <CR>   Output to port 01 a 41 hex.
```

4.8 'S'-----Substitute memory

```
Format: S N1  , XX-N2  , . . (XX indicates monitor's response)
              N1 = Memory address to change
              N2 = New data in hex
              XX = Old data displayed by monitor
```

The byte at address (N1) is displayed on the logical console device followed by a "-" character. The operator responds with one or more characters from the logical console input device. If the input character is a space or comma, no change is made to the displayed memory value and the contents of the next location is displayed, etc. If one or more hexadecimal digits are input before the space or comma, the last two digits to the left of the comma will replace the displayed value(XX) in memory. A carriage return by the user or an error will terminate the command line. For every 8 locations displayed, an automatic carriage return will be generated to prevent line overflow. Input of leading zeros is not necessary.

EXAMPLE:

```
     S100,11-,CD-C3,21-,F0- <CR>
```

Replace data at 101 with a C3 (jump instruction). Only one byte was changed, and three were skipped (unchanged).

5.0 SOFTWARE DEBUG COMMANDS

5.1 'G'-----Goto with breakpoints

```
Format: G N1 , N2 , N3 <CR>
        N1 = Start execution address (GOTO address)
        N2 = Breakpoint #1
        N3 = Breakpoint #2
```

If N1 (goto address) is specified, control is transferred to N1 immediately upon a carriage-return. If N1 is not specified, control is transferred to the address of the last encountered breakpoint, after program status (CPU registers and flags) is restored.

If N2 (breakpoint address) or N3 (2nd breakpoint address) is specified, a RST 1 instruction will replace one byte at each of the specified addresses. These addresses **must** be the opcode address (first byte) of a one, two, three or four byte instruction. If breakpoints are specified, a jump instruction is stored at location 0008 to return control to the monitor when a breakpoint, or any RST 1 instruction, is executed. When a breakpoint is executed, the monitor will save the program and restore the original bytes replaced by any known breakpoints. An asterisk is printed on the console followed by the breakpoint address that was encountered. Twelve bytes of the users stack are used in this process. The program counter in the saved program status is decremented, so that program execution may be resumed with the instruction formerly replaced by the breakpoint. Monitor commands may then be used to display/modify memory or CPU registers, etc.

When the monitor is entered normally, by other than breakpoint execution, recording of existing breakpoints is destroyed. Therefore, if breakpoints are set, but not executed before the monitor is entered, the contents of the bytes containing those breakpoints must be manually restored.

RST 1 instructions other than known breakpoints may be used as pseudo-breakpoints, subject to certain restrictions. The jump instruction must be stored at location 0008 by previously setting a normal breakpoint. RST 1 instructions other than known breakpoints may be executed through normal program execution (RST 1 stored as part of the executing program) or instruction jam (interrupt). When such a RST 1 instruction is encountered, the monitor saves the program status and clears known breakpoints. However, the program counter in the saved program is not decremented, so program execution may be resumed at the next instruction.

```
EXAMPLE:
     G100 <CR>        Go to address 0100 hex.
                      No breakpoints set.
     G200,259 <CR>    Go to Address 0200 hex.
                      One Breakpoint at 0259.
     G <CR>              Continue from last breakpoint address.
                         No new breakpoints set.
     G,200 <CR>          Continue from last breakpoint address.
                         Set one breakpoint at 0200.
```

NOTE: Of course, you CAN NOT place breakpoints in a program set in ROM, but only in RAM.

5.2 'X'-----Examine/Modify registers

For clarification the "X" command will be broken into three types.

```
  Format: X <CR>       (Type I)
          X' <CR>      (Type II)
          X Kl         (Type III)  No carriage-return
```

Kl = One letter name for the register preceded by a prime symbol, if the alternate registers are wanted.

This command allows the user to examine/modify the contents of the Z-80 CPU registers from the last encountered breakpoint displayed. There are three types of commands that can be given.

Type I:
"X" followed by a carriage-return will display half of the Z-80's registers and flags. The flag bits of the flag register are indivually broken out as a zero or one for the sign bit (S:), Zero bit (Z:), Unused bit (X:), Half-carry bit (H:), Overflow/parity bit (V:), Add/Subtract bit (N:) and Carry bit (C:). The registers displayed will be A, B, C, D, E, H, L, M (memory address), P (program counter address), S (stack address) and I (interrupt register).

Type II:
"X" followed by a single-quote(prime) and a carriage-return will display the alternate half of the Z-80's registers and flags. The alternate flag bits are displayed like the Type I command. The registers displayed will be A, B, C, D, E, H, L, M (memory address), X & Y (index registers) and R (refresh register).

Type III:
"X" followed by a letter (Kl) for a specific Z-80 register will display that register. After the register's content is displayed, the operator may change it by supplying the new value followed by a space or comma. If no new value is entered, then the old value is retained and the next register is displayed. The command is terminated by a carriage-return, or display/modication of the I or R register. The flag registers can be changed by typing XF or X'F which the monitor will respond by displaying the HEX value of that register.

6.0 EXTERNALLY REFERENCED SUBROUTINES

Several externally referenced subroutines and entry points are provided for possible use by the user's software. These routines have entry addresses relative to the beginning of the monitor (F000).

6.1 Initialize Monitor (BEGIN)

Address = F000

Normal entry point to cold start the monitor. The I/O configuration is scanned for a carriage-return from the user. The VB3 board is also initialized.

6.2 Console Input (CI)

Address = F003

Console input. One character (byte) is read from the logical console input device and returned in register A. All registers other than A & F are preserved.

6.3 Reader Input (RI)

Address = F006

Reader input. One byte is read from the logical reader device and returned in register A. All registers other than A & F are preserved.

With a certain physical reader selection, this subroutine will "time out" and return with no data and the carry flag set. See section 8 for details.

6.4 Console Output (CO)

Address = F009

Console output. The byte in register C is output to the logical console output device. All registers other than A & F are preserved.

6.5 Punch Output (PR)

Address = F00C

Punch output. The byte in register C is output to the logical punch device. All registers other than A & F are preserved.

6.6 List Output (LO)

Address = F00F

List output. The byte in register C is output to the logical list device. All registers other than A & F are preserved.

6.7 Console Status (CSTS)

Address = F012

Console status. The logical console input device is checked for input availability. Register A is set to zero and the zero flag is set true if no input is available. Register A is set non-zero and the zero flag is set false if a character is available. All registers other than A & F are preserved.

6.8 Check I/O Byte (IOCHK)

Address = F015

The current setting of the IOBYT is returned in register A. IOBYT is the byte of RAM used by the monitor to record the current logical device/physical device assignments. The bits of the IOBYT are as follows:

Field	Function
Bits 0,1	Indicates the physical device currently assigned to the logical console device.
Bits 2,3	Indicates the physical device currently assigned to the logical reader device.
Bits 4,5	Indicates the physical device currently assigned to the logical punch device.
Bits 6,7	Indicates the physical device currently assigned to the logical list device.

Every two bit pair presents the binary number 0 thru 3 which is used by the monitor to pick one of four possible devices. All registers other than A & F are preserved.

6.9 Set I/O Byte (IOSET)

Address = F018

The contents of register C are stored in the IOBYT, thus altering the logical/physical device assignments. All registers are preserved.

6.10 Find Top of Memory (MEMCK)

Address = F01B

The upper limit of RAM available to user programs is returned in registers B (most significant) and A (least significant). The address returned is that of the first byte not available to the user. All registers other than A, B and F are preserved.

6.11 Output String (STRNG)

Address = F01E

The string of characters pointed to by registers H & L is output to the logical console output device. The character string is terminated by a null character (null = 00 Hex), or after a character with bit 7(MSB) set (last character is output with bit 7 set). Registers B, D, E are preserved.

6.12 RE-Enter The Monitor (REENT)

Address = F021

Alternate entry point (warm start) to the monitor. The current I/O configuration is not altered when the monitor is entered at this point. This is the common entry point for external routines after the monitor has been initialized once (see 6.1).

6.13 Special Vector

Address = F024

The special vector input provides an entry point for several subroutines within the System Configuration Package. When JVTR is entered, register A must be set by the user to select the desired subroutine.

6.131 Scratch Address (ADSCR)

Register A = 00
All registers except A, F, H & L preserved.

The address for the top of scratch memory used by the monitor is returned in registers H&L.

6.132 VB3 Constants (RWCL)

Register A = 01
All registers except A, F, H & L preserved.

The address for the two constants (Row & Column) used by the VB3 is returned in registers H & L. The column value is at the address returned in H & L, with the Row value at H & L +1. The two constants only have to be changed if the VB3 is initialized by the user to a format other than 80 X 18. (See 6.133 to set new format).

The monitor initially sets:
Address H&L points to the value 50 hex(80 decimal)
Address H&L+1 points to the value 12 hex(18 decimal)

6.133 Initialize VB3 (VB3IZ)

Register A = 02
All registers except A, F, H & L preserved.

A table of seven parameters is passed to the VB3 board with this subroutine to change its initialization from 80 X 18 to any user defined format. H&L must be set to the address of the beginning of the table of the seven parameters to be passed.

The monitor initializes this table as follows:
70 hex-----Hcount-1, length of raster line.
53 hex-----Interlace bit + Sync parameters
5D hex-----Raster lines/Char. + No. of Char. across
11 hex-----No. of rows down the screen minus one
03 hex-----Raster fill-in constant
14 hex-----Top vertical margin in raster lines.
11 hex-----Sets which line number is bottom of screen

See the VB3 manual for additional information and explaination of VB3 parameters.

6.134 **Output to the VB3 (VB3OUT)**

Register A = 03
All registers except A, C, F, H & L preserved.

Output to the VB3 board. The data must be in register C and the Column constant address must be pointed to by H & L. The Row constant must be at H & L + 1 (See 6.132 for more information.)

This output subroutine gives the ability of running multiple VB3's which have different formats(80X18, 80X24, 80X32, etc.) and different Bank-Switch port addresses and yet have them all at address C000 in memory.

TYPICAL ROUTINE:
Load Reg-C and CALL VB31.

```
VB3-1:  OUT     KSTAT   ;TURN-ON VB3#1
        LXI     H,VB3XY ;SET ROW/COL DATA ADDRESS
        MVI     A,3     ;VECTOR TO VB3OUT
        CALL    0F024H  ;CALL VECTOR
        OUT     KDATA   ;TURN-OFF VB3#1
        RET
VB3XY:  DB      80      ;COLUMN VALUE
        DB      18      ;ROW VALUE
```

If the column/row data (80X18,etc.) is the same for all the VB3 boards then use RWCL (see section 6.132) to change the column/row values within the monitor to meet your needs and replace the above line as follows:

```
                LXI     H,VB3XY
   with:        MVI     A,1
                CALL    0F024H  ;GET ROW/COL DATA ADDRESS
```

6.135 **Output to the VB1 (VDTTY)**

Register A = 04
All registers except A & F preserved.
Registers H & L must be preset to the starting location of three bytes of scratch that can be use by the VB1 to save its cursor address and cursor byte. The data to be displayed is placed in register C. There is already a POP H instruction at the end of the monitor's VB1 routine, so you <u>MUST</u> PUSH H prior jumping to the routine.

TYPICAL ROUTINE:
Load Reg-C with character and CALL VB1OUT.

```
VB1OUT: PUSH    H         ;NEEDED TO COMPLETE ROUTINE
        LXI     H,VB1SCR;POINT TO SCRATCH RAM
        MVI     A,04
        JMP     0F024H    ;VECTOR TO VB1
VB1SCR: DB      0
        DB      0
```

6.14 Summary of Entry Points

NAME	FUNCTION	ADDRESS
BEGIN	Initialize Monitor	F000
CI	Console Input	F003
RI	Reader Input	F006
CO	Console Output	F009
PR	Punch Output	F00C
LO	List Output	F00F
CSTS	Console Status	F012
IOCHK	Read I/O Byte	F015
IOSET	Set I/O Byte	F018
MEMCK	Find Top Of RAM	F01B
STRNG	Output String	F01E
REENT	Re-enter Monitor	F021
JVTR	Special Vector	F024

7.0 SYSTEM CONFIGURATION PACKAGE

The system configuration package is used to tailor the monitor to a particular system hardware/software configuration. The package consists of a table establishing the I/O configuration, several entry points for establishing certain memory addresses, and driver routines for system I/O devices. The table and entry points, their addresses relative to the beginning of the package (SCP), and their functions are listed below.

7.1 IOTAB
Address = SCP + 0

This table consists of six sub-tables, each consisting of four entries of two bytes each. Each sub-table entry is the entry point address of a driver routine for a particular physical device. Each sub-table lists the driver routines for the four physical devices that may be assigned to a logical device. Sub-tables are included for console status, console input, console output, reader input, punch output, and list output.

7.2 ADSCS
Address = SCP + 30H

This is a subroutine to set registers D & E to the address of the top of the monitor scratchpad area of RAM. This subroutine is not called in the usual way. Registers D & E must be set to the return address from this routine and then jump to SCP + 30H.

7.3 ADSCR
Address = SCP + 33H

This is a subroutine to set registers H & L to the address of the top of the monitor scratchpad area of RAM. This subroutine is called in the normal way by a CALL instruction. The monitor requires 96 bytes of RAM for scratchpad.

7.4 ADIOB
Address = SCP + 36H

This is a subroutine to set registers H & L to the address of IOBYT, the byte of RAM where the current I/O configuration is recorded.

7.5 ADUST
Address = SCP + 39H

This is a subroutine to set registers H & L to the address of the users stack. This is a default value (0100 Hex) used only until replaced by the operator (XS command) or by execution of a breakpoint. It is expected that the user routines will initialize the stack pointer when they are first entered.

7.6 JVTR

Address = SCP + 3DH

This is an entry point to several subroutines. The routine is selected by the value in the A register (see Section 6.13).

8.0 I/O ASSIGNMENTS

Using the 'A' command, the following I/O selections are available in the SSM Z-80 Monitor.

AC=0	Console I/O using ports 0 & 1
AC=1	Console I/O using ports 2 & 3
AC=2	Keyboard input on ports 6 & 7, VB1 at E000
AC=3	Keyboard input on ports E0 & E1, VB3 at C000
AR=0	Reader input using ports 0 & 1
AR=1	Reader input using ports 2 & 3
*AR=2	Reader Device on ports 0E & 0F
AR=3	Reader input using ports E0 & E1
AP=0	Punch output using ports 0 & 1
AP=1	Punch output using ports 2 & 3
AP=2	Punch output using ports 0E & 0F
AP=3	Punch output to VB3 at C000
AL=0	Listing device using ports 2 & 3
AL=1	Listing device using ports 0 & 1
AL=2	Listing output to VB3 at C000
AL=3	Listing output to VB1 at E000

* With the indicated selection, the RI (Reader) input routine has a built in timer. If no data is read in approximately 1/2 second, the routine will return with the carry flag set true.

All input boards should use the even port address for STATUS, and the odd address for DATA. The STATUS bit D0 (LSB) will be used for a Data-available flag and will be negative true for inputs.

All output boards should use the even port address for STATUS, and the odd address for DATA. The STATUS bit D7 (MSB) will be used for a Data-acknowledge flag and will be negative true for outputs.

9.0 ERROR INDICATOR

The SSM Z-80 Monitor checks the command line as it is being typed in for errors. When using the monitor's commands, the statement line is preceded by a decimal point (a period) which is sent by the monitor when it is ready for another command. The first character after the decimal point must be a letter (command character) or an error symbol (an asterisk) will be printed. The next character must be a letter or hex value as specified in Section 4.0 or an error symbol will be printed and the command is aborted.

Legal Command Characters:
A, D, F, G, H, I, M, O, S, X

Legal Hex Characters:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Command Line Seperators:
Space and Comma

Command Line Terminators:
Carriage-Return

Error Symbol:
Asterisk (*)

10.0 MONITOR EXPANSION

The SSM Z-80 Monitor has been set-up to allow the user to add commands. By making a duplicate copy of the Monitor, six bytes in the look-up table (TBL) can be preset by the user with two new letter commands and their associated jump addresses to the user defined subroutines. The look-up table (TBL) in the source listing uses three bytes to define a command. The format is:

```
1 byte  DB      'L'     ;user selected ASCII symbol for the
                        ;command.
2 byte  DW      SUBR    ;address of where the subroutine
                        ;starts.
```

The Z-80 Monitor doesn't occupy the top 2K of memory for the Z-80, so there is room for additional routines in a second 2716 EPROM starting at F800 Hex.

11.0 MULTIPLE VB3 APPLICATION

The VB3 routines within the Z-80 Monitor have been broken down into modules to allow for multiple VB3s within one system with different I/O port addresses and different initialization data. Multiple VB3s would allow the Z-80 computer to have (1) more than one user console using the VB3s for output, (2) VB3s as system status indicators for different functions while one VB3 is used as the main console, or (3) three VB3 boards driving the three inputs of a RGB color TV monitor.

There are two routines which can be used to support the VB3 in a Z-80 based system. One routine is labeled VB3IZ (used to initialize a VB3) and the other labeled is VB3OUT (used to output a character to the VB3). Both of these routines can be entered by setting the A-register and CALLing location F024 (see section 6.13).

On power-up of the Z-80 system you should initialize the VB3s within the system by passing a six byte table with VB3IZ to each VB3 board.

Example of a three VB3 initialization routine:

```
;Set up I/O ports for each VB3.
;VB3-1
KSTAT1       EQU     0E0H    ;TURN-ON VB3
KDATA1       EQU     KSTAT1+1;TURN-OFF VB3
;VB3-2
KSTAT2       EQU     0E2H
KDATA2       EQU     KSTAT2+1
;VB3-3
KSTAT3       EQU     0E4H
KDATA3       EQU     KSTAT3+1

;Main entry is here!
VINTZ:       OUT     KSTAT1  ;TURN-ON
             LXI     H,VTBL1 ;POINT TO DATA TBL
             MVI     A,2
             CALL    0F024H  ;GO TO INITIALIZE
             OUT     KDATA1  ;TURN-OFF
             ;
             OUT     KSTAT2
             LXI     H,VTBL2
             MVI     A,2
             CALL    0F024H
             OUT     KDATA2
             ;
             OUT     KSTAT3
             LXI     H,VTBL3
             MVI     A,2
             CALL    0F024H
             OUT     KDATA3
             ;
             RET
```

```
;Data to initialize each VB3.
;VB3-1 (80X16)
VTBL1:          DB      70H
                DB      65H
                DB      5DH
                DB      0FH
                DB      03H
                DB      26H
                DB      0FH

;VB3-2 (80X24)
VTBL2:          DB      70H
                DB      0BCH
                DB      6DH
                DB      17H
                DB      06H
                DB      29H
                DB      17H

;VB3-3 (80X32)
VTBL3:          DB      70H
                DB      0D3H
                DB      5DH
                DB      1FH
                DB      06H
                DB      1EH
                DB      1FH

                END
```

To output to one of many VB3s, use the routine listed in section 6.134.

12.0 VB3 FORMAT CHANGE APPLICATION

If a VB3 and parallel keyboard are being utilized with the Z-80 Monitor, after power-up of the monitor the number of characters across and lines down the VB3 display can be altered. This simple 32 byte routine can be entered and executed at 0100 Hex to set the VB3 into a new mode of operation. As long as all further user routines re-enter the monitor at F021 Hex, instead of F000 Hex, the new VB3 format will be maintained.

```
F024 =          VB3IZ   EQU     0F024H
F024 =          RWCL    EQU     0F024H
F021 =          MONT    EQU     0F021H

                ;START VB3 FORMAT ROUTINE

0100                    ORG     0100H

                ;CHANGE VB3 FORMAT

0100 D3E0               OUT     0E0H            ;TURN-ON VB3
0102 3E02               MVI     A,2             ;SET VECTOR NUMBER
0104 211701             LXI     H,NEW           ;NEW TABLE OF VALUES
0107 CD24F0             CALL    VB3IZ           ;SEND TO VB3

                ;CHANGE MONITOR COLUMN/ROW NUMBERS

010A 3E01               MVI     A,1             ;SET VECTOR NUMBER
010C CD24F0             CALL    RWCL            ;FIND VB3 DATA ADDRESS
010F 23                 INX     H
0110 3620               MVI     M,32            ;SET ROW NUMBER
0112 D3E1               OUT     0E1H            ;TURN-OFF VB3

                ;RETURN TO MONITOR

0114 C321F0             JMP     MONT

                ;TABLE OF DATA

0117 70         NEW:    DB      70H             ;HORIZONTAL WIDTH
0118 D3                 DB      0D3H            ;INTERLACE MODE
0119 5D                 DB      5DH             ;LINES/CHARARACTER & 80 CHARS
011A 1F                 DB      1FH             ;NUMBER OF ROWS-1
011B 06                 DB      06H             ;RASTER FILL-IN
011C 1E                 DB      1EH             ;TOP MARGIN
011D 1F                 DB      1FH             ;SET BOTTOM LINE
```

13.0 SOURCE LISTING

```
                         0001 ;*******SSM Z80 MONITOR V1.10********

                         0002 ;            PROGRAMMER:
                         0003 ;            C. E. OHME,1977

                         0004 ;            TECHNICAL SUPPORT:
                         0005 ;            P. DENNIS,4-21-80

                         0006 ;            MODIFIED BY:
                         0007 ;            M. T. WRIGHT,4-27-80
                         0008 ;            D. M. FISCHLER, 9-15-80

                         0009 ;            COPYRIGHT 1980 BY SSM MICROCOMPUTER PRODUCTS
                         0010 ;            ALL RIGHTS RESERVED

      (FFFF)             0011 TRUE     EQU      0FFFFH
      (0000)             0012 FALSE    EQU      NOT TRUE

                         0013 ;            STARTING ADDRESS OF MONITOR
      (F000)             0014 LOC      EQU      0F000H

                         0015 ;            SIZE OF BREAKPOINT STACK
                         0016 ;            WARNING:
                         0017 ;            DON'T CHANGE THESE VALUES UNLESS YOU
                         0018 ;            REALLY UNDERSTAND THE MONITOR.
      (0016)             0019 BSTK     EQU      22                ;22 BYTES DEEP

                         0020 ;            ASCII CHARACTERS
      (000D)             0021 CR       EQU      0DH
      (000A)             0022 LF       EQU      0AH
      (0008)             0023 BS       EQU      08H
      (000C)             0024 FF       EQU      0CH

0000'                    0025          ORG      LOC

                         0026 ;            EXTERNALLY REFERENCED SUBROUTINE
                         0027 ;            JUMP TABLE
F000  C32AF0             0028          JP       BEGIN
F003  C304F1             0029          JP       CI
F006  C30EF1             0030          JP       RI
F009  C3FFF0             0031          JP       CO
F00C  C313F1             0032          JP       PR
F00F  C318F1             0033          JP       LO
F012  C309F1             0034          JP       CSTS
F015  C33BF1             0035          JP       IOCHK
F018  C342F1             0036          JP       IOSET
F01B  C3DAF1             0037          JP       MEMCK
F01E  C34BF1             0038          JP       STRNG
F021  C327F0             0039          JP       REENT
F024  C35EF5             0040          JP       JVTR

F027  0E00               0041 REENT:   LD       C,0
F029  21                 0042          DEFB     21H               ;"LXI H" TRICK
```

```
F02A  0E01      0043 BEGIN:  LD     C,1
F02C  1132F0    0044         LD     DE,BG0
F02F  C351F5    0045         JP     ADSCS
F032  79        0046 BG0:    LD     A,C            ;SAVE COLD BOOT FLAG
F033  21FDFF    0047         LD     HL,VX3-ENDX
F036  19        0048         ADD    HL,DE          ;POINT TO (VX3-1)
F037  012600    0049         LD     BC,VX3-EXIT    ;NUMBER OF BYTES
F03A  11BBF2    0050         LD     DE,VX3-1       ;TOP 3 OF TEMPLATE
F03D  EB        0051 BG1:    EX     DE,HL
F03E  EDB8      0052         LDDR                  ;MOVE TEMPLATE INTO RAM
F040  EB        0053         EX     DE,HL
F041  23        0054         INC    HL
F042  4F        0055         LD     C,A            ;RESTORE COLD BOOT FLAG
F043  F9        0056         LD     SP,HL
F044  CD5AF5    0057         CALL   ADUST
F047  E5        0058         PUSH   HL
F048  210000    0059         LD     HL,0
                0060 ; *******************************
F04B  060A      0061         LD     B,(BSTK-2)/2
                0062 ; *******************************
F04D  E5        0063 BG1A:   PUSH   HL             ;CLEAR BREAK PT. STACK
F04E  05        0064         DEC    B
F04F  20FC      0065         JR     NZ,BG1A
F051  79        0066         LD     A,C
F052  B7        0067         OR     A
F053  2826      0068         JR     Z,BG4

                0069 ;SET-UP I/O BYTE FOR MONITOR
F055  79        0070 BG2:    LD     A,C            ;GET CONSOLE VALUE
F056  3D        0071         DEC    A              ;TRY NEXT CONSOLE
F057  E603      0072         AND    3              ;ONLY FOUR!
F059  4F        0073         LD     C,A
F05A  CD42F1    0074         CALL   IOSET          ;SET FOR 0 THRU 3
F05D  CD09F1    0075         CALL   CSTS           ;TEST FOR INPUT
F060  28F3      0076         JR     Z,BG2          ;NO STATUS?KEEP LOOKING
F062  CD04F1    0077         CALL   CI             ;GET 1ST CHAR.
F065  FE0D      0078         CP     CR             ;MUST BE A 'CR'.
F067  20EC      0079         JR     NZ,BG2         ;IF ERR, KEEP TRYING.
F069  CD36F6    0080         CALL   VB3MZ          ;INITIALIZE VB3
F06C  CD54F5    0081         CALL   ADSCR          ;FIND TOP OF RAM
F06F  2B        0082         DEC    HL             ;SKIP I/O BYTE
F070  115012    0083         LD     DE,NROWS*256+NCOLS
F073  72        0084         LD     (HL),D         ;SET NO. OF ROWS
F074  2B        0085         DEC    HL
F075  73        0086         LD     (HL),E         ;SET NO. OF COLUMNS
F076  0E0C      0087         LD     C,FF           ;CLEAR SCREEN
F078  CDFFF0    0088         CALL   CO             ;
F07B  21BFF0    0089 BG4:    LD     HL,VERS
F07E  CD4BF1    0090         CALL   STRNG
F081  B7        0091         OR     A

                0092 ;       COMMAND RETURN POINT
F082  D293F0    0093 CMNDR:  JP     NC,START
```

```
                    0094 ;          ERROR RETURN
F085  CD54F5        0095 LER:       CALL     ADSCR
                    0096 ; ******************************
F088  11C2FF        0097            LD       DE,EXIT-ENDX+1-BSTK
                    0098 ;******************************
F08B  19            0099            ADD      HL,DE
F08C  F9            0100            LD       SP,HL
F08D  21D6F0        0101            LD       HL,ERM
F090  CD4BF1        0102            CALL     STRNG

                    0103 ;          INPUT AND EXECUTE NEXT COMMAND
F093  FB            0104 START:     EI
F094  CD7CF1        0105            CALL     CRLF
F097  0E2E          0106            LD       C,'.'
F099  CDFFF0        0107            CALL     CO
F09C  CD5CF1        0108            CALL     TI
F09F  B7            0109            OR       A                ;IS IT A BREAK?
F0A0  CA85F0        0110            JP       Z,LER
F0A3  47            0111            LD       B,A              ;SAVE CMD LETTER
F0A4  2182F0        0112            LD       HL,CMNDR
F0A7  E5            0113            PUSH     HL
F0A8  21D8F0        0114            LD       HL,TBL
F0AB  7E            0115 S1:        LD       A,(HL)           ;GET POSSIBLE CMD.
F0AC  23            0116            INC      HL
F0AD  B8            0117            CP       B                ;IS THERE A MATCH?
F0AE  2808          0118            JR       Z,S2             ;JUMP IF YES.
F0B0  B7            0119            OR       A                ;END OF CMD TABLE?
F0B1  CA85F0        0120            JP       Z,LER
F0B4  23            0121            INC      HL
F0B5  23            0122            INC      HL
F0B6  18F3          0123            JR       S1
F0B8  7E            0124 S2:        LD       A,(HL)
F0B9  23            0125            INC      HL
F0BA  66            0126            LD       H,(HL)
F0BB  6F            0127            LD       L,A
F0BC  0E02          0128            LD       C,2
F0BE  E9            0129            JP       (HL)
F0BF  0D0A5353      0130 VERS:      DEFB     CR,LF,'SSM Z80 MONITOR V1.1'
      4D205A38
      30204D4F
      4E49544F
      52205631
      2E31
F0D5  B0            0131            DEFB     '0' OR 80H
F0D6  0AAA          0132 ERM:       DEFB     LF,'*' OR 80H

                    0133 ;          COMMAND JUMP TABLE
F0D8  41            0134 TBL:       DEFB     'A'
F0D9  BFF2          0135            DEFW     ASSIGN
F0DB  44            0136            DEFB     'D'
F0DC  02F3          0137            DEFW     DISP
F0DE  46            0138            DEFB     'F'
F0DF  31F3          0139            DEFW     FILL
F0E1  47            0140            DEFB     'G'
F0E2  40F3          0141            DEFW     GOTO
```

```
F0E4  48         0142          DEFB     'H'
F0E5  8EF3       0143          DEFW     HEXN
F0E7  49         0144          DEFB     'I'
F0E8  AAF3       0145          DEFW     ZINP               ;Z80 INPUT CMD
F0EA  4D         0146          DEFB     'M'
F0EB  B9F3       0147          DEFW     MOVE
F0ED  4F         0148          DEFB     'O'
F0EE  CBF3       0149          DEFW     ZOUT               ;Z80 OUTPUT CMD
F0F0  53         0150          DEFB     'S'
F0F1  D4F3       0151          DEFW     SUBS
F0F3  58         0152          DEFB     'X'
F0F4  14F4       0153          DEFW     X
F0F6  00         0154          DEFB     0                  ;FUTURE CMD.
F0F7  0000       0155          DEFW     0                  ;FUTURE ROUTINE ADDR.
F0F9  00         0156          DEFB     0                  ;DITTO
F0FA  0000       0157          DEFW     0
F0FC  00         0158          DEFB     0                  ;END OF TABLE

                 0159 ;        UTILITY  SUBROUTINES
F0FD  0E20       0160 BLK:     LD       C,' '
F0FF  CD1DF1     0161 CO:      CALL     IOBR
F102  0110       0162          DEFB     1,10H
F104  CD1DF1     0163 CI:      CALL     IOBR
F107  0108       0164          DEFB     1,8
F109  CD1DF1     0165 CSTS:    CALL     IOBR
F10C  0100       0166          DEFB     1,0
F10E  CD1DF1     0167 RI:      CALL     IOBR
F111  0418       0168          DEFB     4,18H
F113  CD1DF1     0169 PR:      CALL     IOBR
F116  0320       0170          DEFB     3,20H
F118  CD1DF1     0171 LO:      CALL     IOBR
F11B  0228       0172          DEFB     2,28H
F11D  E3         0173 IOBR:    EX       (SP),HL
F11E  C5         0174          PUSH     BC
F11F  46         0175          LD       B,(HL)
F120  23         0176          INC      HL
F121  4E         0177          LD       C,(HL)
F122  CD57F5     0178          CALL     ADIOB
F125  7E         0179          LD       A,(HL)
F126  0F         0180          RRCA
F127  07         0181 IOB1:    RLCA
F128  07         0182          RLCA
F129  05         0183          DEC      B
F12A  20FB       0184          JR       NZ,IOB1
F12C  E606       0185          AND      6
F12E  81         0186          ADD      C
F12F  4F         0187          LD       C,A
F130  2121F5     0188          LD       HL,IOTAB
F133  09         0189          ADD      HL,BC
F134  7E         0190          LD       A,(HL)
F135  23         0191          INC      HL
F136  66         0192          LD       H,(HL)
F137  6F         0193          LD       L,A
F138  C1         0194          POP      BC
F139  E3         0195          EX       (SP),HL
```

```
F13A  C9                       RET

F13B  E5              IOCHK:   PUSH    HL
F13C  CD57F5                   CALL    ADIOB
F13F  7E                       LD      A,(HL)
F140  E1                       POP     HL
F141  C9                       RET

F142  E5              IOSET:   PUSH    HL
F143  F5                       PUSH    AF
F144  CD57F5                   CALL    ADIOB
F147  71                       LD      (HL),C
F148  F1                       POP     AF
F149  E1                       POP     HL
F14A  C9                       RET

F14B  7E              STRNG:   LD      A,(HL)
F14C  E67F                     AND     7FH
F14E  C8                       RET     Z
F14F  4F                       LD      C,A
F150  7E                       LD      A,(HL)
F151  B7                       OR      A
F152  FAFFF0                   JP      M,CO
F155  CDFFF0                   CALL    CO
F158  23                       INC     HL
F159  C34BF1                   JP      STRNG

F15C  CD04F1          TI:      CALL    CI
F15F  E67F                     AND     7FH
F161  C5                       PUSH    BC
F162  4F                       LD      C,A
F163  CDFFF0                   CALL    CO
F166  79                       LD      A,C
F167  C1                       POP     BC
F168  FE61                     CP      61H                 ;LOWER CASE -A-
F16A  3F                       CCF
F16B  D0                       RET     NC                  ;RET IF LESS THAN
F16C  FE7B                     CP      7AH+1               ;LOWER CASE -Z-
F16E  D0                       RET     NC                  ;RET IF GREATER THAN
F16F  E65F                     AND     5FH                 ;CONVERT TO UPPER CASE
F171  C9                       RET

F172  E60F            CONV:    AND     0FH
F174  C690                     ADD     90H
F176  27                       DAA
F177  CE40                     ADC     40H
F179  27                       DAA
F17A  4F                       LD      C,A
F17B  C9                       RET

F17C  0E0D            CRLF:    LD      C,CR
F17E  CDFFF0                   CALL    CO
F181  0E0A                     LD      C,LF
F183  C3FFF0                   JP      CO

```

```
F186  0E01       0251 EXPR1:  LD     C,1
F188  210000     0252 EXPR:   LD     HL,0000H
F18B  CD5CF1     0253 EX0:    CALL   TI
F18E  47         0254 EX1:    LD     B,A              ;SAVE CHARACTER
F18F  CDF8F1     0255         CALL   NIBBL            ;CREATE NYBBLE
F192  DA9EF1     0256         JP     C,EX2            ;JUMP IF ERROR
F195  29         0257         ADD    HL,HL            ;SHIFT HL LEFT 4
F196  29         0258         ADD    HL,HL
F197  29         0259         ADD    HL,HL
F198  29         0260         ADD    HL,HL
F199  B5         0261         OR     L                ;COMBINE WITH NYBBLE
F19A  6F         0262         LD     L,A
F19B  C38BF1     0263         JP     EX0              ;GO BACK FOR MORE.
F19E  E3         0264 EX2:    EX     (SP),HL
F19F  E5         0265         PUSH   HL
F1A0  78         0266         LD     A,B
F1A1  CD0DF2     0267         CALL   P2C
F1A4  D2ACF1     0268         JP     NC,EX3
F1A7  0D         0269         DEC    C
F1A8  C285F0     0270         JP     NZ,LER
F1AB  C9         0271         RET
F1AC  C285F0     0272 EX3:    JP     NZ,LER
F1AF  0D         0273         DEC    C
F1B0  C288F1     0274         JP     NZ,EXPR
F1B3  C9         0275         RET
F1B4  0E01       0276 EXF:    LD     C,1
F1B6  210000     0277         LD     HL,0000H
F1B9  C38EF1     0278         JP     EX1
                 0279
F1BC  23         0280 HILO:   INC    HL
F1BD  7C         0281         LD     A,H
F1BE  B5         0282         OR     L
F1BF  37         0283         SCF
F1C0  C8         0284         RET    Z
F1C1  7B         0285         LD     A,E
F1C2  95         0286         SUB    L
F1C3  7A         0287         LD     A,D
F1C4  9C         0288         SBC    H
F1C5  C9         0289         RET
                 0290
F1C6  7C         0291 LADR:   LD     A,H
F1C7  CDCBF1     0292         CALL   LBYTE
F1CA  7D         0293         LD     A,L
F1CB  F5         0294 LBYTE:  PUSH   AF
F1CC  0F         0295         RRCA
F1CD  0F         0296         RRCA
F1CE  0F         0297         RRCA
F1CF  0F         0298         RRCA
F1D0  CDD4F1     0299         CALL   HXD
F1D3  F1         0300         POP    AF
F1D4  CD72F1     0301 HXD:    CALL   CONV
F1D7  C3FFF0     0302         JP     CO
F1DA  E5         0303 MEMCK:  PUSH   HL
F1DB  D5         0304         PUSH   DE
F1DC  CD54F5     0305         CALL   ADSCR
```

```
F1DF  EB                     EX      DE,HL
F1E0  210000                 LD      HL,0
F1E3  24           MEMO:     INC     H
F1E4  7E                     LD      A,(HL)
F1E5  2F                     CPL
F1E6  77                     LD      (HL),A
F1E7  BE                     CP      (HL)
F1E8  2F                     CPL
F1E9  77                     LD      (HL),A
F1EA  28F7                   JR      Z,MEMO
F1EC  2B                     DEC     HL
F1ED  44                     LD      B,H
F1EE  7C                     LD      A,H
F1EF  BA                     CP      D
F1F0  3EAD                   LD      A,100H-BSTK-20-(ENDX-EXIT)

                             ;BSTK=BREAKPT. STACK
                             ;ENDX-EXIT=TEMPL. SIZE
                             ;20=SOME MONIT. RAM
F1F2  D1                     POP     DE
F1F3  E1                     POP     HL
F1F4  C8                     RET     Z
F1F5  3EFF                   LD      A,0FFH
F1F7  C9                     RET
F1F8  D630         NIBBL:    SUB     '0'
F1FA  D8                     RET     C
F1FB  C6E9                   ADD     '0'-'G' AND 0FFH
F1FD  D8                     RET     C
F1FE  C606                   ADD     6
F200  F206F2                 JP      P,NI0
F203  C607                   ADD     7
F205  D8                     RET     C
F206  C60A         NI0:      ADD     10
F208  B7                     OR      A
F209  C9                     RET
F20A  CD5CF1       PCHK:     CALL    TI
F20D  FE20         P2C:      CP      ' '
F20F  C8                     RET     Z
F210  FE2C                   CP      ','
F212  C8                     RET     Z
F213  FE0D                   CP      CR
F215  37                     SCF
F216  C8                     RET     Z
F217  3F                     CCF
F218  C9                     RET

                   ;         BREAKPOINT ENTRY POINT
F219  E5           RESTRT:   PUSH    HL                 ;PUT REGISTERS ON TO
F21A  D5                     PUSH    DE                 ;   USER'S STACK
F21B  C5                     PUSH    BC
F21C  F5                     PUSH    AF
F21D  CD54F5                 CALL    ADSCR
F220  11D8FF                 LD      DE,EXIT-ENDX+1
F223  19                     ADD     HL,DE              ;DELTA EQUIV. TO 'EXIT'
F224  EB                     EX      DE,HL
```

```
F225  210A00          LD     HL,10
F228  39              ADD    HL,SP              ;SET TO TOP OF USER STX
F229  0604            LD     B,4                ;
F22B  EB              EX     DE,HL              ;DE=USER STX,HL='EXIT'

F22C  2B      RST0:   DEC    HL
F22D  72              LD     (HL),D             ;SAVE DATA IN MONT.RAM
F22E  2B              DEC    HL
F22F  73              LD     (HL),E             ;SAVE USER STX ADDR.
F230  D1              POP    DE                 ;  THEN 'PSW',NEXT'BC'
F231  05              DEC    B                  ;  LAST 'DE'
F232  20F8            JR     NZ,RST0
F234  C1              POP    BC                 ;DE='HL',BC=RET.ADDR.
F235  0B              DEC    BC                 ;MOVE RET.ADDR. BACK
F236  F9              LD     SP,HL              ;START NEW STACK

F237  08              EX     AF,AF'
F238  D9              EXX
F239  E5              PUSH   HL                 ;PUSH HL
F23A  D5              PUSH   DE                 ;PUSH DE
F23B  C5              PUSH   BC                 ;PUSH BC
F23C  F5              PUSH   AF                 ;PUSH AF
F23D  DDE5            PUSH   IX
F23F  FDE5            PUSH   IY
F241  ED57            LD     A,I
F243  47              LD     B,A                ;LD B,A
F244  ED5F            LD     A,R
F246  4F              LD     C,A                ;LD C,A
F247  C5              PUSH   BC                 ;PUSH BC
F248  D9              EXX
F249  213300          LD     HL,TLOC
F24C  39              ADD    HL,SP
F24D  7E              LD     A,(HL)             ;DOES TRAP #1 ADDRESS
F24E  91              SUB    C                  ;  EQUAL RET. ADDRESS
F24F  23              INC    HL
F250  2004            JR     NZ,RST1            ;NO?
F252  7E              LD     A,(HL)
F253  90              SUB    B
F254  280C            JR     Z,RST3             ;YES?
F256  23      RST1:   INC    HL
F257  23              INC    HL                 ;SKIP TRAP#1 INSTRUC.
F258  7E              LD     A,(HL)             ;DOES TRAP#2 ADDRESS
F259  91              SUB    C                  ;  EQUAL RET. ADDRESS
F25A  2005            JR     NZ,RST2            ;NO?
F25C  23              INC    HL
F25D  7E              LD     A,(HL)
F25E  90              SUB    B
F25F  2801            JR     Z,RST3             ;YES?
F261  03      RST2:   INC    BC                 ;MUST BE FALSE BREAK PT
F262  212D00  RST3:   LD     HL,LLOC
F265  39              ADD    HL,SP
F266  73              LD     (HL),E             ;SAVE HL IN TEMPLATE
F267  23              INC    HL                 ;  IN 'LXI'INSTRUCTION
F268  72              LD     (HL),D
F269  213100          LD     HL,PLOC-1
```

```
F26C   39                ADD     HL,SP
F26D   71                LD      (HL),C        ;SAVE RET.ADDRESS IN
F26E   23                INC     HL            ;  TEMPLATE 'JMP'INSTR.
F26F   70                LD      (HL),B
F270   C5                PUSH    BC            ;SAVE BREAK ADDR.
F271   21D6F0            LD      HL,ERM        ;
F274   CD4BF1            CALL    STRNG         ;PRINT '*'
F277   E1                POP     HL            ;GET BREAK ADDR.
F278   CDC6F1            CALL    LADR          ;PRINT ADDRESS

                ;RESTORE BREAK PT. BYTES IN PROGRAM
F27B   213300            LD      HL,TLOC
F27E   39                ADD     HL,SP         ;PT. TO TRAP ADDR. 1
F27F   1602              LD      D,2
F281   4E       RST4:    LD      C,(HL)        ;GET LOW HALF
F282   3600              LD      (HL),0
F284   23                INC     HL
F285   46                LD      B,(HL)        ;GET HIGH HALF
F286   3600              LD      (HL),0
F288   23                INC     HL
F289   79                LD      A,C
F28A   B0                OR      B             ;TRAP ADDR.=0?
F28B   2802              JR      Z,RST5        ;SKIP TRAP IF YES.
F28D   7E                LD      A,(HL)        ;GET OLD INSTR. BYTE
F28E   02                LD      (BC),A        ;RESTORE PROGRAM BYTE
F28F   23       RST5:    INC     HL            ;NEXT TRAP
F290   15                DEC     D
F291   20EE              JR      NZ,RST4
F293   C393F0            JP      START

                ;        SCRATCHPAD TEMPLATE
F296   C1       EXIT:    POP     BC
F297   79                LD      A,C
F298   ED4F              LD      R,A
F29A   78                LD      A,B
F29B   ED47              LD      I,A
F29D   FDE1              POP     IY
F29F   DDE1              POP     IX
F2A1   F1                POP     AF
F2A2   C1                POP     BC
F2A3   D1                POP     DE
F2A4   E1                POP     HL
F2A5   D9                EXX
F2A6   08                EX      AF,AF'
F2A7   D1                POP     DE
F2A8   C1                POP     BC
F2A9   F1                POP     AF
F2AA   E1                POP     HL
F2AB   F9                LD      SP,HL
F2AC   210000            LD      HL,0
       (F2AD)   HLX      EQU     $-2
F2AF   FB                EI
F2B0   C30000            JP      0
       (F2B1)   PCX      EQU     $-2
F2B3   0000     T1A:     DEFW    0             ;TRAP 1 ADDR
```

```
F2B5  00          0465        DEFB    0                        ;TRAP 1 INST
F2B6  0000        0466        DEFW    0                        ;TRAP 2 ADDR
F2B8  00          0467        DEFB    0                        ;TRAP 2 INST
F2B9  0000        0468 VX1:   DEFW    0                        ;VIDEO POINTER
F2BB  00          0469        DEFB    0                        ;VIDEO HOLD
F2BC  50          0470 VX3:   DEFB    NCOLS                    ;VB3 COLUMN CONSTANT
F2BD  12          0471        DEFB    NROWS                    ;VB3 ROW CONSTANT
F2BE  00          0472        DEFB    0                        ;IOBYT
                  0473 ENDX:
      (0013)      0474 ALOC   EQU     13H
      (0011)      0475 BLOC   EQU     11H
      (0010)      0476 CLOC   EQU     10H
      (000F)      0477 DLOC   EQU     0FH
      (000E)      0478 ELOC   EQU     0EH
      (0012)      0479 FLOC   EQU     12H
      (002E)      0480 HLOC   EQU     HLX-EXIT+BSTK+1
      (002D)      0481 LLOC   EQU     HLX-EXIT+BSTK
      (0032)      0482 PLOC   EQU     PCX-EXIT+BSTK+1
      (0015)      0483 SLOC   EQU     15H
      (0033)      0484 TLOC   EQU     T1A-EXIT+BSTK
                  0485 ;
      (0007)      0486 APLOC  EQU     07H
      (0009)      0487 BPLOC  EQU     09H
      (0008)      0488 CPLOC  EQU     08H
      (000B)      0489 DPLOC  EQU     0BH
      (000A)      0490 EPLOC  EQU     0AH
      (0006)      0491 FPLOC  EQU     06H
      (000D)      0492 HPLOC  EQU     0DH
      (000C)      0493 LPLOC  EQU     0CH
      (0005)      0494 XLOC   EQU     05H
      (0003)      0495 YLOC   EQU     03H
      (0001)      0496 ILOC   EQU     01H
      (0000)      0497 RLOC   EQU     00H

                  0498 ;      COMMAND IMPLEMENTATION

                  0499 ;      ASSIGN COMMAND
F2BF  CD5CF1      0500 ASSIGN: CALL   TI
F2C2  0600        0501        LD      B,0
F2C4  FE43        0502        CP      'C'
F2C6  280F        0503        JR      Z,AS1
F2C8  04          0504        INC     B
F2C9  FE52        0505        CP      'R'
F2CB  280A        0506        JR      Z,AS1
F2CD  04          0507        INC     B
F2CE  FE50        0508        CP      'P'
F2D0  2805        0509        JR      Z,AS1
F2D2  04          0510        INC     B
F2D3  FE4C        0511        CP      'L'
F2D5  2029        0512        JR      NZ,EREXT
F2D7  CD5CF1      0513 AS1:   CALL    TI
F2DA  FE3D        0514        CP      '='
F2DC  20F9        0515        JR      NZ,AS1
F2DE  CD5CF1      0516        CALL    TI
F2E1  D630        0517        SUB     '0'
```

```
F2E3   6F          0518           LD       L,A
F2E4   FA00F3      0519           JP       M,EREXT
F2E7   FE04        0520           CP       4
F2E9   F200F3      0521           JP       P,EREXT
F2EC   2603        0522           LD       H,3
F2EE   05          0523 AS2:      DEC      B
F2EF   FAF6F2      0524           JP       M,AS3
F2F2   29          0525           ADD      HL,HL
F2F3   29          0526           ADD      HL,HL
F2F4   18F8        0527           JR       AS2
F2F6   EB          0528 AS3:      EX       DE,HL
F2F7   CD57F5      0529           CALL     ADIOB
F2FA   7E          0530           LD       A,(HL)
F2FB   B2          0531           OR       D
F2FC   AA          0532           XOR      D
F2FD   B3          0533           OR       E
F2FE   77          0534           LD       (HL),A
F2FF   C9          0535           RET
F300   37          0536 EREXT:    SCF
F301   C9          0537           RET

                   0538 ;          DISPLAY  COMMAND
F302   CD88F1      0539 DISP:     CALL     EXPR
F305   D1          0540           POP      DE
F306   E1          0541           POP      HL
F307   CD7CF1      0542 DI0:      CALL     CRLF
F30A   CDC6F1      0543           CALL     LADR
F30D   CDFDF0      0544 DI1:      CALL     BLK
F310   7E          0545           LD       A,(HL)
F311   CDCBF1      0546           CALL     LBYTE
F314   CDBCF1      0547           CALL     HILO
F317   3F          0548           CCF
F318   D0          0549           RET      NC
F319   7D          0550           LD       A,L
F31A   E60F        0551           AND      0FH
F31C   20EF        0552           JR       NZ,DI1
F31E   CD09F1      0553           CALL     CSTS                ;TEST KEYBOARD STATUS
F321   28E4        0554           JR       Z,DI0               ;JUMP IF NO DATA
F323   CD04F1      0555           CALL     CI                  ;GET CHARACTER
F326   FE53        0556           CP       'S'                 ;
F328   C8          0557           RET      Z                   ;RET, IF COMMANDED
F329   CD04F1      0558           CALL     CI                  ;WAIT.  GET CHARACTER
F32C   FE53        0559           CP       'S'
F32E   20D7        0560           JR       NZ,DI0              ;CONTINUE IF NO 'S'
F330   C9          0561           RET

                   0562 ;          FILL COMMAND
F331   0C          0563 FILL:     INC      C
F332   CD88F1      0564           CALL     EXPR
F335   C1          0565           POP      BC
F336   D1          0566           POP      DE
F337   E1          0567           POP      HL
F338   71          0568 FI0:      LD       (HL),C
F339   CDBCF1      0569           CALL     HILO
F33C   30FA        0570           JR       NC,FI0
```

```
F33E  B7          0571          OR      A
F33F  C9          0572          RET

                  0573 ;        GOTO COMMAND
F340  E1          0574 GOTO:    POP     HL
F341  CD0AF2      0575          CALL    PCHK
F344  3840        0576          JR      C,GO3
F346  281B        0577          JR      Z,GO0
F348  CDB4F1      0578          CALL    EXF
F34B  D1          0579          POP     DE
F34C  213200      0580          LD      HL,PLOC
F34F  39          0581          ADD     HL,SP
F350  72          0582          LD      (HL),D
F351  2B          0583          DEC     HL
F352  73          0584          LD      (HL),E
F353  78          0585          LD      A,B
F354  FE0D        0586          CP      CR
F356  282E        0587          JR      Z,GO3
F358  3EC3        0588          LD      A,0C3H           ;C3 = JMP CODE
F35A  320800      0589          LD      (8),A
F35D  2119F2      0590          LD      HL,RESTRT
F360  220900      0591          LD      (9),HL
F363  1602        0592 GO0:     LD      D,2
F365  213300      0593          LD      HL,TLOC
F368  39          0594          ADD     HL,SP
F369  E5          0595 GO1:     PUSH    HL
F36A  CD86F1      0596          CALL    EXPR1
F36D  58          0597          LD      E,B
F36E  C1          0598          POP     BC
F36F  E1          0599          POP     HL
F370  78          0600          LD      A,B
F371  B1          0601          OR      C
F372  280A        0602          JR      Z,GO2
F374  71          0603          LD      (HL),C
F375  23          0604          INC     HL
F376  70          0605          LD      (HL),B
F377  23          0606          INC     HL
F378  0A          0607          LD      A,(BC)
F379  77          0608          LD      (HL),A
F37A  23          0609          INC     HL
F37B  3ECF        0610          LD      A,0CFH           ;CF = RST1 CODE
F37D  02          0611          LD      (BC),A
F37E  7B          0612 GO2:     LD      A,E
F37F  FE0D        0613          CP      CR
F381  2803        0614          JR      Z,GO3
F383  15          0615          DEC     D
F384  20E3        0616          JR      NZ,GO1
F386  CD7CF1      0617 GO3:     CALL    CRLF
                  0618 ; ********************
F389  211600      0619          LD      HL,BSTK
                  0620 ; ********************
F38C  39          0621          ADD     HL,SP
F38D  E9          0622          JP      (HL)
```

```
                ;          HEXADECIMAL COMMAND
F38E  CD88F1    HEXN:      CALL    EXPR
F391  D1                   POP     DE
F392  E1                   POP     HL
F393  CD7CF1               CALL    CRLF
F396  E5                   PUSH    HL
F397  19                   ADD     HL,DE
F398  CDC6F1               CALL    LADR
F39B  CDFDF0               CALL    BLK
F39E  E1                   POP     HL
F39F  7D                   LD      A,L
F3A0  93                   SUB     E
F3A1  6F                   LD      L,A
F3A2  7C                   LD      A,H
F3A3  9A                   SBC     D
F3A4  67                   LD      H,A
F3A5  CDC6F1               CALL    LADR
F3A8  B7                   OR      A
F3A9  C9                   RET

                ;          INPUT COMMAND
                ; SUGGESTED BY DAN FISCHLER,1980
F3AA  CD86F1    ZINP:      CALL    EXPR1           ;GET PORT NUMBER
F3AD  C1                   POP     BC              ;PLACE NO. IN REG.C
F3AE  ED40                 IN      B,(C)
F3B0  CDFDF0               CALL    BLK             ;PRINT BLANK
F3B3  78                   LD      A,B             ;GET DATA
F3B4  CDCBF1               CALL    LBYTE           ;PRINT DATA
F3B7  B7                   OR      A               ;CLEAR ERR FLAG
F3B8  C9                   RET
                ;          MOVE COMMAND
F3B9  0C        MOVE:      INC     C
F3BA  CD88F1               CALL    EXPR
F3BD  C1                   POP     BC
F3BE  D1                   POP     DE
F3BF  E1                   POP     HL
F3C0  7E        MV0:       LD      A,(HL)
F3C1  02                   LD      (BC),A
F3C2  03                   INC     BC
F3C3  CDBCF1               CALL    HILO
F3C6  D2C0F3               JP      NC,MV0
F3C9  B7                   OR      A
F3CA  C9                   RET

                ;          OUTPUT COMMAND
F3CB  CD88F1    ZOUT:      CALL    EXPR            ;GET PORT & DATA
F3CE  D1                   POP     DE              ;PUT DATA IN REG.E
F3CF  C1                   POP     BC              ;PUT PORT NO. IN REG.C
F3D0  ED59                 OUT     (C),E
F3D2  B7                   OR      A               ;CLEAR ERR FLAG
F3D3  C9                   RET

                ;          SUBSTITUTE COMMAND
F3D4  CD86F1    SUBS:      CALL    EXPR1
F3D7  CD0DF2               CALL    P2C
```

```
F3DA  E1                  POP    HL
F3DB  D8                  RET    C
F3DC  0608                LD     B,8               ;SUBST. EIGHT PER LINE
F3DE  C5            SU0:  PUSH   BC                ;SAVE COUNT
F3DF  7E                  LD     A,(HL)
F3E0  CDCBF1              CALL   LBYTE
F3E3  0E2D                LD     C,'-'
F3E5  CDFFF0              CALL   CO
F3E8  CD0AF2              CALL   PCHK
F3EB  3F                  CCF
F3EC  C1                  POP    BC
F3ED  D0                  RET    NC
F3EE  280D                JR     Z,SU1
F3F0  C5                  PUSH   BC                ;SAVE COUNT
F3F1  E5                  PUSH   HL
F3F2  CDB4F1              CALL   EXF
F3F5  D1                  POP    DE
F3F6  E1                  POP    HL
F3F7  73                  LD     (HL),E
F3F8  78                  LD     A,B
F3F9  FE0D                CP     CR
F3FB  C1                  POP    BC
F3FC  C8                  RET    Z
F3FD  23            SU1:  INC    HL
F3FE  CD03F4              CALL   SU2               ;CHK FOR 8 SUBST.
F401  18DB                JR     SU0
F403  05            SU2:  DEC    B                 ;COUNT SUBSTITUTES
F404  C0                  RET    NZ
F405  CD7CF1              CALL   CRLF
F408  CDFDF0              CALL   BLK
F40B  CDFDF0              CALL   BLK
F40E  CDFDF0              CALL   BLK
F411  0608                LD     B,8
F413  C9                  RET

                    ;     REGISTER COMMAND
F414  CD5CF1        X:    CALL   TI
F417  21E6F4              LD     HL,ACTBL
F41A  FE0D                CP     CR
F41C  CA86F4              JP     Z,X6
F41F  FE27                CP     '''
F421  200B                JR     NZ,XA
F423  21FFF4              LD     HL,PRMTB
F426  CD5CF1              CALL   TI
F429  FE0D                CP     CR
F42B  CA86F4              JP     Z,X6
                    XA:
F42E  47                  LD     B,A
F42F  7E            X0:   LD     A,(HL)
F430  E67F                AND    7FH
F432  B8                  CP     B
F433  2808                JR     Z,X1
F435  7E                  LD     A,(HL)
F436  B7                  OR     A
F437  C8                  RET    Z
```

```
F438  23         0729       INC   HL
F439  23         0730       INC   HL
F43A  78         0731       LD    A,B
F43B  18F2       0732       JR    X0
F43D  CDFDF0     0733 X1:   CALL  BLK
F440  7E         0734 X2:   LD    A,(HL)
F441  E680       0735       AND   80H
F443  07         0736       RLCA
F444  47         0737       LD    B,A
F445  04         0738       INC   B
F446  23         0739       INC   HL
F447  7E         0740       LD    A,(HL)
F448  EB         0741       EX    DE,HL
F449  6F         0742       LD    L,A
F44A  2600       0743       LD    H,0
F44C  39         0744       ADD   HL,SP
F44D  EB         0745       EX    DE,HL
F44E  23         0746       INC   HL
F44F  1A         0747       LD    A,(DE)
F450  CDCBF1     0748       CALL  LBYTE
F453  05         0749       DEC   B
F454  2805       0750       JR    Z,X3
F456  1B         0751       DEC   DE
F457  1A         0752       LD    A,(DE)
F458  CDCBF1     0753       CALL  LBYTE
F45B  04         0754 X3:   INC   B
F45C  0E2D       0755       LD    C,'-'
F45E  CDFFF0     0756       CALL  CO
F461  CD0AF2     0757       CALL  PCHK
F464  3F         0758       CCF
F465  D0         0759       RET   NC
F466  2814       0760       JR    Z,X5
F468  E5         0761       PUSH  HL
F469  C5         0762       PUSH  BC
F46A  CDB4F1     0763       CALL  EXF
F46D  E1         0764       POP   HL
F46E  F1         0765       POP   AF
F46F  C5         0766       PUSH  BC
F470  F5         0767       PUSH  AF
F471  7D         0768       LD    A,L
F472  12         0769       LD    (DE),A
F473  C1         0770       POP   BC
F474  05         0771       DEC   B
F475  2803       0772       JR    Z,X4
F477  13         0773       INC   DE
F478  7C         0774       LD    A,H
F479  12         0775       LD    (DE),A
F47A  C1         0776 X4:   POP   BC
F47B  E1         0777       POP   HL
F47C  7E         0778 X5:   LD    A,(HL)
F47D  B7         0779       OR    A
F47E  C8         0780       RET   Z
F47F  78         0781       LD    A,B
F480  FE0D       0782       CP    CR
F482  C8         0783       RET   Z
```

```
F483  C340F4      0784         JP      X2
F486  CD7CF1      0785 X6:     CALL    CRLF
F489  EB          0786         EX      DE,HL
F48A  13          0787         INC     DE
F48B  1A          0788         LD      A,(DE)
F48C  6F          0789         LD      L,A
F48D  2600        0790         LD      H,0
F48F  39          0791         ADD     HL,SP
F490  46          0792         LD      B,(HL)
F491  2118F5      0793         LD      HL,FLGTBL
F494  CDFDF0      0794 X7:     CALL    BLK
F497  7E          0795 X7.5:   LD      A,(HL)
F498  23          0796         INC     HL
F499  B7          0797         OR      A
F49A  4F          0798         LD      C,A
F49B  78          0799         LD      A,B
F49C  07          0800         RLCA
F49D  47          0801         LD      B,A
F49E  2816        0802         JR      Z,X8
F4A0  FA97F4      0803         JP      M,X7.5
F4A3  CDFFF0      0804         CALL    CO
F4A6  0E3A        0805         LD      C,':'
F4A8  CDFFF0      0806         CALL    CO
F4AB  78          0807         LD      A,B
F4AC  E601        0808         AND     1
F4AE  C630        0809         ADD     '0'
F4B0  4F          0810         LD      C,A
F4B1  CDFFF0      0811         CALL    CO
F4B4  18DE        0812         JR      X7
F4B6  EB          0813 X8:     EX      DE,HL
F4B7  23          0814         INC     HL
F4B8  CD7CF1      0815         CALL    CRLF
F4BB  CDFDF0      0816 X9:     CALL    BLK
F4BE  7E          0817         LD      A,(HL)
F4BF  23          0818         INC     HL
F4C0  B7          0819         OR      A
F4C1  C8          0820         RET     Z
F4C2  47          0821         LD      B,A
F4C3  E67F        0822         AND     7FH                 ;REMOVE MSB FOR A
                  0823                                     ;CLEAN DISPLAY ON
                  0824                                     ;THE VB1B/C & VB3
F4C5  4F          0825         LD      C,A
F4C6  CDFFF0      0826         CALL    CO
F4C9  0E3D        0827         LD      C,'='
F4CB  CDFFF0      0828         CALL    CO
F4CE  7E          0829         LD      A,(HL)
F4CF  23          0830         INC     HL
F4D0  EB          0831         EX      DE,HL
F4D1  6F          0832         LD      L,A
F4D2  2600        0833         LD      H,0
F4D4  39          0834         ADD     HL,SP
F4D5  EB          0835         EX      DE,HL
F4D6  1A          0836         LD      A,(DE)
F4D7  CDCBF1      0837         CALL    LBYTE
F4DA  78          0838         LD      A,B
```

```
F4DB  E680      0839          AND    80H
F4DD  28DC      0840          JR     Z,X9
F4DF  1B        0841          DEC    DE
F4E0  1A        0842          LD     A,(DE)
F4E1  CDCBF1    0843          CALL   LBYTE
F4E4  18D5      0844          JR     X9
                0845
F4E6  4614      0846 ACTBL:   DEFB   'F',                FLOC+2
F4E8  4115      0847          DEFB   'A',                ALOC+2
F4EA  4213      0848          DEFB   'B',                BLOC+2
F4EC  4312      0849          DEFB   'C',                CLOC+2
F4EE  4411      0850          DEFB   'D',                DLOC+2
F4F0  4510      0851          DEFB   'E',                ELOC+2
F4F2  4830      0852          DEFB   'H',                HLOC+2
F4F4  4C2F      0853          DEFB   'L',                LLOC+2
F4F6  CD30      0854          DEFB   'M' OR 80H,         HLOC+2
F4F8  D034      0855          DEFB   'P' OR 80H,         PLOC+2
F4FA  D317      0856          DEFB   'S' OR 80H,         SLOC+2
F4FC  4903      0857          DEFB   'I',                ILOC+2
F4FE  00        0858          DEFB   0

F4FF  4608      0859 PRMTB:   DEFB   'F',                FPLOC+2
F501  4109      0860          DEFB   'A',                APLOC+2
F503  420B      0861          DEFB   'B',                BPLOC+2
F505  430A      0862          DEFB   'C',                CPLOC+2
F507  440D      0863          DEFB   'D',                DPLOC+2
F509  450C      0864          DEFB   'E',                EPLOC+2
F50B  480F      0865          DEFB   'H',                HPLOC+2
F50D  4C0E      0866          DEFB   'L',                LPLOC+2
F50F  CD0F      0867          DEFB   'M' OR 80H,         HPLOC+2
F511  D807      0868          DEFB   'X' OR 80H,         XLOC+2
F513  D905      0869          DEFB   'Y' OR 80H,         YLOC+2
F515  5202      0870          DEFB   'R',                RLOC+2
F517  00        0871          DEFB   0

F518  53        0872 FLGTBL:  DEFB   'S'
F519  5A        0873          DEFB   'Z'
F51A  58        0874          DEFB   'X'
F51B  48        0875          DEFB   'H'
F51C  58        0876          DEFB   'X'
F51D  56        0877          DEFB   'V'
F51E  4E        0878          DEFB   'N'
F51F  43        0879          DEFB   'C'
F520  00        0880          DEFB   0
```

```
                 ;*****  SYSTEM CONFIGURATION PACKAGE  *****
                 ;       LOGICAL DEVICE/DEVICE DRIVER TABLES
                 ;
                 ;       EACH 4 ENTRY TABLE LISTS THE ADDRESSES
                 ;       OF THE DRIVER ROUTINES TO BE USED FOR
                 ;       THE PHYSICAL DEVICES WHICH MAY ASSIGNED
                 ;       TO THAT LOGICAL DEVICE.

                 IOTAB:
                 ;       CONSOLE STATUS
                 ;
                 ;       RETURN WITH REGISTER A = 0 IF NO
                 ;       CONSOLE CHARACTER AVAILABLE.
F521   B7F5      CSTAB:  DW      TTSA    ;0 STATUS,SERIAL-A
F523   BCF5              DW      TTSB    ;1 STATUS,SERIAL-B
F525   C1F5              DW      KYS1    ;2 STATUS,PAR. KYBD
F527   C6F5              DW      KYS2    ;3 STATUS,VB3 KYBD

                 ;       CONSOLE INPUT
                 ;       RETURN CONSOLE INPUT CHARACTER
                 ;       IN REGISTER A.
F529   CBF5      CITAB:  DW      TTIA    ;0 DATA-IN,SERIAL-A
F52B   D0F5              DW      TTIB    ;1 DATA-IN,SERIAL-B
F52D   D5F5              DW      KYI1    ;2 DATA-IN,PAR. KYBD
F52F   DAF5              DW      KYI2    ;3 DATA-IN,VB3 KYBD

                 ;       CONSOLE OUTPUT
                 ;       OUTPUT BYTE IN REGISTER C
                 ;       TO CONSOLE OUTPUT DEVICE.
F531   F6F5      COTAB:  DW      TTOA    ;0 DATA-OUT,SERIAL-A
F533   FBF5              DW      TTOB    ;1 DATA-OUT,SERIAL-B
F535   05F6              DW      CRT     ;2 DATA-OUT,VB1B
F537   6AF6              DW      VB3O    ;3 DATA-OUT,VB3

                 ;       READER INPUT
                 ;       RETURN READER INPUT BYTE IN
                 ;       REGISTER A, CARRY OFF.  SET
                 ;       CARRY IF NO BYTE AVAILABLE.
F539   CBF5      RITAB:  DW      TTIA    ;0 READER-IN,SERIAL-A
F53B   D0F5              DW      TTIB    ;1 READER-IN,SERIAL-B
F53D   DFF5              DW      RDR     ;2 READER-IN,EXT RDR
F53F   DAF5              DW      KYI2    ;3 READER-IN

                 ;       PUNCH OUTPUT
                 ;       OUTPUT BYTE IN REGISTER C
                 ;       TO PUNCH DEVICE.
F541   F6F5      POTAB:  DW      TTOA    ;0 PUNCH-OUT,SERIAL-A
F543   FBF5              DW      TTOB    ;1 PUNCH-OUT,SERIAL-B
F545   00F6              DW      PUNCH   ;2 PUNCH-OUT,EXT PUNCH
F547   6AF6              DW      VB3O    ;3 PUNCH-OUT

                 ;       LISTING OUTPUT
                 ;       OUTPUT BYTE IN REGISTER C
                 ;       TO LISTING DEVICE.
F549   FBF5      LOTAB:  DW      TTOB    ;0 LIST DEV.,SERIAL-B
```

```
F54B  F6F5                   DW      TTOA    ;1 LIST DEV.,SERIAL-A
F54D  6AF6                   DW      VB30    ;2 LIST DEV.,VB3
F54F  05F6                   DW      CRT     ;3 LIST DEV.,VB1

                    ;        SET ASIDE SCRATCH RAM FOR MONITOR DATA.
                    ;
                    ;SET THE TOP OF MONITOR SCRATCH TO BE USED ON
                    ;POWER-UP. THE ADDRESS IS RETURNED IN REGISTERS
                    ;D&E. **NOTE** THIS ROUTINE IS NOT CALLED
                    ;INSTEAD THE RETURN ADDRESS IS PLACED IN REG.
                    ;D&E AND THE SUBROUTINE IS ENTERED BY A JUMP
                    ;INSTRUCTION. THE ROUTINE RETURNS BY PLACING
                    ;THE ADDRESS IN H&L REG. AND EXECUTING A PCHL
                    ;INSTRUCTION. REGISTER C MUST EQUAL A ZERO OR
                    ;A ONE.
F551  C36EF5        ADSCS:   JP      ADS2

                    ;
                    ;LOCATE THE 84  BYTES OF RAM USED BY THE
                    ;MONITOR. ADDRESS IS RETURNED IN H&L.
F554  C361F5        ADSCR:   JP      ADS1

                    ;
                    ;LOCATE THE IOBYT ADDRESS. ADDRESS IS RETURNED
                    ;IN H&L.
F557  C361F5        ADIOB:   JP      ADS1

                    ;
                    ;SET UP USER DEFAULT STACK. ADDRESS IS
                    ;RETURNED IN H&L.
F55A  210001        ADUST:   LD      HL,100H
F55D  C9                     RET

                    ;SPECIAL JUMP VECTOR.  REGISTER A=INDEX.
F55E  C317F6        JVTR:    JP      JENTRY
F561  C5            ADS1:    PUSH    BC
F562  D5                     PUSH    DE
F563  0E00                   LD      C,0
F565  116AF5                 LD      DE,ADS1A
F568  1804                   JR      ADS2
F56A  EB            ADS1A:   EX      DE,HL
F56B  D1                     POP     DE
F56C  C1                     POP     BC
F56D  C9                     RET
      (0800)        M2K      EQU     2*1024
      (C000)        M48K     EQU     48*1024
      (C000)        LIMIT    EQU     M48K    ;UPPER 16K FOR USER.
F56E  210000        ADS2:    LD      HL,0
F571  0608                   LD      B,M2K SHR 8
F573  09            ADS3:    ADD     HL,BC
F574  7C                     LD      A,H     ;GET HIGH ADDR.
F575  FEC8                   CP      (LIMIT+M2K) SHR 8
F577  DA7EF5                 JP      C,ADS4
F57A  26C0                   LD      H,(LIMIT) SHR 8
F57C  180B                   JR      ADS5
```

```
                  ;
F57E  7E          ADS4:   LD      A,(HL)   ;GET MEMORY BYTE
F57F  F3                  DI
F580  2F                  CPL
F581  77                  LD      (HL),A   ;TRY TO CHG. IT
F582  BE                  CP      (HL)     ;DID WE CHG. IT?
F583  2F                  CPL
F584  77                  LD      (HL),A   ;RESTORE MEMORY
F585  FB                  EI
F586  CA73F5              JP      Z,ADS3   ;IF RAM, TRY AGAIN
F589  2E00        ADS5:   LD      L,0
F58B  2B                  DEC     HL
F58C  EB                  EX      DE,HL
F58D  E9                  JP      (HL)

                  ;******* GENERAL PURPOSE I/O ROUTINES *******
                  ;THREE ROUTINES FOR CHECKING STATUS, INPUTTING
                  ;DATA AND OUTPUTTING DATA. REGISTERS A & C
                  ;ARE USED.
                  ;ENTER WITH:
                  ;       REG.-A = STATUS PORT ADDRESS.
                  ;       REG.-C = OUTPUT DATA (GPRT ONLY)
                  ;EXITS WITH:
                  ;       REG.-A = DATA READ (GDAT ONLY)
                  ;       REG.-A = STATUS CONDITION (GSTS ONLY)

                  ;GET STATUS CONDITION OF INPUT DEVICE
                  ;EXITS WITH REG.-A SET TO ZERO IF NO DATA YET.
F58E  C5          GSTS:   PUSH    BC
F58F  4F                  LD      C,A      ;SET STATUS ADDRESS
F590  ED78                IN      A,(C)    ;INPUT STATUS
F592  E601                AND     1        ;TEST LSB
F594  D601                SUB     1
F596  9F                  SBC     A,A
F597  C1                  POP     BC
F598  C9                  RET

                  ;GET INPUT DATA.
                  ;EXITS WITH REG.-A SET WITH THE DATA
F599  C5          GDAT:   PUSH    BC
F59A  4F                  LD      C,A      ;SET STATUS ADDRESS
F59B  ED78        GD:     IN      A,(C)    ;INPUT STATUS
F59D  E601                AND     1        ;TEST LSB
F59F  20FA                JR      NZ,GD    ;IF NO DATA,GO BACK
F5A1  0C                  INC     C        ;SET DATA ADDRESS
F5A2  ED78                IN      A,(C)    ;GET DATA
F5A4  E67F                AND     7FH      ;REMOVE PARITY
F5A6  C1                  POP     BC
F5A7  C9                  RET

                  ;PRINT OUT DATA.
                  ;ENTER WITH DATA IN REG.-C
F5A8  C5          GPRT:   PUSH    BC
F5A9  41                  LD      B,C
F5AA  4F                  LD      C,A      ;SET PORT ADDRESS
```

```
F5AB  ED78      GP:     IN    A,(C)    ;INPUT STATUS
F5AD  E680              AND   80H      ;PRINTER BUSY?
F5AF  20FA              JR    NZ,GP    ;BUSY?IF TRUE,JUMP BACK
F5B1  0C                INC   C        ;SET OUTPUT PORT
F5B2  78                LD    A,B
F5B3  ED79              OUT   (C),A    ;PRINT!
F5B5  C1                POP   BC
F5B6  C9                RET

                ;********** STATUS ROUTINES **********
                ;ALL THE MONITOR'S STATUS ROUTINES ARE HERE.
                ;       SERIAL-A
F5B7  3E00      TTSA:   LD    A,0      ;PORT 0
F5B9  C38EF5            JP    GSTS

                ;       SERIAL-B
F5BC  3E02      TTSB:   LD    A,2      ;PORT 2
F5BE  C38EF5            JP    GSTS

                ;       PARALLEL KEYBOARD
F5C1  3E06      KYS1:   LD    A,6      ;PORT 6
F5C3  C38EF5            JP    GSTS

                ;       VB3 KEYBOARD
F5C6  3EE0      KYS2:   LD    A,0E0H   ;PORT E0
F5C8  C38EF5            JP    GSTS

                ;********** INPUT ROUTINES **********
                ;ALL THE MONITOR'S INPUT ROUTINES ARE HERE.

                ;       SERIAL-A
F5CB  3E00      TTIA:   LD    A,0      ;PORTS 0&1
F5CD  C399F5            JP    GDAT

                ;       SERIAL-B
F5D0  3E02      TTIB:   LD    A,2      ;PORTS 2&3
F5D2  C399F5            JP    GDAT

                ;       PARALLEL KEYBOARD
F5D5  3E06      KYI1:   LD    A,6      ;PORTS 6&7
F5D7  C399F5            JP    GDAT

                ;       VB3 KEYBOARD
F5DA  3EE0      KYI2:   LD    A,0E0H   ;PORTS E0 & E1
F5DC  C399F5            JP    GDAT

                ;       READER ROUTINE
                ;THIS ROUTINE HAS A BUILT-IN TIMER AND WILL
                ;RETURN WITH THE CARRY SET IF NO DATA IS
                ;FOUND IN A SET PERIOD OF TIME.
F5DF  E5        RDR:    PUSH  HL       ;SAVE H&L
F5E0  210000            LD    HL,0     ;SET TIMER
F5E3  DB0E      RD1:    IN    A,(0EH)  ;GET STATUS
F5E5  E601              AND   1        ;TEST STATUS
F5E7  2808              JR    Z,RD2    ;JUMP OUT,IF DATA
```

```
F5E9  2B          1076          DEC    HL
F5EA  7C          1077          LD     A,H
F5EB  B5          1078          OR     L         ;TEST TIMER
F5EC  20F5        1079          JR     NZ,RD1    ;IF STILL TIME,GO BACK
F5EE  37          1080          SCF              ;TIME OUT,SET ERROR
F5EF  E1          1081          POP    HL
F5F0  C9          1082          RET
F5F1  DB0F        1083 RD2:     IN     A,(0FH)   ;GET DATA
F5F3  B7          1084          OR     A         ;CLEAR ANY CARRYS
F5F4  E1          1085          POP    HL
F5F5  C9          1086          RET

                  1087 ;********** OUTPUT ROUTINES **********
                  1088 ;ALL THE MONITOR'S OUTPUT ROUTINES ARE HERE.

                  1089 ;         SERIAL-A
F5F6  3E00        1090 TTOA:    LD     A,0       ;PORTS 0&1
F5F8  C3A8F5      1091          JP     GPRT

                  1092 ;         SERIAL-B
F5FB  3E02        1093 TTOB:    LD     A,2       ;PORTS 2&3
F5FD  C3A8F5      1094          JP     GPRT

                  1095 ;         PUNCH DEVICE
F600  3E0E        1096 PUNCH:   LD     A,0EH     ;PORTS E&F
F602  C3A8F5      1097          JP     GPRT

                  1098 ;         VBIB OUTPUT
F605  79          1099 CRT:     LD     A,C       ;GET DATA
F606  B7          1100          OR     A         ;CHECK FOR NULL
F607  C8          1101          RET    Z         ;RET IF TRUE
F608  FE7F        1102          CP     7FH       ;CHECK FOR RUB-OUT
F60A  C8          1103          RET    Z         ;RET IF TRUE
F60B  E5          1104          PUSH   HL
F60C  CD54F5      1105          CALL   ADSCR     ;GET ADDR. OF SCRATCH
F60F  2B          1106          DEC    HL
F610  2B          1107          DEC    HL        ;VIDEO POINTER
F611  2B          1108          DEC    HL
F612  2B          1109          DEC    HL
F613  2B          1110          DEC    HL        ;PNT TO VB1B SCRATCH
F614  C34DF7      1111          JP     VDTTY
```

```
                     ;**********VB3 OUTPUT**********
                     ;"VBIOS"; REV 0
                     ; WRITTEN BY BEN L GEE; DEC 9, 1979
                     ; MODIFIED BY MALCOLM T. WRIGHT,4-27-80
                     ;
                     ;VBIOS IS A DRIVER FOR THE SSM VB3 VIDEO BOARD.
                     ;VBIOS CONTAINS ONLY ENOUGH CODE TO RUN CP/M
                     ;AND SOME CP/M PROGRAMS.

                     ;
                     ;**      HARDWARE CONFIGURATION
                     ;        16 MHZ DOT CLOCK
                     ;        9 DOTS PER CHARACTER
                     ;        80 CHARACTER PROMS

                     ;**      DEFINE SYSTEM EQUATES.
      (C000)         VIDEO   EQU     0C000H  ;ADDRESS OF THE VB3
      (1000)         OFFSET  EQU     01000H  ;OFFSET TO ATTRIBUTE
      (0003)         NORMAL  EQU     3       ;NORMAL ALPHA ATTRIB.
      (0005)         CODE    EQU     5       ;SET 80 CHAR. MODE
      (00D0)         VTAC    EQU     0D0H    ;I/O ADDR. OF CRT CTRL
      (00E0)         KSTAT   EQU     0E0H    ;KEYBOARD STATUS
      (00E1)         KDATA   EQU     KSTAT+1 ;KEYBOARD DATA
      (0050)         NCOLS   EQU     80      ;SCREEN SIZE
      (0012)         NROWS   EQU     18      ;NO. OF TEXT LINES
      (0000)         SKEW    EQU     0
      (0106)         SCANF   EQU     262     ;SCANS PER FRAME
      (000C)         SCANR   EQU     12      ;SCANS PER DATA ROW
      (0071)         HCOUNT  EQU     113     ;SET HORIZ SCAN LINE
      (0000)         INTERL  EQU     FALSE   ;INTERLACED?

                     ;**      VIDEO DRIVER SUBROUTINES
                     ;
                     ;ENTRY POINTS ARE:
                     ;
                     ;               VB3MZ - TO INITALIZE THE VB3
                     ;               VB3O - TO OUTPUT A CHARACTER
                     ;CALLING SEQUENCES:
                     ;
                     ;               CALL    VB3MZ  ;INITIALIZE VB3
                     ;
                     ;
                     ;               LD      A,DATA  ;GET CHARACTER
                     ;               MOV     C,A     ;SAVE IN REG.C
                     ;               CALL    VB3O    ;OUTPUT IT
                     ;

                     ;VECTOR TO SPECIAL SUBROUTINES.
F617  B7             JENTRY: OR      A
F618  CA54F5                 JP      Z,ADSCR ;TOP OF SCRATCH RAM
F61B  3D                     DEC     A
F61C  CA2CF6                 JP      Z,RWCL  ;GET NCOL/NROW ADDR.
F61F  3D                     DEC     A
F620  CA41F6                 JP      Z,VB3IZ ;INITIALIZE VB3
F623  3D                     DEC     A
```

```
F624  CA77F6                JP      Z,VB3OUT;OUTPUT TO VB3
F627  3D                    DEC     A
F628  CA4DF7                JP      Z,VDTTY ;OUTPUT TO VB1
F62B  C9                    RET

              ;FIND SCRATCH CONSTANTS FOR NCOLS & NROWS.
              ;HL = ADDRESS OF NCOLS CONSTANT.
F62C  CD54F5  RWCL:         CALL    ADSCR
F62F  D5                    PUSH    DE
F630  11FEFF                LD      DE,VX3-ENDX+1
F633  19                    ADD     HL,DE   ;HL PNTS TO NCOLS
F634  D1                    POP     DE
F635  C9                    RET
              VB3MZ:
F636  D3E0                  OUT     (KSTAT),A ;ENABLE THE VB3 BOARD
F638  CD3EF6                CALL    VB3I     ;INITIALIZE CHIP
F63B  D3E1                  OUT     (KDATA),A ;DISABLE VB3
F63D  C9                    RET

              ; INITALIZE ALL OF THE CONTROL REGISTERS
F63E  2163F6  VB3I:         LD      HL,X80X18
F641  D3DE    VB3IZ:        OUT     (VTAC+14),A
F643  D3DA                  OUT     (VTAC+10),A      ;RESET THE VTAC
                            ;SEND HCOUNT
F645  7E                    LD      A,(HL)
F646  D3D0                  OUT     (VTAC+0),A
F648  23                    INC     HL
                            ;SEND SYNC POSITION
F649  7E                    LD      A,(HL)
F64A  D3D1                  OUT     (VTAC+1),A
F64C  23                    INC     HL
                            ;SEND NO. OF CHAR.
F64D  7E                    LD      A,(HL)
F64E  D3D2                  OUT     (VTAC+2),A
F650  23                    INC     HL
                            ;SEND NO. OF ROWS
F651  7E                    LD      A,(HL)
F652  D3D3                  OUT     (VTAC+3),A
F654  23                    INC     HL
                            ;SEND RASTER FILL-IN
F655  7E                    LD      A,(HL)
F656  D3D4                  OUT     (VTAC+4),A
F658  23                    INC     HL
                            ;SEND TOP MARGIN
F659  7E                    LD      A,(HL)
F65A  D3D5                  OUT     (VTAC+5),A
F65C  23                    INC     HL
                            ;SEND BOTTOM LINE NO.
F65D  7E                    LD      A,(HL)
F65E  D3D6                  OUT     (VTAC+6),A
F660  D3DE                  OUT     (VTAC+14),A ;START CRT CHIP
F662  C9                    RET

              ;VB3 PARAMETER TABLE
F663  70      X80X18: DB            HCOUNT-1 ;HCOUNT-1
```

```
                        ;INTERLACE + SYNC POSITION
F664  53                DB      (INTERL*128)+53H
                        ;RASTER LINES PER CHAR. + NO. OF CHAR.
                        IF      NOT INTERL
F665  5D                DB      (SCANR-1)*8+CODE
                        ENDIF
                        IF      INTERL
                        DB      (SCANR-2)*8+CODE
                        ENDIF
                        ;NO. OF ROWS DOWN THE SCREEN
F666  11                DB      SKEW*64+NROWS-1
                        ;FILL-IN THE RASTER FOR A FRAME
                        IF      NOT INTERL
F667  03                DB      (SCANF-256)/2
                        ENDIF
                        IF      INTERL
                        DB      (SCANF-513)/2
                        ENDIF
                        ;SET THE TOP MARGIN
F668  14                DB      (SCANF-NROWS*SCANR)/4+9
                        ;SET WHICH LINE IS BOTTOM
F669  11                DB      NROWS-1

              ; VB30 IS THE MAIN DISPLAY DRIVER FOR THE VB3.
F66A  D3E0    VB30:     OUT     (KSTAT),A ;TURN-ON VB3
F66C  E5                PUSH    HL
F66D  CD2CF6            CALL    RWCL      ;PNT TO CONSTANTS
F670  CD77F6            CALL    VB3OUT    ;OUTPUT TO VB3
F673  E1                POP     HL
F674  D3E1              OUT     (KDATA),A ;TURN-OFF VB3
F676  C9                RET
F677  DDE5    VB3OUT:   PUSH    IX        ;SAVE IX REGISTER
F679  E5                PUSH    HL        ;SWITCH HL WITH IX
F67A  DDE1              POP     IX        ;
F67C  D5                PUSH    DE        ;SAVE D,E,B,C REG.'S
F67D  C5                PUSH    BC        ;
F67E  CD27F7            CALL    CUROFF    ;TURN OFF THE CURSOR
F681  CD8CF6            CALL    PROCESS   ;PROCESS THE CHARACTER
F684  CD1EF7            CALL    CURON     ;TURN CURSOR BACK ON
F687  C1                POP     BC        ;RESTORE THE REGISTERS
F688  D1                POP     DE
F689  DDE1              POP     IX
F68B  C9                RET

              ; ON ENTRY, REG HL = LOGICAL CURSOR ADDRESS
              ; AND REG C = DATA.
              PROCESS:
F68C  79                LD      A,C       ;MOVE DATA TO REG A.
F68D  E67F              AND     7FH       ;STRIP PARITY/SET FLAGS
F68F  C8                RET     Z         ;SKIP NULLS.
F690  FEFF              CP      0FFH      ;RUB-OUT
F692  C8                RET     Z
F693  FE0D              CP      CR        ;CHECK FOR CTRL-CHAR.
F695  CAB5F6            JP      Z,DOCR    ;
F698  FE0A              CP      LF
```

```
F69A  CAB8F6      1268          JP      Z,DOLF
F69D  FE08        1269          CP      BS
F69F  CA0CF7      1270          JP      Z,LEFT   ;DO A BACKSPACE
F6A2  FE0C        1271          CP      FF
F6A4  CAC3F6      1272          JP      Z,DOFF

                  1273 ; MUST BE A DATA CHARACTER.
F6A7  E5          1274          PUSH    HL       ;SAVE LOGIC CUR. ADDR.
F6A8  CD32F7      1275          CALL    GETBA    ;GET PHYSICAL CUR/ADDR.
F6AB  71          1276          LD      (HL),C   ;PUT THE DATA THERE.
F6AC  110010      1277          LD      DE,OFFSET
F6AF  19          1278          ADD     HL,DE    ;ADDRESS OF ATTRIBUTE
F6B0  E1          1279          POP     HL       ;RESTORE LOGICAL CURSOR
F6B1  CD15F7      1280          CALL    RIGHT    ;MOVE CURSOR RIGHT.
F6B4  C0          1281          RET     NZ       ;

                  1282 ; PUT THE CURSOR ON COL.1 OF THE CURRENT LINE.
F6B5  2E00        1283 DOCR:    LD      L,0      ;PUT CURSOR ON COL 1
F6B7  C9          1284          RET

                  1285 ; PUT CURSOR ON THE NEXT LINE, BLANK THE LINE.
F6B8  CD03F7      1286 DOLF:    CALL    DOWN     ;MOVE CURSOR DOWN
F6BB  E5          1287          PUSH    HL       ;SAVE CURSOR ADDRESS.
F6BC  2E00        1288          LD      L,0      ;PUT CURSOR ON COL 1.
F6BE  CDD2F6      1289          CALL    EEOL     ;ERASE TO END OF LINE.
F6C1  E1          1290          POP     HL       ;RESTORE CURSOR ADDR.
F6C2  C9          1291          RET              ;RETURN TO CALLER.

                  1292 ; DO A FORM-FEED.
F6C3  210000      1293 DOFF:    LD      HL,0     ;START AT UPPER-LEFT
                  1294                           ;AND CLEAR SCREEN.

                  1295 ; ERASE FROM THE CURSOR POSITION TO THE END
                  1296 ; OF THE SCREEN.
F6C6  DD7E00      1297 EEOS:    LD      A,(IX)   ;NCOLS
F6C9  95          1298          SUB     L
F6CA  47          1299          LD      B,A
F6CB  DD7E01      1300          LD      A,(IX+1)
F6CE  94          1301          SUB     H
F6CF  4F          1302          LD      C,A
F6D0  1806        1303          JR      ERASE

                  1304 ; ERASE FROM THE CURSOR TO THE END OF THE
                  1305 ; CURRENT LINE.
F6D2  3E50        1306 EEOL:    LD      A,NCOLS
F6D4  95          1307          SUB     L
F6D5  47          1308          LD      B,A
F6D6  0E01        1309          LD      C,1

                  1310 ; ERASE THE SCREEN BEGINNING AT THE CURSOR
                  1311 ; POSITION.  THE NUMBER OF CHARACTERS TO
                  1312 ; ERASE IS IN REG BC.
                  1313 ERASE:
F6D8  E5          1314          PUSH    HL
F6D9  E5          1315 ERASE1:  PUSH    HL
```

```
F6DA  CD32F7                CALL   GETBA
F6DD  110010                LD     DE,OFFSET
F6E0  EB                    EX     DE,HL
F6E1  19                    ADD    HL,DE
F6E2  EB                    EX     DE,HL
F6E3  3E03                  LD     A,NORMAL ;USE NORMAL ATTRIBUTE
F6E5  3620         ERASE2:  LD     (HL),' '
F6E7  12                    LD     (DE),A   ;
F6E8  23                    INC    HL
F6E9  13                    INC    DE
F6EA  05                    DEC    B
F6EB  20F8                  JR     NZ,ERASE2
F6ED  E1                    POP    HL
F6EE  2E00                  LD     L,0
F6F0  CD03F7                CALL   DOWN
F6F3  0650                  LD     B,NCOLS
F6F5  0D                    DEC    C
F6F6  20E1                  JR     NZ,ERASE1
F6F8  E1                    POP    HL
F6F9  C9                    RET

                   ; MOVE THE CURSOR UP.
F6FA  7C           UP:      LD     A,H
F6FB  25                    DEC    H
F6FC  B7                    OR     A
F6FD  C0                    RET    NZ
F6FE  DD6601                LD     H,(IX+1)  ;GET NROWS VALUE
F701  25                    DEC    H         ;NROWS-1
F702  C9                    RET

                   ; MOVE THE CURSOR DOWN.
F703  24           DOWN:    INC    H
F704  DD7E01                LD     A,(IX+1)  ;NROWS
F707  BC                    CP     H
F708  C0                    RET    NZ
F709  2600                  LD     H,0
F70B  C9                    RET

                   ; MOVE THE CURSOR LEFT.
F70C  7D           LEFT:    LD     A,L
F70D  2D                    DEC    L
F70E  B7                    OR     A
F70F  C0                    RET    NZ
F710  DD6E00                LD     L,(IX)  ;NCOLS
F713  2D                    DEC    L         ;NCOLS-1
F714  C9                    RET

                   ; MOVE THE CURSOR RIGHT.
F715  2C           RIGHT:   INC    L
F716  DD7E00                LD     A,(IX)   ;NCOLS
F719  BD                    CP     L
F71A  C0                    RET    NZ
F71B  2E00                  LD     L,0
F71D  C9                    RET
```

```
                ; TURN THE CURSOR ON.
                ; THE LOGICAL ADDRESS OF THE CURSOR MUST BE
                ; IN REG HL.
F71E  7D        CURON:  LD      A,L
F71F  D3DC              OUT     (VTAC+12),A
F721  7C                LD      A,H
F722  D3DD              OUT     (VTAC+13),A
F724  D3D6              OUT     (VTAC+6),A ;SET CURSOR AT BOTTOM.
F726  C9                RET

                ; TURN THE CURSOR OFF.
                ; THE LOGICAL ADDRESS OF THE CURSOR IS RETURN-
                ; ED IN REG HL.
F727  DBD9      CUROFF: IN      A,(VTAC+9) ;READ COL. REGISTER
F729  6F                LD      L,A        ;TO REG L
F72A  DBD8              IN      A,(VTAC+8) ;READ ROW REGISTER
F72C  67                LD      H,A        ;TO REG H
F72D  3EFF              LD      A,0FFH     ;TURN CURSOR OFF
F72F  D3DC              OUT     (VTAC+12),A
F731  C9                RET

                ; CONVERT A LOGICAL ADDRESS TO A PHYSICAL
                ; ADDRESS.
                ; ON ENTRY, HL CONTAINS THE LOGICAL ADDRESS.
                ; ON EXIT, HL CONTAINS THE PHYSICAL ADDRESS.
                ; A LOGICAL ADDRESS IS IN THE FORM (ROW,
                ; COLUMN) WITH ROW IN REG H AND COLUMN
                ; IN REG L.  ROW MUST BE IN THE RANGE OF 0
                ; TO NROWS-1 AND COLUMN MUST BE IN
                ; THE RANGE OF 0 TO NCOLS-1.
                ; A PHYSICAL ADDRESS IS THE ACTUAL MEMORY
                ; ADDRESS IN THE VIDEO MEMORY.
F732  D5        GETBA:  PUSH    DE
F733  C5                PUSH    BC
F734  EB                EX      DE,HL
F735  14                INC     D
F736  0600              LD      B,0
F738  DD4E00            LD      C,(IX)     ;NCOLS
F73B  210000            LD      HL,0
F73E  B7                OR      A          ;CLEAR CARRY
F73F  ED42              SBC     HL,BC      ;BC=NCOLS,HL=-NCOLS
F741  09        GETBA2: ADD     HL,BC
F742  15                DEC     D
F743  20FC              JR      NZ,GETBA2
F745  19                ADD     HL,DE
F746  1100C0            LD      DE,VIDEO
F749  19                ADD     HL,DE
F74A  C1                POP     BC
F74B  D1                POP     DE
F74C  C9                RET
```

```
                  ;         VIDEO BOARD DRIVER
                  ;         THIS SUBROUTINE FACILITATES THE
                  ;         USE OF THE SSM VB1B/C VIDEO BOARD
                  ;         AND A VIDEO DISPLAY DEVICE AS A
                  ;         CONSOLE OUTPUT DEVICE.
                  ;         ASCII CHARACTERS PRESENTED TO THE
                  ;         SUBROUTINE IN THE C REGISTER ARE
                  ;         DISPLAYED ON THE SCREEN.  CERTAIN
                  ;         CHARACTERS, LISTED BELOW, RECEIVE
                  ;         SPECIAL TREATMENT.  ALL REGISTERS
                  ;         ARE PRESERVED BY THIS SUBROUTINE.
                  ;         VID IS THE BEGINNING ADDRESS ASSIGNED
                  ;         TO THE DISPLAY RAM LOCATED ON THE VB1
                  ;         BOARD.

     (E000)       VID       EQU     0E000H   ;START OF VIDEO

                  ;         THREE BYTES OF RAM ARE REQUIRED FOR
                  ;         HOUSEKEEPING.  THESE BYTES MUST BE
                  ;         IN AN AREA UNUSED BY OTHER PROGRAMS.
                  ;         MAIN ENTRY POINT
                  ;         THIS ENTRY POINT MAY BE USED IF
                  ;         THE CURSOR POINTER AND CHARACTER
                  ;         HOLD ARE AT LOCATIONS SPECIFIED
                  ;         IN REGISTERS H & L.
                  ;         THE USER MUST SUPPLY SUBROUTINE
                  ;         ENTRY CODE AS FOLLOWS:

                  ;ENTR:    PUSH    HL       ;SAVE HL
                  ;         LD      HL,PNTR  ;ADDR OF CURSOR PNTR
                  ;         JP      VDTTY    ;JOIN THIS CODE

F74D  D5          VDTTY:    PUSH    DE       ;SAVE DE
F74E  C5                    PUSH    BC       ;SAVE BC
F74F  5E                    LD      E,(HL)   ;LPTR
F750  23                    INC     HL       ;
F751  7E                    LD      A,(HL)   ;HPTR
F752  E603                  AND     3        ;CONVERT TO VIDEO
F754  C6E0                  ADD     A,(VID) SHR 8    ;RAM ADDRESS
F756  57                    LD      D,A      ;
F757  23                    INC     HL       ;
F758  46                    LD      B,(HL)   ;CHAR UNDER CURSOR
F759  EB                    EX      DE,HL    ;PNTR TO HL
F75A  70                    LD      (HL),B   ;RESTORE PREV CHAR

                  ;         IDENTIFY INPUT CHAR
F75B  79                    LD      A,C      ;NEW CHAR
F75C  FE0C                  CP      FF       ;
F75E  CA9EF7                JP      Z,VIDFF  ;FORM FEED
F761  FE0D                  CP      CR       ;
F763  CAABF7                JP      Z,VIDCR  ;CARRIAGE RETURN
F766  FE0A                  CP      LF       ;
F768  CAB2F7                JP      Z,VIDLF  ;LINE FEED
F76B  FE08                  CP      BS       ;
F76D  CA98F7                JP      Z,CRLT   ;BACKSPACE
```

```
F770  71          1464          LD      (HL),C   ;
F771  010100      1465 CRRT:    LD      BC,1

                  1466 ;        ADJUST CURSOR POINTER
F774  09          1467 CRADJ:   ADD     HL,BC    ;

                  1468 ;        CHECK FOR OVERFLOW
F775  7C          1469          LD      A,H      ;
F776  FEE4        1470          CP      (VID+1024) SHR 8
F778  C283F7      1471          JP      NZ,VIDRT
F77B  21C0E3      1472          LD      HL,VID+960
F77E  CDCAF7      1473          CALL    ROLL0
F781  1806        1474          JR      VIDR1    ;

                  1475 ;        COMMON EXIT CODE
                  1476 ;        NORMALIZE CURSOR POINTER
F783  26E3        1477 VIDRT:   LD      H,(VID+960) SHR 8
F785  7D          1478          LD      A,L      ;
F786  F6C0        1479          OR      0C0H     ;
F788  6F          1480          LD      L,A      ;
F789  7E          1481 VIDR1:   LD      A,(HL)   ;CHAR UNDER CURSOR
F78A  367F        1482          LD      (HL),7FH ;CURSOR
F78C  EB          1483          EX      DE,HL    ;PNTR TO DE
F78D  77          1484          LD      (HL),A   ;CHAR UNDER CURSOR
F78E  2B          1485          DEC     HL       ;
F78F  72          1486          LD      (HL),D   ;HPTR
F790  2B          1487          DEC     HL       ;
F791  73          1488          LD      (HL),E   ;LPTR

                  1489 ;        RESTORE REGISTERS, EXIT
F792  C1          1490          POP     BC       ;
F793  D1          1491          POP     DE       ;
F794  E1          1492          POP     HL       ;
F795  79          1493          LD      A,C      ;
F796  B7          1494          OR      A
F797  C9          1495          RET

                  1496 ;        PROCESS BACKSPACE
                  1497 ;        MOVE CURSOR LEFT ONE CHARACTER.
F798  01FFFF      1498 CRLT:    LD      BC,-1
F79B  C374F7      1499          JP      CRADJ

                  1500 ;        PROCESS FORM FEED
                  1501 ;        FILL SCREEN WITH SPACES,
                  1502 ;        LDE CURSOR TO TOP LEFT
F79E  2100E0      1503 VIDFF:   LD      HL,VID   ;
F7A1  E5          1504          PUSH    HL       ;
F7A2  3620        1505 VIDFC:   LD      (HL),' ' ;
F7A4  23          1506          INC     HL       ;
F7A5  7C          1507          LD      A,H      ;
F7A6  FEE4        1508          CP      (VID+1024) SHR 8 ;
F7A8  38F8        1509          JR      C,VIDFC  ;
F7AA  E1          1510          POP     HL       ;
```

```
                   ;         PROCESS CARRIAGE RETURN
                   ;         MOVE CURSOR TO BEGINNING
                   ;         OF LINE
F7AB  7D           VIDCR:    LD      A,L       ;
F7AC  E6C0                   AND     0C0H      ;
F7AE  6F                     LD      L,A       ;
F7AF  C383F7                 JP      VIDRT     ;

                   ;         PROCESS LINE FEED
                   ;         MOVE CURSOR DOWN ONE LINE,
                   ;         FILL NEW LINE WITH SPACES
F7B2  D5           VIDLF:    PUSH    DE        ;
F7B3  114000                 LD      DE,64     ;
F7B6  19                     ADD     HL,DE     ;
F7B7  7C                     LD      A,H       ;
F7B8  FEE4                   CP      (VID+1024) SHR 8
F7BA  D1                     POP     DE        ;
F7BB  C283F7                 JP      NZ,VIDRT          ;

                   ;         ROLL THE WHOLE DISPLAY UP ONE
                   ;         LINE.
F7BE  CDCAF7                 CALL    ROLL0
F7C1  7D                     LD      A,L       ;
F7C2  F6C0                   OR      0C0H      ;
F7C4  6F                     LD      L,A       ;
F7C5  26E3                   LD      H,(VID+960) SHR 8
F7C7  C383F7                 JP      VIDRT     ;

                   ;         ROLL SUBROUTINE
F7CA  D5           ROLL0:    PUSH    DE
F7CB  E5                     PUSH    HL
F7CC  1100E0                 LD      DE,VID;
F7CF  2140E0                 LD      HL,VID+64;
F7D2  7E           ROLL1:    LD      A,(HL)
F7D3  12                     LD      (DE),A    ;
F7D4  3620                   LD      (HL),' ' ;
F7D6  13                     INC     DE
F7D7  23                     INC     HL        ;
F7D8  7C                     LD      A,H       ;
F7D9  FEE4                   CP      (VID+1024) SHR 8
F7DB  20F5                   JR      NZ,ROLL1;
F7DD  E1                     POP     HL        ;
F7DE  D1           ROLL2:    POP     DE        ;
F7DF  C9                     RET               ;

F7E0  (0000)                 END

Errors             0
```