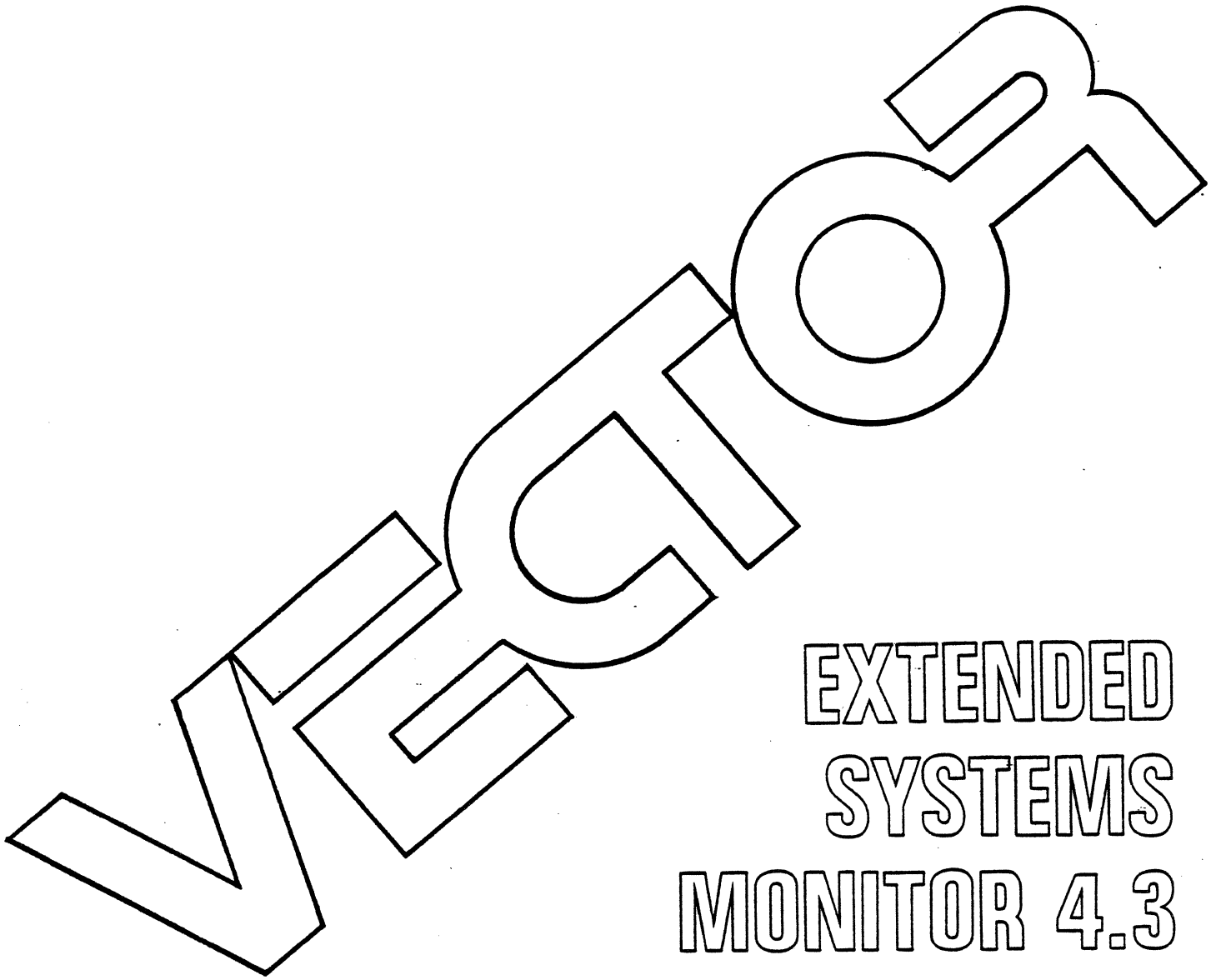

EXTENDED SYSTEMS MONITOR 4.3

User's Manual

EXTENDED SYSTEMS MONITOR

Version 4.3

USERS MANUAL

Revision A

July 10, 1981

P/N 7100-0245-00-00

Copyright 1981 Vector Graphic Inc.
Made in U.S.A.

Copyright 1981 by Vector Graphic Inc.
All rights reserved.

Disclaimer

Vector Graphic makes no representations or warranties with respect to the contents of this manual itself, even if the product it describes is covered by a warranty or repair agreement. Further, Vector Graphic reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Vector Graphic to notify any person of such revision or changes, except when an agreement to the contrary exists.

Revision Numbers

The date and revision of each page herein appears at the bottom of each page. The revision letter such as A or B changes if the manual has been improved but the product itself has not been significantly modified. The date and revision on the Title Page corresponds to that of the page most recently revised. When the product itself is modified significantly, the product will get a new revision number, as shown on the manual's title page, and the manual will revert to revision A, as if it were treating a brand new product. THIS MANUAL SHOULD ONLY BE USED WITH THE PRODUCT(S) IDENTIFIED ON THE TITLE PAGE.

TABLE OF CONTENTS

GENERAL DESCRIPTION

The Version 4.3 Monitor is a complete systems Monitor, able to support the Flashwriter II (80 X 24) board, and the Vector Graphic Keyboard. Thus it is recommended for use with the Mindless Terminal. All keyboard and video I/O can be done through the Monitor's I/O routines, freeing higher level software from carrying a variety of versions for different hardware configurations. Version 4.3 was designed to be used with the Flashwriter II board. Use Version 4.0C for serial terminals.

Version 4.3 differs from 4.2 in that the serial port initialization routine has been slowed down to accomodate Vector systems using 6 MHz. ZCB boards. 4 MHz. ZCB boards are also appropriate with this Monitor program.

In addition to I/O, the Monitor includes an extensive command executive, a compactly written program designed to facilitate manipulation and display of memory data. The "prompt" which indicates that the Monitor Executive is waiting for operator entry is "Mon>".

There are 26 commands which are entered as a single letter followed by up to four hexadecimal data fields. After each field is entered, a space is automatically output as a prompt. Either upper or lower case alpha characters may be used, but lower case characters will be converted to upper case, and any non-hex characters will be ignored. Allowable hex characters are 0-9, A-F. Address fields are four digits long; other fields are two digits long. The executive is useful in debugging hardware and software, particularly assembly language software, because it is resident in the system.

If a space is typed at any time during field entry, a default value of zero is assumed for all leading zeroes. This applies to an entire field as well as one that has been partially entered, and the cursor will advance to the next field if required. For example, typing (SP) will have the same effect as typing 0000; typing 100(SP) will have the same effect as 0100.

Any command that generates a display can be temporarily halted with a space and continued with another space. The ESCape key will abort a display or command entry.

The 4.3 Monitor is located at address E000H - E7FFH in Vector Graphic systems.

The hexadecimal number system may seem confusing if you are not familiar with it, but it has become the standard of the microcomputer field and is clearly the best system with 16 bit addresses and 8 bit data. It is usually not necessary to convert between number systems, as this is usually done by software (i.e. assemblers). Remembering a few values in hex should make things easy:

HEX NUMBER	DECIMAL VALUE	JARGON	BINARY BITS
A	10		4
B	11		4
C	12		4
D	13		4
E	14		4
F	15		4
10	16		5
FF	255		8
100	256	1 PAGE	9
3FF	1,023		10
400	1,024	1K	11
FFF	4,095		12
1000	4,096	4K	13
4000	16,384	16K	15
8000	32,768	32K	16
FFFF	65,535	64K-1	16

The familiar rules of arithmetic work just the same in hex as in decimal:

```
    10          Hex (trivial)
40) 400
```

COMMAND FORMAT

Mon>A <ADR1> <ADR2> - ASCII DUMP

Memory contents from ADR1 through ADR2 will be displayed as ASCII characters, or graphic symbols for values less than 20 hex. If the most significant bit is high, reverse video is displayed. This command is useful for examining files such as those created by SCOPE, BASIC or MEMORITE. ASCII strings embedded in object code are easy to recognize.

Mon>B - BOOT FLOPPY

Typing this command causes a jump to location E80CH which is located on the disk boot PROM. This will cause the disk operating system to be loaded into memory and transfer control to CP/M. This is designed to be used with a Vector system using the DualMode controller board. If a Micropolis Disk Controller board is present in the system, it may be accessed by typing G F800 in response to the "Mon>" prompt.

Mon>C <ADR1> <ADR2> <ADR3> - COMPARE BLOCKS

A byte-by-byte comparison will be made between the block of memory data starting at ADR1 and ending at ADR2 and a block of identical length starting at ADR3. The differences will be printed out with the address, the byte in the first block and the byte in the second block. This command is useful to compare two versions of a program or to verify that proms have been programmed correctly.

Mon>D <ADR1> <ADR2> - DUMP IN HEX

Memory contents from ADR1 through ADR2 will be displayed as pairs of hexadecimal characters. The left character in each pair represents the four most significant bits of the memory location. The display may be halted and interrupted as described above. The ASCII representation is displayed in a column on the right.

Mon>E - EXTERNAL COMMUNICATIONS

The monitor will output anything typed on the keyboard through port 4 on the ZCB single board computer, the Bitstreamer II I/O board or an appropriately addressed Bitstreamer I board. Anything received on this port will be displayed on the screen. Normally a 300 baud modem would be connected to the serial RS 232 output from the I/O board, and this feature allows the system to be used as a simple terminal to communicate with a host in a full duplex mode. Operation at speeds above 300 baud requires the host to send null characters after linefeeds, so that characters are not lost when the screen scrolls up.

Mon>F <ADR1> <ADR2> <BYTE1> <BYTE2> - FIND TWO BYTES

This memory range from ADR1 through ADR2 will be searched for the particular code combination BYTE 1 BYTE 2. This is useful for locating particular commands or jump addresses. For example, if you wish to change a control character (say control D) in a program you may try FE 04, which is CPI 04 since this is a common way of testing input characters. If you wish to find all locations that call or jump to a particular address, say C700H, then search for 00C7. There is no guarantee that each location displayed is valid object code - it may be part of a data table, ASCII string, or second and third bytes of a three byte instruction.

Mon>G <ADR1> - GO TO AND EXECUTE

This command will cause a jump to ADR1 to execute a program or user subroutine. As with all Monitor jump commands, the address contained on the stack is "START" (E04CH) and if the user routine at ADR1 ends in "RET", program execution will return to the Monitor. Approximately 96 levels of stack space is available, but of course, pushing more registers on the stack than are popped will defeat the return feature with undesirable effects.

Mon>H - JUMP TO HI RAM

This command jumps to FC00H which is the start of the 1K scratchpad RAM. This is a useful area for small machine language programs.

Mon>I <PORT> - INPUT FROM A PORT

Execution of this command will cause the CPU to execute an "IN PORT" instruction and the accumulator contents immediately following this to be displayed. This command is useful in checking out peripheral equipment. Only those ports used by the terminal, cassette interface, etc., will contain interesting values. All others will read FF since the data bus will be floating when the "IN" command is executed.

Mon>J - JUMP TO LOADED DOS

This command permits easy return to the MDOS disk operating system at 04E7H, or if not present, jump will be 0000H, which is the CP/M warm start location.

Mon>K - SET BREAKPOINTS

This command expects a 4 digit address, and will place a RESTART 7 (FF) at that location in RAM. When that instruction is executed, which is a call to location 0038H, the CPU will jump to the monitor routine that dumps the register contents. The instruction replaced with FF will also be restored. If a program is loaded over 0038H, the breakpoint instruction will be defeated unless RESET is depressed. Entry of the monitor at E000H will clear the breakpoint, as will pressing the RESET switch.

Mon>L - JUMP TO LOW RAM AT 0000H

This command jumps to memory location 0000H which is the beginning of program memory. This is the CP/M warm start location.

Mon>M <ADR1> <ADR2> <ADR3> - MOVE MEMORY BLOCK

The data contained in memory starting at ADR1 and ending at ADR2 is moved to memory locations starting at ADR3. This command is useful for moving a program from a temporary storage location to its correct address. If there is an overlap of the two memory areas, interesting results are obtained. For example, M 6000 7BFF 6400 will cause the block of data from 6000H through 63FFH to be repeated 8 times from 6000H through 7FFFH, since by the time location 6400H is read, it has been overwritten with data from 6000H. This is useful for bank programming of proms, or for creating repeating instruction sequences for test purposes.

Mon>N - NON-DESTRUCTIVE MEMORY TEST

Memory locations starting at 0000H are read and the data temporarily stored. The memory location is then tested to see if 00 and FF can be written and read correctly. This continues after rewriting the original data until the first error is detected, whereupon the address is displayed followed by the data written into memory and what was read from it. This command is most useful for checking how much memory a system contains. For example, if the system contains 16K of memory, 4000 00 FF should be printed, indicating that there is no memory at address 4000H. Since the test is non-destructive to data in memory, it can be used at any time.

Mon>O <PORT> <DATA> - OUTPUT TO PORT

The two hex digits "DATA" are loaded into the accumulator and the instruction "OUT PORT" is executed. This command is useful for checking out peripheral equipment. For example, if a printer is connected to I/O port 6, O 06 41 will cause an "A" to be printed since 41 is the hex ASCII code for "A".

Mon>P <ADR1> - PROGRAM MEMORY

The contents of 16 bytes of memory containing ADR1 are displayed in both hex and ASCII, allowing preceding and following instructions to be viewed. Advancing to the next instruction is accomplished by typing space or cursor right (right arrow). Backspace or cursor left (left arrow) goes backwards. The cursor up and down keys move to an adjacent 16 byte block. Any hex characters typed will replace the existing contents of RAM. After every keypress, the screen display is refreshed by reading from memory, so the display reflects the exact memory contents. To terminate, depress ESCAPE.

Mon>Q <ADR1> <ADR2> - COMPUTE CHECKSUM

The MOD 256 checksum of memory contents in the address range specified is computed and displayed. This command is useful for checking proms or files to see if anything has changed. Any source file or program written in pure code (it does not write on itself) will have the same checksum as when it was loaded. While debugging assembly language programs, it is useful to be able to verify that a program being debugged has not written garbage in the source file or assembler.

Mon>R - REGISTER DUMP

This command will print a header identifying the Z-80 registers, and immediately below it the contents of all the registers. The flags are displayed with the letters Z C M E H for the zero, carry, minus, parity even, and auxiliary or half carry flags respectively. The presence of the letter indicates the flag is true. The contents of the memory locations pointed to by the B, D, and H register pairs are also displayed as is the return address on the stack.

Mon>S <ADR1> <ADR2> <BYTE> - SEARCH FOR SINGLE BYTE

This is similar to the "F" command, except that only one byte is searched for instead of two. An example of the use of this command is to display all locations in a program where an output to a port occurs (D3). The address of each location will be displayed followed by "D3" and the next byte (the port number).

Mon>T <ADR1> <ADR2> - TEST MEMORY

This is an extremely useful command, especially when first setting up a system. This command permits thorough testing of the system memory. A portion of a 64K byte pseudorandom number sequence is written into memory from ADR1 through ADR2, and the exact same sequence is regenerated from the initial point and compared with what is read from memory. If all locations compare, another portion of the sequence is used to repeat the test which continues until it is interrupted. Any memory errors are displayed with the address, what was written into memory and what was read from memory, respectively. This information is all that is needed to pinpoint a malfunctioning memory chip. This test is quite exhaustive if used for at least 10 cycles and is far superior to incrementing or complementing tests which may not reveal addressing problems. The only area of system memory that cannot be tested with this routine is the few bytes required for the stack and video flags in the vicinity of FFD0H on the ZCB board.

Mon>U - JUMP TO 0100H

This command permits easy return to programs in the transient program area of CP/M.

Mon>V - 8" DRIVE BOOT

Typing this command will cause a jump to E800H (contained on all current Disk Boot PROMs) which is the location of the 8" drive bootstrap loader. The boot program will cause the CP/M operating system to be loaded into memory and control to be transferred to CP/M.

Mon>W - WINCHESTER DRIVE BOOT

Typing this command will cause a jump to E802H (contained on all current Disk Boot PROMs) which is the location of the Winchester drive bootstrap loader. The boot program will cause the CP/M operating system to be loaded into memory and control to be transferred to CP/M.

Mon>X <ADR1> <ADR2> <ADR3> - EXCHANGE MEMORY BLOCKS

A block of memory from ADR1 through ADR2 is exchanged with an equal length block starting at ADR3. This command is useful in comparing the operation of two versions of a program, or for rapid switching of portions of a program without destroying the original. A loaded BASIC program can be exchanged with another if care is used to include the stack area (usually below the top of allowed memory).

Mon>Y - KEYBOARD ECHO

This command causes keyboard input to be echoed directly to the video driver and can be used for demonstration purposes. An ESCape returns to the Monitor.

Mon>Z <ADR1> <ADR2> <DATA> - ZERO OR FILL MEMORY

The memory block from ADR1 through ADR2 is filled with the byte "DATA". This is useful for setting memory to Zero. The end of a file or assembled program will stand out more clearly if memory is first zeroed. For test purposes, single instructions can be executed continuously so that bus waveforms are more easily interpreted. This is done by filling a block of memory with a repeated instruction sequence with a jump to the start of the block so that the program loops continuously.

ENTRY POINTS

A jump table at the beginning of the Monitor can be used to access several routines:

E000 - The normal cold entry point to the Monitor Executive, this is a jump to the initialization routine which clears the screen and initializes 8251 USARTS through I/O ports 3, 5, and 7. This is compatible with the Bitstreamer I addressed starting at port 4 ,the Bitstreamer II addressed starting at port 2 or all ZCB's with standard port addressing. The USARTS are set for an X16 baud rate factor and other parameters as would be used with a serial printer or extra terminal.

E003 - This is a jump to the routine which should be used for console keyboard status test. Return with the zero flag set indicates no keyboard input.

E006 - This is a jump to the keyboard data input which returns with the character in the "A" register. The keyboard code conversions described below are carried out. There is no checking for ESC key depression.

E009 - This is a jump to the video driver which displays the character in "A" on the screen.

E00C - This is a jump to the "ESCAPE" routine which returns zero if no input, or with the character in the "A" register if there is. Keyboard code conversions are carried out. If the ESC key was pressed, the system returns to the Monitor Executive.

VIDEO DRIVER

Version 4.x of the Monitor contains a more elaborate video driver than previous versions. The purpose of the video driver is to accept a stream of ASCII codes, and to write them into the screen memory in the proper place, interpreting certain non printing control codes in a special way. There are several entry points to the video driver. E009H is recommended. The character code to be printed must be in the A register. A CALL E009 will cause the character to be printed on the screen at the cursor position. All registers will be preserved.

Control codes are generated by the keyboard by holding the contrgd (CTRL) key down while a letter key is pressed. Control codes have values between 0 and 31, and are 64 less than the codes for the corresponding upper case letters. To demonstrate the features of the video driver, type Y after the Monitor prompt, and any keyboard generated code will be echoed to the video driver. The following control codes are interpreted as special functions, while all others are ignored:

Decimal Value	Hex Value	Control Code	Description
2	2	(^B)	HOME THE CURSOR
4	4	(^D)	CLEAR THE SCREEN AND HOME CURSOR
5	5	(^E)	DISPLAY THE CODE IN B REGISTER
8	8	(^H)	DESTRUCTIVE BACKSPACE (also BACKSPACE key)
9	9	(^I)	TAB OVER TO THE NEXT 8 MULTIPLE (also TAB)
10	A	(^J)	LINEFEED (also LF Key)
13	D	(^M)	CARRIAGE RETURN (also RETURN key)
14	E	(^N)	TOGGLE CURSOR
16	10	(^P)	CLEAR TO END OF SCREEN
17	11	(^Q)	CLEAR TO END OF LINE
18	12	(^R)	CURSOR DOWN
20	14	(^T)	TOGGLE REVERSE VIDEO
21	15	(^U)	CURSOR UP
23	17	(^W)	CURSOR LEFT
24	18	(^X)	CLEAR TO START OF LINE
26	1A	(^Z)	CURSOR RIGHT
27	1B	ESC	CURSOR XY POSITION LEAD-IN

Experiment with the keys. There are special keys on the keyboard to generate some of the codes such as RETURN, TAB and linefeed (LF). If you are using the Vector Graphic Keyboard or Mindless Terminal, there are also keys for the cursor control and BACKSPACE. A few of the functions are not self explanatory. A Control D sets the reverse video flag to normal in addition to clearing the screen and homing the cursor. A Control T will then toggle the reverse video flag from normal to reverse and back without printing on the screen.

In some cases it is desirable to print the symbol for a control code on the screen. This can be done in assembly language programs by putting the code for the symbol in the B register and calling the video driver with Control E (05) in A. Enter the following machine code at FC00H and execute it to demonstrate this feature:

```
at FC00  06 01 3E 05 04 CD 09 E0 CD 0C E0 C3 02 FC
```

CURSOR X Y POSITIONING

Many programs utilize random X Y positioning of the cursor. This is done by outputting a three byte sequence to the video driver. The first code is ESC (1BH) followed by the desired X position and Y position in hex. The top left corner of the screen is 0, 0. The assembly language sequence 1B 40 08 would cause the cursor to move to line 8, character position 64 on the screen. To send the same sequence to the Monitor via Microsoft Basic, the following statement would be used: "PRINT CHR$(27);CHR$(X+128);CHR$(Y+128);" where X would equal 64 (40H) and Y would equal 08 (08H). Adding the value of 128 to X and Y in this example sets the eighth bit high. This is done to avoid Microsoft Basic from confusing the values as control codes. This may not be demonstrated using the keyboard since ESC causes a return to the monitor.

The video driver provides an extensive range of special controls, however, they must be incorporated into the software generating the video stream to be meaningful. For instance a piece of software that merely echoes all characters as they go into its input buffer will allow cursor motion on the screen, but this will probably be meaningless to the software.

KEYBOARD CODE CONVERSION - VECTOR GRAPHIC KEYBOARDS

Due to limitations in the keyboard encoder chip, the [] key on Vector Graphic keyboards is not encoded properly. The correct code is generated by a conversion routine in the Monitor's CONVERT routine. The codes for backslash and tilde are also produced by the control and control shift mode of this key.

[] KEY CONVERSION:

MODE	KEYCODE	CONVERTED CODE	ASCII SYMBOL
unshifted	F1	5B	[
shifted	E1	5D	]
control	B1	5C	\
control shift	A1	7E	~

The cursor up key is also converted from 60H to 15H which is interpreted correctly by the video driver. Room is provided in the routine for up to 15 keycode conversions. Foreign languages require additional conversions, and versions are available for French, German, Swedish and Spanish. It is

essential that software utilize the monitor conversion routine for this reason.

USING THE I/O ROUTINES

The I/O routines in the Monitor are used as the Main System I/O in Vector Graphic Systems. This makes software I/O independent and easily interchangeable between systems. An example of how this is done is shown below:

```
INPUT ROUTINE:      INPT    CALL E00CH
                            JZ   INPT
                            RET  (RETURNS WITH CHAR INPUT IN A)

OUTPUT ROUTINE:     OUTPT   JMP  E009H (CHARACTER IN A)

BREAK TEST:         CONTL   CALL E00CH
                            RET  (RETURNS WITH ZERO FLAG SET IF NO
                                 INPUT, OR CHARACTER IN A.  JUMPS
                                 TO MONITOR EXECUTIVE IF ESCAPE
                                 INPUT.)
```

Note that the ESC key will break to the Monitor, which provides a convenient way of transferring control from any executive such as the DOS or BASIC to the Monitor, but necessitates the use of another character (Control C is standard) for a single level break. The routines above are merely given to illustrate how simple it is to use the Monitor I/O routines. Many programs require additional instructions to move the character to be output into the accumulator, or may require different flag conditions or accumulator contents on return from the input and Break Test routine, but the variations are easily implemented.

OTHER USEFUL MONITOR ROUTINES

The Monitor contains a number of routines that can be called by user programs, and which will save considerable programming effort. In addition to the keyboard input and video output described elsewhere, we have:

AHEX inputs four hex digits from the keyboard and returns the binary value in D,E registers. A space is automatically output at the end. All registers, except B, are used. Entry at AHE0 with a value of 1-3 in C will convert that many digits. Non hex values will be ignored.

CRLF will output a carriage return and line feed to the screen. The A register is used.

SPCE will output a space to the screen. The A register is used.

RNDM returns a new random number in B,C based on the seed in B,C as it is called. B,C should not contain 0000. The pseudorandom number sequence generated is $2^{16}-1$ entries long and is based on a software simulation of a shift register with maximum length feedback. PSW is used.

PTAD first outputs a CRLF, then outputs the binary value in H,L as four hex digits followed by a space. PSW used.

PT2 outputs (A) as two hex digits.

TAHEX calls AHEX twice, inputting two address fields of four hex digits. The first value is returned in H,L; the second in D,E.

The addresses of these routines and others may be found by consulting the listing which follows.

```
0000  E000 =  BASE          EQU     0E000H          ;ASSEMBLY ADDRESS
0000  E000 =  PR            EQU     0E000H          ;PROM/RAM ADDRESS
0000                        LINK    'M6'
0000          ************************************************
0000          *                                              *
0000          *        VECTOR MZ MONITOR - VERSION 4.3       *
0000          *      R. S. HARP  7/16/79  MODIFIED 1/12/81   *
0000          *                                              *
0000          ************************************************
0000          *
0000          * SYSTEM EQUATES
0000  0000 =  CONS          EQU     0               ;CONS STATUS PRT
0000  0001 =  COND          EQU     1               ;CONS DATA PORT
0000  0040 =  RDA           EQU     40H             ;RECEIVE FLAG
0000  0000 =  STPOL         EQU     0               ;STATUS POLARITY
0000  FFD0 =  SPTR          EQU     PR+01FD0H       ;STACK POINTER
0000  E800 =  DSBOOT        EQU     0E800H          ;DUALSTOR BOOTSTRAP
0000  E802 =  MSBOOT        EQU     0E802H          ;MEGASTOR BOOTSTRAP
0000  E80C =  FLBOOT        EQU     0E80CH          ;FLOPPY BOOTSTRAP
0000  FF10 =  DBUSY         EQU     0FF10H          ;CONTROLLER BUSY
0000          *
0000          ***************** COMMAND FORMAT ******************
0000          *    A SSSS FFFF ASCII DUMP OF MEMORY             *
0000          *    B JUMP TO BOOTSTRAP LOADER                   *
0000          *    C SSSS FFFF CCCC COMPARE BLOCKS              *
0000          *    D SSSS FFFF DUMP MEMORY IN HEX & ASCII       *
0000          *    E EXTERNAL COMMUNICATIONS                    *
0000          *    F SSSS FFFF DD DD TWO BYTE SEARCH            *
0000          *    G SSSS GO TO AND EXECUTE                     *
0000          *    H JUMP TO HIGH RAM AT FC00                   *
0000          *    I PP INPUT FROM PORT                         *
0000          *    J JUMP TO DOS                                *
0000          *    K LLLL SET A BREAKPOINT                      *
0000          *    L JUMP TO LOW RAM AT 0                       *
0000          *    M SSSS FFFF DDDD MOVE BLOCK                  *
0000          *    N NON DESTRUCTIVE MEMORY TEST                *
0000          *    O PP DD OUTPUT TO PORT                       *
0000          *    P LLLL PROGRAM MEMORY                        *
0000          *    Q SSSS FFFF COMPUTE CHECKSUM                 *
0000          *    R DUMP Z-80 REGISTERS                        *
0000          *    S SSSS FFFF DD SEARCH FOR SINGLE BYTE        *
0000          *    T SSSS FFFF TEST MEMORY                      *
0000          *    U JUMP TO USER AREA AT 100H                  *
0000          *    V BOOT FROM 8 INCH DISK                      *
0000          *    W BOOT WINCHESTER DISK                       *
0000          *    X SSSS FFFF DDDD EXCHANGE BLOCK              *
0000          *    Y KEYBOARD ECHO                              *
0000          *    Z SSSS FFFF DD ZERO OR FILL MEMORY           *
0000          ************************************************
0000          *
0000                        ORG     BASE
E000          * JUMP TABLE OF ENTRY POINTS
E000 C315E0   MONIT         JMP     INIT            ;INITIALIZE ALL
E003 C33CE1   KEYTST        JMP     KEYSTAT         ;TEST KEYBOARD
E006 C341E1   KEYDATA       JMP     CONVERT         ;INPUT KEYBOARD
E009 C37BE3   CRT           JMP     VIDEO           ;OUTPUT TO SCREEN
E00C C32FE1   ESC           JMP     ESCAPE          ;KEYBOARD INPUT
```



```
E00F          *
E00F          * TABLE OF COMMANDS FOR USART
E00F 00000040 INITABLE      DB      0,0,0,40H,0CEH,27H
E013 CE27
E015          *
E015 31D0FF   INIT          LXI     SP,SPTR         ;INIT STACK
E018 CD2FE1                 CALL    ESCAPE          ;DUMP LATCH
E01B AF                     XRA     A
E01C 32EAFF                 STA     XYFLAG
E01F 3210FF                 STA     DBUSY           ;CLEAR CONTROLLER FLAG
E022          * INITIALIZE USARTS AT PORTS 3,5,7
E022 3E03                   MVI     A,3             ;STARTING PORT
E024 4F                     MOV     C,A
E025 0606     INILOOP       MVI     B,6             ;NO OF COMMANDS
E027 210FE0                 LXI     H,INITABLE
E02A EDA3     OUTLOOP       OUTI                    ;OUTPUT A BYTE
E02C E3                     XTHL                    ;DELAY FOR 6 MIZ.
E02D E3                     XTHL
E02E 20FA                   JRNZ    OUTLOOP         ;SEND NEXT BYTE
E030 0C                     INR     C
E031 0C                     INR     C
E032 3D                     DCR     A               ;DO 3 PORTS IN ALL
E033 20F0                   JRNZ    INILOOP
E035          * PATCH RST 7
E035 3EC3                   MVI     A,0C3H          ;JUMP
E037 323800                 STA     38H             ;RST 7
E03A 21CBE6                 LXI     H,DUMPREGS
E03D          * DISPLAY SIGN ON
E03D CDCFE4                 CALL    SIGN
E040          * CLEAR BREAKPOINT
E040 2AE7FF   CLRBRK        LHLD    BKPTLOC
E043 11E9FF                 LXI     D,BRKCODE
E046 ED53E7FF               SDED    BKPTLOC
E04A 1A                     LDAX    D
E04B 77                     MOV     M,A
E04C 31D0FF   START         LXI     SP,SPTR         ;INITIALIZE STACK
E04F 2100F0                 LXI     H,PAGE          ;FULL SCREEN SCROLL
E052 22DFFF                 SHLD    TOSCN
E055 CD2EE5                 CALL    PROMPT
E058 CD2FE1   KEYPOL        CALL    ESCAPE          ;READ KEYBOARD
E05B 28FB                   JRZ     KEYPOL
E05D E65F                   ANI     5FH             ;UPPER AND LOWER
E05F 214CE0                 LXI     H,START
E062 E5                     PUSH    H
E063 FE04                   CPI     'D'-64
E065 CC7BE3                 CZ      VIDEO           ;ECHO CLEARSCN
E068 FE41                   CPI     'A'
E06A D8                     RC                      ;TOO SMALL
E06B FE5B                   CPI     05BH
E06D D0                     RNC                     ;TOO LARGE
E06E 21F9E0                 LXI     H,CMDTB+7EH
E071 F5                     PUSH    PSW
E072 87                     ADD     A
E073 85                     ADD     L
E074 6F                     MOV     L,A
E075 5E                     MOV     E,M
E076 23                     INX     H
```

```
E077 56                  MOV     D,M
E078 EB                  XCHG
E079 F1                  POP     PSW
E07A E9                  PCHL                    ;AWAY WE GO
E07B             * COMMAND TABLE
E07B 37E5        CMDTB   DW      WASCII          ;A
E07D 0CE8                DW      FLBOOT          ;B
E07F E2E2                DW      COMPR           ;C
E081 BBE5                DW      HEXRUL          ;D
E083 D0E7                DW      EXTCOM          ;E
E085 05E3                DW      FIND            ;F
E087 AFE0                DW      EXEC            ;G
E089 56E2                DW      RAM             ;H
E08B 53E3                DW      PINPT           ;I
E08D 96E1                DW      WARM            ;J
E08F B5E7                DW      SETBRK          ;K
E091 62E2                DW      LORAM           ;L
E093 96E2                DW      MOVEB           ;M
E095 BEE2                DW      NDMT            ;N
E097 65E3                DW      POUTP           ;O
E099 08E6                DW      PROGRAM         ;P
E09B 79E1                DW      CHKSM           ;Q
E09D BFE6                DW      DREGS           ;R
E09F 12E3                DW      SRCH            ;S
E0A1 C3E1                DW      TMEM            ;T
E0A3 47E2                DW      USER            ;U
E0A5 00E8                DW      DSBOOT          ;V
E0A7 02E8                DW      MSBOOT          ;W
E0A9 87E2                DW      EXCHG           ;X
E0AB AEE1                DW      ECHO            ;Y
E0AD 6EE2                DW      ZEROM           ;Z
E0AF             *
E0AF             *** EXECUTE THE PROGRAM AT THE ADDRESS ***
E0AF             *
E0AF CDC4E4      EXEC    CALL    PTSTNG
E0B2 474F2054            DTH     'GO TO '
E0B6 4FA0
E0B8 CDBDE0              CALL    AHEX            ;READ ADD FROM KB
E0BB EB                  XCHG
E0BC E9                  PCHL                    ;JUMP TO IT
E0BD             *
E0BD             *** CONVERT UP TO 4 HEX DIGITS TO BIN
E0BD             *
E0BD 0E04        AHEX    MVI     C,4             ;COUNT OF 4 DIGITS
E0BF 210000      AHE0    LXI     H,0             ;16 BIT ZERO
E0C2 CD2FE1      AHE1    CALL    ESCAPE
E0C5 FE20                CPI     ' '             ;SPACE?
E0C7 CAE8E0              JZ      SPCOVR
E0CA CDEDE0              CALL    HEX             ;CHECK VALUE
E0CD 38F3                JRC     AHE1
E0CF 29                  DAD     H               ;MULT H*16
E0D0 29                  DAD     H
E0D1 29                  DAD     H
E0D2 29                  DAD     H
E0D3 85                  ADD     L
E0D4 6F                  MOV     L,A
E0D5 0D                  DCR     C               ;4 DIGITS?
```



```
E0D6 C2C2E0              JNZ     AHE1            ;KEEP READING
E0D9 EB                  XCHG
E0DA 3E20        SPCE    MVI     A,20H           ;PRINT SPACE
E0DC C37BE3      PTCN    JMP     VIDEO
E0DF 3E0D        CRLF    MVI     A,0DH           ;PRINT CR
E0E1 CDDCE0              CALL    PTCN
E0E4 3E0A                MVI     A,0AH
E0E6 18F4                JR      PTCN
E0E8             *
E0E8 CD7BE3      SPCOVR  CALL    VIDEO
E0EB 18EC                JR      SPCE-1
E0ED             *
E0ED             * CHECK FOR HEX VALUE, CONVERT
E0ED FE30        HEX     CPI     30H             ;<0
E0EF D8                  RC
E0F0 FE3A                CPI     ':'             ;>9
E0F2 3809                JRC     NUM
E0F4 E65F                ANI     5FH             ;UPPER & LOWER CASE
E0F6 FE41                CPI     'A'             ;<A
E0F8 D8                  RC
E0F9 FE47                CPI     'G'             ;>F
E0FB 3F                  CMC
E0FC D8                  RC
E0FD CD7BE3      NUM     CALL    VIDEO
E100 D630                SUI     48              ;ASCII BIAS
E102 FE0A                CPI     10              ;DIGIT 0-10
E104 3802                JRC     ALFA
E106 D607                SUI     7               ;ALPHA BIAS
E108 A7          ALFA    ANA     A               ;CLEAR CY
E109 C9                  RET                     ;WITH CY CLEAR
E10A             *
E10A             * READ 2 DIGITS FROM THE CONSOLE
E10A 0E02        AHE2    MVI     C,2
E10C 18B1                JR      AHE0
E10E             *
E10E             * SHORT ROUTINE TO SAVE CODE
E10E CDBDE0      TAHEX   CALL    AHEX
E111 18AA                JR      AHEX
E113             *
E113             *** READ FROM CONSOLE TO REG A ***
E113             *
E113 CD2FE1      RDCN    CALL    ESCAPE          ;READ KEYBOARD
E116 28FB                JRZ     RDCN
E118 FE60                CPI     60H
E11A 38C0                JRC     PTCN
E11C E65F                ANI     5FH
E11E 18BC                JR      PTCN
E120             *
E120 CD2FE1      PAUSE   CALL    ESCAPE
E123 FE20                CPI     20H
E125 C0                  RNZ
E126 CD2FE1      PLOOP   CALL    ESCAPE
E129 FE20                CPI     20H
E12B C226E1              JNZ     PLOOP
E12E C9                  RET
E12F             *
E12F CD3CE1      ESCAPE  CALL    KEYSTAT
```

```
E132 C8                         RZ
E133 CD41E1                     CALL    CONVERT
E136 FE1B                       CPI     1BH             ;ESCAPE
E138 CA4CE0                     JZ      START
E13B C9                         RET
E13C            *
E13C DB00       KEYSTAT         IN      CONS
E13E E640                       ANI     RDA
E140 C9                         RET
E141            *
E141            * KEYBOARD CODE CONVERSION
E141 DB01       CONVERT         IN      COND            ;KEYBOARD DATA
E143 E5                         PUSH    H
E144 C5                         PUSH    B
E145 010500                     LXI     B,TABLEND-KTABL/2
E148 215BE1                     LXI     H,KTABL
E14B EDA1       LOOP            CCI                     ;COMPARE TABLE
E14D 2806                       JRZ     FND
E14F 23                         INX     H
E150 EA4BE1                     JPE     LOOP            ;CONT LOOKING
E153 1801                       JR      NFND
E155 7E         FND             MOV     A,M             ;NEW CODE
E156 E67F       NFND            ANI     7FH             ;MASK DOWN
E158 C1                         POP     B
E159 E1                         POP     H
E15A C9                         RET
E15B            *
E15B            * THIS TABLE CAN BE EXTENDED IF DESIRED
E15B E15D       KTABL           DD      0E15DH          ;]
E15D F15B                       DD      0F15BH          ;[
E15F A17E                       DD      0A17EH          ;~
E161 B15C                       DD      0B15CH          ;•
E163 6015                       DD      06015H          ;CURSOR UP
E165   E165 =    TABLEND         EQU     $
E165                            ORG     KTABL+30        ;ROOM FOR 15 CONVS
E179            *
E179            * CHECKSUM ROUTINE
E179 CDC4E4     CHKSM           CALL    PTSTNG
E17C 43484543                   DTH     'CHECKSUM '
E180 4B53554D
E184 A0
E185 CD0EE1                     CALL    TAHEX
E188 0600                       MVI     B,0
E18A 7E         CHKSMLP         MOV     A,M
E18B 80                         ADD     B
E18C 47                         MOV     B,A
E18D CD3FE2                     CALL    BMP
E190 20F8                       JRNZ    CHKSMLP
E192 78                         MOV     A,B
E193 C326E2                     JMP     PT2
E196            *
E196            * WARM START
E196            *
E196 CDC4E4     WARM            CALL    PTSTNG
E199 4A554D50                   DTH     'JUMP TO DOS'
E19D 20544F20
E1A1 444FD3
```



```
E1A4 21E704                     LXI     H,04E7H         ;MDOS RESTART
E1A7 7E                         MOV     A,M
E1A8 FEC3                       CPI     0C3H
E1AA C20000                     JNZ     0               ;CP/M RESTART
E1AD E9                         PCHL                    ;MDOS WARM START
E1AE            *
E1AE            * KEYBOARD ECHO ROUTINE
E1AE CDC4E4     ECHO            CALL    PTSTNG
E1B1 4543484F                   DTH     'ECHO KEYS '
E1B5 204B4559
E1B9 53A0
E1BB CD2FE1     ECOLP           CALL    ESCAPE          ;LOOK AT KEYBOARD
E1BE C4DCE0                     CNZ     PTCN            ;PRINT IF KEYPRESS
E1C1 18F8                       JR      ECOLP           ;CONTINUE LOOPING
E1C3            *
E1C3            *** MEMORY TEST ROUTINE ***
E1C3            *
E1C3 CDC4E4     TMEM            CALL    PTSTNG
E1C6 54455354                   DTH     'TEST '
E1CA A0
E1CB CD0EE1                     CALL    TAHEX           ;READ ADDRESSES
E1CE 015A5A                     LXI     B,5A5AH         ;INI B,C
E1D1 CDFDE1     CYCL            CALL    RNDM
E1D4 C5 •                       PUSH    B               ;KEEP ALL REGS
E1D5 E5                         PUSH    H
E1D6 D5                         PUSH    D
E1D7 CDFDE1     TLOP            CALL    RNDM
E1DA 70                         MOV     M,B             ;WRITE IN MEM
E1DB CD3FE2                     CALL    BMP
E1DE C2D7E1                     JNZ     TLOP            ;REPEAT LOOP
E1E1 D1                         POP     D
E1E2 E1                         POP     H               ;RESTORE ORIG
E1E3 C1                         POP     B               ;VALUES OF
E1E4 E5                         PUSH    H
E1E5 D5                         PUSH    D
E1E6 CDFDE1     RLOP            CALL    RNDM            ;GEN NEW SEQ
E1E9 7E                         MOV     A,M             ;READ MEM
E1EA B8                         CMP     B               ;COMP MEM
E1EB C41DE2                     CNZ     ERR             ;CALL ERROR RTN
E1EE CD3FE2                     CALL    BMP
E1F1 C2E6E1                     JNZ     RLOP
E1F4 D1                         POP     D
E1F5 E1                         POP     H
E1F6 3E2E                       MVI     A,'.'
E1F8 CD7BE3                     CALL    VIDEO
E1FB 18D4                       JR      CYCL
E1FD            *** THIS ROUTINE GENERATES RANDOM NOS ***
E1FD CD20E1     RNDM            CALL    PAUSE
E200 78                         MOV     A,B             ;LOOK AT B
E201 E6B4                       ANI     0B4H            ;MASK BITS
E203 A7                         ANA     A               ;CLEAR CY
E204 EA08E2                     JPE     PEVE            ;JUMP IF EVEN
E207 37                         STC
E208 79         PEVE            MOV     A,C             ;LOOK AT C
E209 17                         RAL                     ;ROTATE CY IN
E20A 4F                         MOV     C,A             ;RESTORE C
E20B 78                         MOV     A,B             ;LOOK AT B
```

```
E20C 17                         RAL             ;ROTATE CY IN
E20D 47                         MOV     B,A     ;RESTORE B
E20E C9                         RET             ;RETURN W NEW B,C
E20F                    *
E20F                    *** ERROR PRINT OUT ROUTINE
E20F                    *
E20F CDDFE0             PTAD    CALL    CRLF    ;PRINT CR,LF
E212 CD20E1                     CALL    PAUSE
E215 7C                         MOV     A,H     ;PRINT
E216 CD26E2                     CALL    PT2     ;ASCII
E219 7D                         MOV     A,L     ;CODES
E21A C31FE7                     JMP     PT2S    ;FOR ADDRESS
E21D                    *
E21D F5                 ERR     PUSH    PSW     ;SAVE ACC
E21E CD0FE2                     CALL    PTAD    ;PRINT ADD.
E221 78                         MOV     A,B     ;DATA
E222 CD1FE7                     CALL    PT2S    ;WRITTEN
E225 F1                         POP     PSW     ;DATA READ
E226 F5                 PT2     PUSH    PSW
E227 CD20E2                     CALL    BINH
E22A F1                         POP     PSW
E22B 1804                       JR      BINL
E22D 1F                 BINH    RAR             ;SHIFT RHT 4 BITS
E22E 1F                         RAR
E22F 1F                         RAR
E230 1F                         RAR
E231 E60F               BINL    ANI     0FH     ;LOW 4 BITS
E233 C630                       ADI     48      ;ASCII BIAS
E235 FE3A                       CPI     58      ;DIGIT 0-9
E237 DADCE0                     JC      PTCN
E23A C607                       ADI     7       ;DIGIT A-F
E23C C3DCE0                     JMP     PTCN
E23F                    *
E23F                    * COMPARE ADDRESSES AND INCREMENT H
E23F 7B                 BMP     MOV     A,E
E240 95                         SUB     L
E241 2002                       JRNZ    GOON
E243 7A                         MOV     A,D
E244 9C                         SBB     H
E245 23                 GOON    INX     H
E246 C9                         RET
E247                    *
E247                    * JUMP TO USER RAM
E247 CDC4E4             USER    CALL    PTSTNG
E24A 55534552                   DTH     'USER AREA'
E24E 20415245
E252 C1
E253 C30001                     JMP     0100H
E256                    *
E256                    * JUMP TO RAM AT PR+1C00
E256 CDC4E4             RAM     CALL    PTSTNG
E259 48492052                   DTH     'HI RAM'
E25D 41CD
E25F C300FC                     JMP     PR+1C00H
E262                    *
E262                    * JUMP TO RAM AT 0
E262 CDC4E4             LORAM   CALL    PTSTNG
```



```
E265 4C4F2052                   DTH     'LO RAM'
E269 41CD
E26B C30000                     JMP     0
E26E                    *
E26E                    * ZERO OR FILL MEMORY WITH A CONSTANT
E26E CDC4E4             ZEROM   CALL    PTSTNG
E271 46494C4C                   DTH     'FILL '
E275 A0
E276 CD0EE1                     CALL    TAHEX   ;READ ADDRESSES
E279 E5                         PUSH    H       ;SAVE H
E27A CD0AE1                     CALL    AHE2    ;READ 2 DIGITS
E27D EB                         XCHG
E27E E3                         XTHL            ;RESTORE H,L
E27F C1                         POP     B
E280 71                 ZLOOP   MOV     M,C     ;WRITE INTO MEM
E281 CD3FE2                     CALL    BMP     ;COMP ADD, INCR H
E284 C8                         RZ              ;RETURN IF DONE
E285 18F9                       JR      ZLOOP   ;CONTINUE TIL DONE
E287                    * EXCHANGE OR MOVE A BLOCK OF MEMORY
E287 47                 EXCHG   MOV     B,A
E288 CDC4E4                     CALL    PTSTNG
E28B 45584348                   DTH     'EXCHANGE '
E28F 414E4745
E293 A0
E294 1809                       JR      MOVENTR
E296 47                 MOVEB   MOV     B,A     ;SAVE CODE
E297 CDC4E4                     CALL    PTSTNG
E29A 4D4F5645                   DTH     'MOVE '
E29E A0
E29F CD0EE1             MOVENTR CALL    TAHEX   ;READ ADDRESSES
E2A2 E5                         PUSH    H
E2A3 CDBDE0                     CALL    AHEX
E2A6 EB                         XCHG
E2A7 E3                         XTHL            ;BACK TO NORMAL
E2A8 4E                 MLOOP   MOV     C,M
E2A9 E3                         XTHL
E2AA 78                         MOV     A,B
E2AB FE4D                       CPI     'M'
E2AD 2804                       JRZ     NEXCH
E2AF 7E                         MOV     A,M
E2B0 E3                         XTHL
E2B1 77                         MOV     M,A
E2B2 E3                         XTHL
E2B3 71                 NEXCH   MOV     M,C
E2B4 23                         INX     H
E2B5 E3                         XTHL
E2B6 CD3FE2                     CALL    BMP
E2B9 CA4CE0                     JZ      START
E2BC 18EA                       JR      MLOOP
E2BE                    * NON DESTRUCTIVE MEMORY TEST
E2BE CDC4E4             NDMT    CALL    PTSTNG
E2C1 4D454D20                   DTH     'MEM CHECK'
E2C5 43484543
E2C9 CB
E2CA 210000                     LXI     H,0     ;START AT ZERO
E2CD 4E                 NDLOP   MOV     C,M
E2CE 06FF                       MVI     B,0FFH
```

```
E2D0 70                     MOV     M,B
E2D1 7E                     MOV     A,M
E2D2 B8                     CMP     B
E2D3 C2DBE2                 JNZ     ERRJP           ;PRINT ERROR
E2D6 0600                   MVI     B,0
E2D8 70                     MOV     M,B
E2D9 7E                     MOV     A,M
E2DA B8                     CMP     B
E2DB C21DE2         ERRJP   JNZ     ERR
E2DE 71                     MOV     M,C
E2DF 23                     INX     H
E2E0 18EB                   JR      NDLOP
E2E2                * COMPARE TWO BLOCKS OF MEMORY
E2E2 CDC4E4         COMPR   CALL    PTSTNG
E2E5 434F4D50               DTH     'COMPARE '
E2E9 415245A0
E2ED CD0EE1                 CALL    TAHEX
E2F0 E5                     PUSH    H
E2F1 CDBDE0                 CALL    AHEX
E2F4 EB                     XCHG
E2F5 7E             VMLOP   MOV     A,M
E2F6 23                     INX     H
E2F7 E3                     XTHL
E2F8 BE                     CMP     M
E2F9 46                     MOV     B,M
E2FA C41DE2                 CNZ     ERR
E2FD CD3FE2                 CALL    BMP
E300 E3                     XTHL
E301 20F2                   JRNZ    VMLOP
E303 F1                     POP     PSW
E304 C9                     RET
E305                * SEARCH FOR SPECIFIC CODES
E305 F5             FIND    PUSH    PSW
E306 CDC4E4                 CALL    PTSTNG
E309 46494E44               DTH     'FIND-2 '
E30D 2D32A0
E310 180D                   JR      SRCHENT
E312 F5             SRCH    PUSH    PSW
E313 CDC4E4                 CALL    PTSTNG
E316 53454152               DTH     'SEARCH-1 '
E31A 43482D31
E31E A0
E31F CD0EE1         SRCHENT CALL    TAHEX
E322 E5                     PUSH    H               ;SAVE H
E323 CD0AE1                 CALL    AHE2            ;READ 2 DIGITS
E326 EB                     XCHG                    ;H=CODE,D=F
E327 45                     MOV     B,L             ;PUT CODE IN B
E328 E1                     POP     H               ;RESTORE H
E329 F1                     POP     PSW
E32A FE53                   CPI     'S'
E32C F5                     PUSH    PSW
E32D 2807                   JRZ     CONT
E32F E5                     PUSH    H
E330 CD0AE1                 CALL    AHE2            ;READ 2 DIGITS
E333 EB                     XCHG
E334 4D                     MOV     C,L
E335 E1                     POP     H
```



```
E336 7E             CONT    MOV     A,M             ;READ MEMORY
E337 B8                     CMP     B               ;COMPARE TO CODE
E338 2012                   JRNZ    SKP             ;SKIP IF NO COMP
E33A F1                     POP     PSW             ;FETCH CONTROL
E33B FE53                   CPI     'S'
E33D F5                     PUSH    PSW
E33E 2806                   JRZ     OBCP
E340 23                     INX     H
E341 7E                     MOV     A,M
E342 2B                     DCX     H
E343 B9                     CMP     C
E344 2006                   JRNZ    SKP
E346 23             OBCP    INX     H
E347 7E                     MOV     A,M             ;READ NEXT BYTE
E348 2B                     DCX     H               ;DECR ADDRESS
E349 CD1DE2                 CALL    ERR             ;PRINT CODES
E34C CD3FE2         SKP     CALL    BMP             ;CHECK IF DONE
E34F 20E5                   JRNZ    CONT            ;BACK FOR MORE
E351 F1                     POP     PSW
E352 C9                     RET
E353                *
E353                * INPUT DATA FROM A PORT
E353 CDC4E4         PINPT   CALL    PTSTNG
E356 494E5055               DTH     'INPUT '
E35A 54A0
E35C CD0AE1                 CALL    AHE2            ;READ 2 DIGITS
E35F 4B                     MOV     C,E
E360 ED78                   INP     A
E362 C326E2                 JMP     PT2
E365                *
E365                * OUTPUT TO A PORT
E365 CDC4E4         POUTP   CALL    PTSTNG
E368 4F555450               DTH     'OUTPUT '
E36C 5554A0
E36F CD0AE1                 CALL    AHE2            ;READ 2 DIGITS
E372 CD0AE1                 CALL    AHE2            ;READ 2 DIGITS
E375 4D                     MOV     C,L
E376 ED59                   OUTP    E
E378 C9                     RET
E379                *
```

```
E379                    *
E379                    ********************************************
E379                    *                                          *
E379                    *     VIDEO DRIVER FOR  FLASHWRITER II     *
E379                    *                                          *
E379                    ********************************************
E379                    *
E379    F000 =          PAGE            EQU     PR+1000H        ;SCREEN LOCATION
E379    0020 =          SPACE           EQU     20H
E379    0004 =          CLRSCRN         EQU     4
E379                    ********************************************
E379                    *                                          *
E379                    * CONTROL CODE COMMANDS:                   *
E379                    *   (B) HOME CURSOR                        *
E379                    *   (D) CLEAR SCREEN                       *
E379                    *   (E) PRINT CONTROL CODE                 *
E379                    *   (H) BACKSPACE                          *
E379                    *   (I) TAB                                *
E379                    *   (J) LINEFEED                           *
E379                    *   (M) CARRIAGE RETURN                    *
E379                    *   (N) NO CURSOR                          *
E379                    *   (P) CLEAR TO END OF SCREEN             *
E379                    *   (Q) CLEAR TO END OF LINE               *
E379                    *   (R) CURSOR DOWN                        *
E379                    *   (T) TOGGLE REVERSE VIDEO               *
E379                    *   (U) CURSOR UP                          *
E379                    *   (W) CURSOR LEFT                        *
E379                    *   (X) CLEAR TO START OF LINE             *
E379                    *   (Z) CURSOR RIGHT                       *
E379                    *   ESC XY POSITION LEAD-IN                *
E379                    *                                          *
E379                    ********************************************
E379                    *
E379                    * VIDEO BOARD PARAMETERS
E379    0050 =          HORIZ           EQU     80              ;NO. OF CHARACTERS
E379    0018 =          VERT            EQU     24              ;NO. OF LINES
E379                    *
E379 3E14               TVIDEO          MVI     A,'T'-64        ;TOGGLE VIDEO
E37B                    *
E37B F5                 VIDEO           PUSH    PSW
E37C C5                                 PUSH    B
E37D D5                                 PUSH    D
E37E E5                                 PUSH    H
E37F E67F                               ANI     07FH
E381 4F                                 MOV     C,A
E382 3A00E8                             LDA     BASE+800H
E385 FEC3                               CPI     0C3H            ;PROM THERE?
E387 79                                 MOV     A,C
E388 CC00E8                             CZ      BASE+800H       ;CALL IT IF SO
E38B CD60E4             DISPL           CALL    LIFTCURS        ;ERASE CURSOR
E38E 3AEAFF                             LDA     XYFLAG
E391 A7                                 ANA     A
E392 280A                               JRZ     NOKY
E394 3D                                 DCR     A
E395 32EAFF                             STA     XYFLAG
E398 CAAFE4                             JZ      YPOS
E39B C3A6E4                             JMP     XPOS
```



```
E39E 79                 NOKY            MOV     A,C             ;RECOVER CHARACTER
E39F FE20                               CPI     SPACE           ;PRINTING CODE?
E3A1 F2D5E3                             JP      PRINT
E3A4 FE1C                               CPI     PCL-TABL        ;TOO LARGE?
E3A6 F24284                             JP      RET
E3A9 E5                                 PUSH    H               ;CURSOR IN MEMORY
E3AA 21B8E3                             LXI     H,TABL          ;TABLE START
E3AD 5F                                 MOV     E,A
E3AE 1600                               MVI     D,0
E3B0 19                                 DAD     D
E3B1 5E                                 MOV     E,M
E3B2 21D4E3                             LXI     H,PCL
E3B5 19                                 DAD     D
E3B6 E3                                 XTHL                    ;RECOVER H
E3B7 C9                                 RET                     ;EXECUTE ROUTINE
E3B8                    * CONTROL CHARACTER JUMP TABLE
E3B8 6E                 TABL            DB      RET-PCL         ;@
E3B9 6E                                 DB      RET-PCL         ;A
E3BA 63                                 DB      HOME-PCL        ;B HOME CURSOR
E3BB 6E                                 DB      RET-PCL         ;C
E3BC 60                                 DB      FORM-PCL        ;D CLEAR SCREEN
E3BD 00                                 DB      PCL-PCL         ;E PRT CONTROL
E3BE 6E                                 DB      RET-PCL         ;F
E3BF 6E                                 DB      RET-PCL         ;G
E3C0 42                                 DB      DBACKSP-PCL     ;H BACKSPACE
E3C1 59                                 DB      TAB-PCL         ;I TAB OVER
E3C2 12                                 DB      LINF-PCL        ;J LINE FEED
E3C3 6E                                 DB      RET-PCL         ;K
E3C4 6E                                 DB      RET-PCL         ;L
E3C5 6A                                 DB      CRET-PCL        ;M CARRIAGE RET
E3C6 71                                 DB      RET+3-PCL       ;N NO CURSOR
E3C7 6E                                 DB      RET-PCL         ;O
E3C8 A7                                 DB      CLEND-PCL       ;P CLR SCN TO END
E3C9 AC                                 DB      CLLINE-PCL      ;Q CLR LINE TO END
E3CA 12                                 DB      LINF-PCL        ;R CURSOR DOWN
E3CB 6E                                 DB      RET-PCL         ;S
E3CC 76                                 DB      TVIDF-PCL       ;T TOGGLE VIDEO
E3CD 80                                 DB      CURSUP-PCL      ;U CURSOR UP
E3CE 6E                                 DB      RET-PCL         ;V
E3CF 50                                 DB      BACKSP-PCL      ;W CURSOR LEFT
E3D0 E4                                 DB      CLSTRT-PCL      ;X CLR START OF LN
E3D1 6E                                 DB      RET-PCL         ;Y
E3D2 06                                 DB      EOL-PCL         ;Z CURSOR RIGHT
E3D3 CB                                 DB      LEDIN-PCL       ;[ ESC=XY LEADIN
E3D4                    *
E3D4                    * PRINT CODE IN B REGARDLESS
E3D4 48                 PCL             MOV     C,B
E3D5                    * PRINT THE CHARACTER ON THE SCREEN
E3D5 3ADDFF             PRINT           LDA     VFL
E3D8 A9                                 XRA     C
E3D9 77                                 MOV     M,A
E3DA                    * EOL CHECKS THE CURS POS FOR END OF LINE
E3DA 3ADBFF             EOL             LDA     CURPOS
E3DD 3C                                 INR     A
E3DE FE50                               CPI     HORIZ
E3E0 385D                               JRC     TABRET
E3E2 AF                                 XRA     A
```

```
E3E3 32DBFF                    STA     CURPOS
E3E6           * MOVE DN 1 LINE
E3E6 3ADCFF    LINF            LDA     LINENO
E3E9 FE17                      CPI     VERT-1
E3EB 2023                      JRNZ    NOSCRL
E3ED           * SCROLL UP ONE LINE
E3ED 215000    SCROLL          LXI     H,HORIZ
E3F0 ED5BDFFF                  LDED    TOSCN
E3F4 19                        DAD     D
E3F5 EDA0      SCRL            LDI
E3F7 EDA0                      LDI
E3F9 7C                        MOV     A,H
E3FA FEF7                      CPI     HORIZ*VERT+PAGE/256
E3FC 20F7                      JRNZ    SCRL
E3FE 7D                        MOV     A,L
E3FF FE80                      CPI     HORIZ*VERT+PAGE&0FFH
E401 20F2                      JRNZ    SCRL
E403 3ADCFF                    LDA     LINENO
E406           * ERASE BOTTOM LINE
E406 EB        EBOTL           XCHG
E407 0650                      MVI     B,HORIZ
E409 3620      ELOP            MVI     M,SPACE
E40B 23                        INX     H
E40C 05                        DCR     B
E40D 20FA                      JRNZ    ELOP
E40F 3D                        DCR     A
E410 3C        NOSCRL          INR     A
E411 32DCFF                    STA     LINENO
E414 182C                      JR      RET
E416           *
E416           * ERASE BEFORE BACKSPACING
E416 3620      DBACKSP         MVI     M,20H
E418 3ADBFF                    LDA     CURPOS
E41B A7                        ANA     A
E41C 2824                      JRZ     RET
E41E 3D                        DCR     A
E41F 2B                        DCX     H
E420 3620                      MVI     M,20H
E422 181B                      JR      TABRET
E424           * MOVE THE CURSOR BACK
E424 3ADBFF    BACKSP          LDA     CURPOS
E427 3D                        DCR     A
E428 F23FE4                    JP      TABRET
E42B 1811                      JR      CRET
E42D           * TAB OVER TO THE NEXT 8 MULTIPLE
E42D 3ADBFF    TAB             LDA     CURPOS
E430 F607                      ORI     7
E432 18A9                      JR      EOL+3
E434           * CLEAR THE SCREEN AND HOME UP
E434 CD8DE4    FORM            CALL    CLEAR
E437 AF        HOME            XRA     A
E438 32DCFF                    STA     LINENO
E43B 32DDFF                    STA     VFL             ;CLR VID FLAG
E43E           * CARRIAGE RETURN
E43E AF        CRET            XRA     A
E43F 32DBFF    TABRET          STA     CURPOS
E442           * RETURN TO THE CALLING ROUTINE
```



```
E442 CD60E4    RET             CALL    LIFTCURS
E445 E1                        POP     H
E446 D1                        POP     D
E447 C1                        POP     B
E448 F1                        POP     PSW
E449 C9                        RET
E44A 3ADDFF    TVIDF           LDA     VFL
E44D EE80                      XRI     80H
E44F 32DDFF                    STA     VFL
E452 18EE                      JR      RET
E454           *
E454           * MOVE THE CURSOR UP
E454 3ADCFF    CURSUP          LDA     LINENO
E457 A7                        ANA     A
E458 28E8                      JRZ     RET
E45A 3D                        DCR     A
E45B 32DCFF    STORLN          STA     LINENO
E45E 18E2                      JR      RET
E460           * CALCULATE MEM ADD FROM CURSOR POSITION
E460 2180F7    LIFTCURS        LXI     H,HORIZ*VERT+PAGE
E463 11B0FF                    LXI     D,-HORIZ
E466 3ADCFF                    LDA     LINENO
E469 3C        CLOP            INR     A
E46A 19                        DAD     D
E46B FE18                      CPI     VERT
E46D 20FA                      JRNZ    CLOP            ;OPTIMIZED AT BOTTOM
E46F ED5BDBFF  CFIN            LDED    CURPOS
E473 1600                      MVI     D,0
E475 19                        DAD     D
E476           * REVERSE THE VIDEO
E476 7E                        MOV     A,M
E477 EE80                      XRI     80H
E479 77                        MOV     M,A
E47A C9                        RET
E47B           * CLEAR TO END OF SCREEN
E47B CD96E4    CLEND           CALL    WRSPC
E47E 18C2                      JR      RET
E480           * CLEAR TO END OF LINE
E480 3ADBFF    CLLINE          LDA     CURPOS
E483 3620                      MVI     M,20H
E485 23                        INX     H
E486 3C                        INR     A
E487 FE50                      CPI     50H
E489 20F8                      JRNZ    CLLINE+3
E48B 18B5                      JR      RET
E48D           * CLEAR THE SCREEN
E48D 2100F0    CLEAR           LXI     H,PAGE
E490 22DFFF                    SHLD    TOSCN
E493 22EAFF                    SHLD    XYFLAG
E496 3620      WRSPC           MVI     M,20H
E498 23                        INX     H
E499 7C                        MOV     A,H
E49A FEF8                      CPI     PAGE+2048/256
E49C 20F8                      JRNZ    WRSPC
E49E C9                        RET
E49F           *
E49F           * PROCESS LEAD IN CODE
```

```
E49F 3E02      LEDIN    MVI   A,2
E4A1 32EAFF             STA   XYFLAG
E4A4 189C               JR    RET
E4A6           *
E4A6           * SET X AND Y CURSOR POSITIONS
E4A6 79        XPOS     MOV   A,C
E4A7 FE50               CPI   80
E4A9 3802               JRC   XINRG
E4AB 3E4F               MVI   A,79
E4AD 1890      XINRG    JR    TABRET
E4AF           *
E4AF 79        YPOS     MOV   A,C
E4B0 FE18               CPI   24
E4B2 3802               JRC   YINRG
E4B4 3E17               MVI   A,23
E4B6 18A3      YINRG    JR    STORLN
E4B8           *
E4B8 AF        CLSTRT   XRA   A
E4B9 32DBFF             STA   CURPOS
E4BC CD60E4             CALL  LIFTCURS
E4BF 18BF               JR    CLLINE
E4C1   E4C1 =  MSEND    EQU   $
E4C1           * CURSOR STORAGE LOCATIONS
E4C1                    ORG   SPTR+0BH
FFDB           CURPOS   DS    1          ;POS ON LINE
FFDC           LINENO   DS    1          ;LINE NUMBER
FFDD           VFL      DS    1          ;REVERSE VID FLAG
FFDE           WIDTH    DS    1          ;PRINT WIDTH
FFDF           TOSCN    DS    2          ;TOP OF SCREEN
FFE1           TCURPOS  DS    2          ;TEMP POSITION
FFE3                    LINK  'M5'
FFE3           * ADDITIONS TO 4.0 MONITOR
FFE3                    ORG   MSEND
E4C1           * PRINT A STRING
E4C1 CDDFE0    RPTSTNG  CALL  CRLF       ;CRLF FIRST
E4C4 E3        PTSTNG   XTHL
E4C5 7E                 MOV   A,M
E4C6 23                 INX   H
E4C7 E3                 XTHL
E4C8 A7                 ANA   A
E4C9 CD7BE3             CALL  VIDEO       ;PRINT IT
E4CC F8                 RM
E4CD 18F5               JR    PTSTNG
E4CF           * SIGN ON MESSAGE
E4CF 223900    SIGN     SHLD  39H         ;REMNANT FROM RST 7 PATCH
E4D2 3E04               MVI   A,4         ;CLEAR SCREEN
E4D4 CD7BE3             CALL  VIDEO
E4D7 2150F1             LXI   H,PAGE+150H
E4DA E5                 PUSH  H
E4DB 1151F1             LXI   D,PAGE+151H
E4DE 013000             LXI   B,30H
E4E1 3612               MVI   M,12H       ;GRAPHIC CHARACTER
E4E3 EDB0               LDIR
E4E5 E1                 POP   H
E4E6 11A0F1             LXI   D,PAGE+1A0H
E4E9 018002             LXI   B,640
E4EC EDB0               LDIR
```



```
E4EE CDC4E4             CALL  PTSTNG
E4F1 1B                 DB    27          ;ESC
E4F2 2007               DD    2007H       ;X=32 Y=7
E4F4 20564543           DT    ' VECTOR GRAPHIC '
E4F8 544F5220
E4FC 47524150
E500 48494320
E504 1B                 DB    27          ;ESC
E505 2008               DD    2008H       ;X=32 Y=8
E507 20202020           DT    '    MONITOR    '
E50B 4D4F4E49
E50F 544F5220
E513 20202020
E517 1B                 DB    27          ;ESC
E518 2009               DD    2009H       ;X=32 Y=9
E51A 20205645           DT    '  VERSION 4.3  '
E51E 5253494F
E522 4E20342E
E526 33202020
E52A 1B                 DB    27          ;ESC
E52B 008D               DD    8DH         ;X=0 Y=13
E52D C9                 RET
E52E CDC1E4    PROMPT   CALL  RPTSTNG
E531 4D6F6E3E           DTH   'Mon> '
E535 A0
E536 C9                 RET
E537           *
E537           *WIDE ASCII DUMP
E537 CDC4E4    WASCII   CALL  PTSTNG
E53A 41534349           DTH   'ASCII DUMP '
E53E 49204455
E542 4D50A0
E545 CD0EE1             CALL  TAHEX
E548 CD8BE5             CALL  HOMEC
E54B           * MAKE A RULER FOR ASCII DUMP
E54B 78        RULELP   MOV   A,B
E54C FE40               CPI   64
E54E 281A               JRZ   TERMLIN
E550 E60F               ANI   0FH
E552 2810               JRZ   NUMBER
E554 E603               ANI   3
E556 2808               JRZ   MARKER
E558 3E20               MVI   A,' '
E55A CD7BE3    REENTR   CALL  VIDEO
E55D 04                 INR   B
E55E 18EB               JR    RULELP
E560 3E6C      MARKER   MVI   A,'l'
E562 18F6               JR    REENTR
E564 78        NUMBER   MOV   A,B
E565 CD2DE2             CALL  BINH
E568 18F3               JR    REENTR+3
E56A           * TOGGLE REVERSE VIDEO
E56A CD79E3    TERMLIN  CALL  TVIDEO
E56D CDF7E5    WDMP1    CALL  SETSCRLL
E570 CD0FE2             CALL  PTAD
E573 0E3F               MVI   C,63
E575 CD7CE5             CALL  WDMP2
```

```
E578 FA6DE5                    JM      WDMP1
E57B C8                        RZ
E57C 7E         WDMP2          MOV     A,M
E57D 47                        MOV     B,A
E57E 3E05                      MVI     A,'E'-64
E580 CD7BE3                    CALL    VIDEO
E583 CD3FE2                    CALL    BMP
E586 C8                        RZ
E587 0D                        DCR     C
E588 F8                        RM
E589 18F1                      JR      WDMP2
E58B            * HOME CURSOR, PRINT "ADDR"
E58B CDC1E4     HOMEC          CALL    RPTSTNG
E58E 14                        DB      'T'-64
E58F 41444452                  DTH     'ADDR '
E593 A0
E594 0600                      MVI     B,0
E596 3E18                      MVI     A,24
E598 32DEFF                    STA     WIDTH
E59B C9                        RET
E59C            * MAKE A RULER FOR HEX DUMP
E59C 78         HEXRULER       MOV     A,B
E59D FE10                      CPI     16
E59F 2806                      JRZ     HEXRCT
E5A1 CD1FE7                    CALL    PT2S
E5A4 04                        INR     B
E5A5 18F5                      JR      HEXRULER
E5A7            * EXTEND FOR ASCII
E5A7 CDDAE0     HEXRCT         CALL    SPCE
E5AA CDDAE0                    CALL    SPCE
E5AD 0600                      MVI     B,0
E5AF 78         HEXRLP         MOV     A,B
E5B0 FE10                      CPI     16
E5B2 C8                        RZ
E5B3 E60F                      ANI     0FH
E5B5 CD31E2                    CALL    SINL
E5B8 04                        INR     B
E5B9 18F4                      JR      HEXRLP
E5BB            * HEX DUMP ROUTINE
E5BB CDC4E4     HEXRUL         CALL    PTSTNG
E5BE 48455820                  DTH     'HEX DUMP '
E5C2 44554D50
E5C6 A0
E5C7 CD0EE1                    CALL    TAHEX
E5CA CD8BE5                    CALL    HOMEC
E5CD CD9CE5                    CALL    HEXRULER
E5D0 CD79E3                    CALL    TVIDEO
E5D3 CDF7E5                    CALL    SETSCRLL
E5D6 CD0FE2     HLP1           CALL    PTAD
E5D9 E5                        PUSH    H
E5DA D5                        PUSH    D
E5DB 0E10                      MVI     C,16
E5DD 7E         HLP2           MOV     A,M
E5DE CD1FE7                    CALL    PT2S
E5E1 23                        INX     H
E5E2 0D                        DCR     C
E5E3 C2DDE5                    JNZ     HLP2
```



```
E5E6 D1                        POP     D
E5E7 E1                        POP     H
E5E8 0E0F                      MVI     C,15
E5EA CDDAE0                    CALL    SPCE
E5ED CDDAE0                    CALL    SPCE
E5F0 CD7CE5                    CALL    WDMP2
E5F3 FAD3E5                    JM      HLP1-3
E5F6 C9                        RET
E5F7            * CHECK TO SET SCROLL POINT
E5F7 3ADEFF     SETSCRLL       LDA     WIDTH
E5FA 3D                        DCR     A
E5FB 32DEFF                    STA     WIDTH
E5FE 2007                      JRNZ    CTSCRL
E600 0150F0                    LXI     B,PAGE+50H      ;2ND LINE
E603 ED43DFFF                  SBCD    TOSCN           ;SCROLL POINT
E607 C9         CTSCRL         RET
E608            *
E608            * PROGRAM MEMORY
E608 CDC4E4     PROGRAM        CALL    PTSTNG
E60B 50524F47                  DTH     'PROGRAM '
E60F 52414DA0
E613 CDBDE0                    CALL    AHEX            ;ADDR IN HL
E616 ED53E1FF                  SDED    TCURPOS
E61A CD8BE5                    CALL    HOMEC           ;PRINT "ADDR"
E61D CD9CE5                    CALL    HEXRULER
E620 CD79E3                    CALL    TVIDEO
E623 AF                        XRA     A
E624 32DEFF                    STA     WIDTH
E627 CD91E6                    CALL    PRT1LINE        ;PRINT LINE CONT H
E62A CD2FE1     POLLOOP        CALL    ESCAPE
E62D CDEDE0                    CALL    HEX
E630 2AE1FF                    LHLD    TCURPOS
E633 301A                      JRNC    MODMEM
E635            * CONTROL CODE TABLE
E635 FE20                      CPI     ' '
E637 2846                      JRZ     CSRT
E639 FE08                      CPI     8
E63B 2845                      JRZ     CSLT
E63D FE12                      CPI     'R'-64
E63F 2839                      JRZ     CSDN
E641 FE15                      CPI     'U'-64
E643 282F                      JRZ     CSUP
E645 FE17                      CPI     'W'-64
E647 2839                      JRZ     CSLT
E649 FE1A                      CPI     'Z'-64
E64B 2832                      JRZ     CSRT
E64D 18DB                      JR      POLLOOP
E64F            * MODIFY A MEMORY LOCATION
E64F 2AE1FF     MODMEM         LHLD    TCURPOS
E652 4F                        MOV     C,A
E653 3ADEFF                    LDA     WIDTH
E656 A7                        ANA     A
E657 7E                        MOV     A,M
E658 280D                      JRZ     LSNIBL
E65A E6F0                      ANI     0F0H
E65C B1                        ORA     C
E65D 77         REMEM          MOV     M,A
```

```
E65E 3ADEFF                LDA     WIDTH
E661 EE01                  XRI     1
E663 201F                  JRNZ    RTRIN+1
E665 1818                  JR      CSRT
E667 17          LSNIBL    RAL
E668 17                    RAL
E669 17                    RAL
E66A 17                    RAL
E66B E6F0                  ANI     0F0H
E66D B1                    ORA     C
E66E 0F                    RRC
E66F 0F                    RRC
E670 0F                    RRC
E671 0F                    RRC
E672 18E9                  JR      REMEM
E674             * MOVE UP ONE LINE
E674 11F0FF      CSUP      LXI     D,-16
E677 19                    DAD     D
E678 1809                  JR      RTRIN
E67A             * MOVE DOWN ONE LINE
E67A 111000      CSDN      LXI     D,16
E67D 18F8                  JR      CSUP+3
E67F             * MOVE RIGHT ONE SPACE
E67F 23          CSRT      INX     H
E680 1801                  JR      RTRIN
E682             * MOVE LEFT ONE SPACE
E682 2B          CSLT      DCX     H
E683             *
E683 AF          RTRIN     XRA     A
E684 32DEFF                STA     WIDTH
E687 22E1FF                SHLD    TCURPOS
E68A 3E15        UPAROW    MVI     A,'U'-64
E68C CD78E3                CALL    VIDEO
E68F 1896                  JR      POLLOOP-3
E691             * PRINT A LINE CONTAINING ((H))
E691 2AE1FF      PRT1LINE  LHLD    TCURPOS
E694 E5                    PUSH    H
E695 D1                    POP     D
E696 7D                    MOV     A,L
E697 F60F                  ORI     0FH
E699 5F                    MOV     E,A
E69A E6F0                  ANI     0F0H
E69C 6F                    MOV     L,A
E69D CDD6E5                CALL    HLP1
E6A0             * NOW PUT CURSOR WHERE IT GOES
E6A0 CD60E4                CALL    LIFTCURS
E6A3 2AE1FF                LHLD    TCURPOS
E6A6 7D                    MOV     A,L
E6A7 E60F                  ANI     0FH
E6A9 6F                    MOV     L,A
E6AA 3E05                  MVI     A,5
E6AC 2D          PLOP1     DCR     L
E6AD FAB4E6                JM      PGCONT
E6B0 C603                  ADI     3
E6B2 18F8                  JR      PLOP1
E6B4 6F          PGCONT    MOV     L,A
E6B5 3ADEFF                LDA     WIDTH
```



```
E6B8 85                    ADD     L
E6B9             * A = 5+3*L+W
E6B9 32DBFF                STA     CURPOS
E6BC C360E4                JMP     LIFTCURS
E6BF             *
E6BF             *
E6BF             * DISPLAY REGISTERS
E6BF CDC4E4      DREGS     CALL    PTSTNG
E6C2 52454749              DTH     'REGISTERS'
E6C6 53544552
E6CA D3
E6CB             * DUMP REGISTERS AFTER ENTRY FROM RST 7
E6CB E3          DUMPREGS  XTHL
E6CC F5                    PUSH    PSW
E6CD CD25E7                CALL    DISPREGS
E6D0 2B                    DCX     H               ;GET BREAK ADD
E6D1 CD0FE2                CALL    PTAD
E6D4 E1                    POP     H
E6D5 C5                    PUSH    B
E6D6 CD7AE7                CALL    PRTFLGS
E6D9 C1                    POP     B
E6DA CD12E2                CALL    PTAD+3          ;PRINT AF
E6DD E1                    POP     H
E6DE 22E3FF                SHLD    HLTEMP
E6E1 CD9BE7                CALL    PTHREE          ;PRINT B D H
E6E4 DDE5                  PUSH    IX
E6E6 E1                    POP     H
E6E7 CD12E2                CALL    PTAD+3          ;PRINT IX
E6EA FDE5                  PUSH    IY
E6EC E1                    POP     H
E6ED CD12E2                CALL    PTAD+3          ;PRINT IY
E6F0 210000                LXI     H,0
E6F3 39                    DAD     SP
E6F4 22E5FF                SHLD    SPTEMP
E6F7 CD12E2                CALL    PTAD+3          ;PRINT SP
E6FA 08                    EXAF
E6FB F5                    PUSH    PSW
E6FC E1                    POP     H
E6FD CD12E2                CALL    PTAD+3
E700 D9                    EXX
E701 CD9BE7                CALL    PTHREE
E704 D9                    EXX
E705 0A                    LDAX    B
E706 CD1FE7                CALL    PT2S
E709 1A                    LDAX    D
E70A CD1FE7                CALL    PT2S
E70D 2AE3FF                LHLD    HLTEMP
E710 7E                    MOV     A,M
E711 CD1FE7                CALL    PT2S
E714 2AE5FF                LHLD    SPTEMP
E717 F9                    SPHL
E718 E1                    POP     H
E719 CD12E2                CALL    PTAD+3
E71C C340E0                JMP     CLRBRK          ;CLEAR BREAKPOINT
E71F             *
E71F CD26E2      PT2S      CALL    PT2             ;PRINT 2 CHARS
E722 C3DAE0                JMP     SPCE            ;PRINT SPACE
```

```
E725                    * DISPLAY REGISTER HEADER ON SCREEN
E725 CDC1E4     DISPREGS        CALL    RPTSTNG
E728 14                         DB      'T'-64
E729 41444452                   DT      'ADDR FLAGS  AF   BC   DE'
E72D 20464C41
E731 47532020
E735 41462020
E739 20424320
E73D 20204445
E741 20202048                   DT      '   HL   IX   IY   SP '
E745 4C202020
E749 49582020
E74D 20495920
E751 20205350
E755 20
E756 20204146                   DT      '  AF'
E75A 27                         DB      27H             ;'
E75B 20204243                   DT      '  BC'
E75F 27                         DB      27H
E760 20204445                   DT      '  DE'
E764 27                         DB      27H
E765 2020484C                   DT      '  HL'
E769 27                         DB      27H
E76A 20404220                   DT      ' @B @D @H @SP '
E76E 40442040
E772 48204053
E776 5020
E778 94                         DB      'T'+64
E779 C9                         RET
E77A                    *
E77A                    * PRINT FLAGS
E77A 015A40     PRTFLGS         LXI     B,405AH         ;Z
E77D CDAAE7                     CALL    MASKFLG
E780 014301                     LXI     B,143H          ;C
E783 CDAAE7                     CALL    MASKFLG
E786 014D80                     LXI     B,804DH         ;M
E789 CDAAE7                     CALL    MASKFLG
E78C 014504                     LXI     B,445H          ;E
E78F CDAAE7                     CALL    MASKFLG
E792 014810                     LXI     B,1048H         ;H
E795 CDAAE7                     CALL    MASKFLG
E798 C3DAE0                     JMP     SPCE
E79B                    *
E79B                    * PRINT BC DE HL IN ORDER
E79B E5         PTHREE          PUSH    H
E79C C5                         PUSH    B
E79D E1                         POP     H
E79E CD12E2                     CALL    PTAD+3
E7A1 D5                         PUSH    D
E7A2 E1                         POP     H
E7A3 CD12E2                     CALL    PTAD+3
E7A6 E1                         POP     H
E7A7 C312E2                     JMP     PTAD+3
E7AA                    *
E7AA 7D         MASKFLG         MOV     A,L
E7AB A0                         ANA     B
E7AC 3E20                       MVI     A,20H
```



```
E7AE CA7BE3                     JZ      VIDEO
E7B1 79                         MOV     A,C
E7B2 C37BE3                     JMP     VIDEO
E7B5                    *
E7B5                    * SET BREAKPOINT
E7B5 CDC4E4     SETBRK          CALL    PTSTNG
E7B8 42524541                   DTH     'BREAK AT '
E7BC 4B204154
E7C0 A0
E7C1 CDBDE0                     CALL    AHEX
E7C4 1A                         LDAX    D
E7C5 32E9FF                     STA     BRKCODE
E7C8 ED53E7FF                   SDED    BKPTLOC
E7CC 3EFF                       MVI     A,0FFH          ;RST 7
E7CE 12                         STAX    D
E7CF C9                         RET
E7D0                    *
E7D0                    * EXTERNAL COMMUNICATIONS
E7D0 CDC4E4     EXTCOM          CALL    PTSTNG
E7D3 45585420                   DTH     'EXT COM '
E7D7 434F4DA0
E7DB DB05       RECEIVE         IN      5
E7DD E602                       ANI     2
E7DF 2805                       JRZ     NEXCHR
E7E1 DB04                       IN      4
E7E3 CD7BE3                     CALL    VIDEO
E7E6 CD2FE1     NEXCHR          CALL    ESCAPE
E7E9 28F0                       JRZ     RECEIVE
E7EB D304                       OUT     4
E7ED 18EC                       JR      RECEIVE
E7EF                    *
E7EF                    * TEMPORARY STORAGE LOCATIONS FOR REGISTERS,ETC.
E7EF                            ORG     TCURPOS+2
FFE3            HLTEMP          DS      2
FFE5            SPTEMP          DS      2
FFE7            BKPTLOC         DS      2               ;BREAKPT LOCATION
FFE9            BRKCODE         DS      1               ;CODE AT BREAKPT
FFEA            XYFLAG          DS      1               ;CURSOR XY FLAG
```