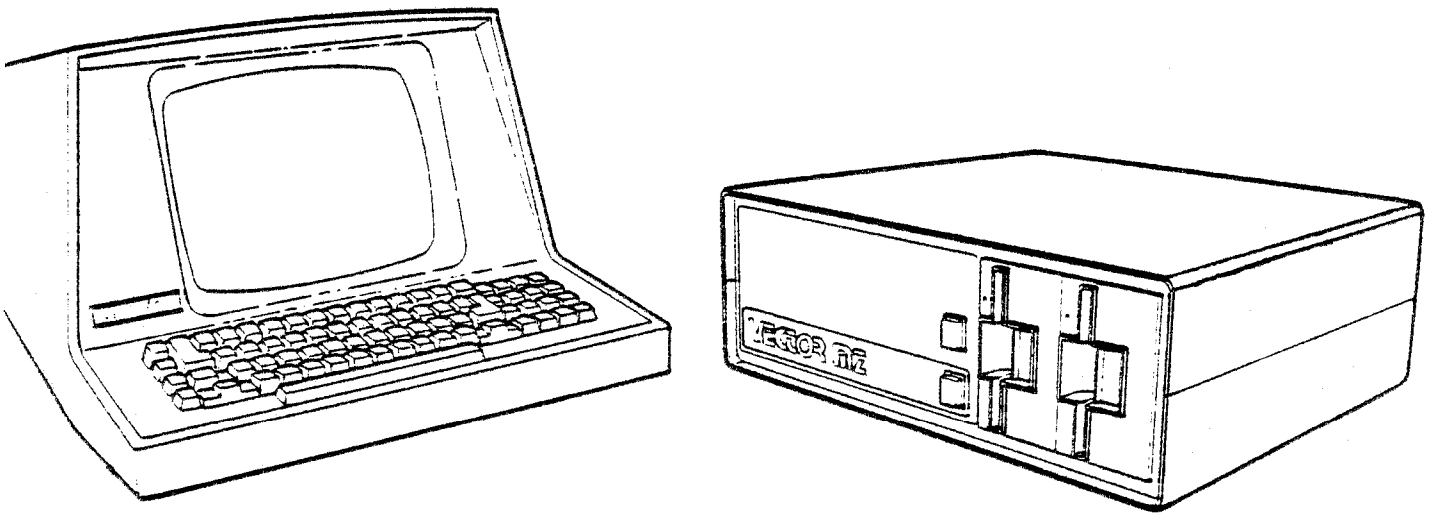
EXTENDED SYSTEMS MONITOR 4.1

USERS GUIDE



VECTOR
VECTOR GRAPHIC, INC.

EXTENDED SYSTEMS MONITOR

Version 4.1

USERS MANUAL

Revision A

SEPTEMBER 5, 1980

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

Section Page

General Description....1

Table of Hex values....2

Command Format

A - ASCII Dump....3
B - Jump to Bootstrap Loader-5-1/4" drives
C - Compare Blocks
D - Dump in Hex
E - External Communications

F - Find Two Bytes....4
G - Go to and Execute
H - Jump to HI Ram
I - Input from a Port
J - Jump to Loaded DOS
K - Set Breakpoints

L - Jump to 0000H....5
M - Move Memory Block
N - Non-desructive Memory Test
O - Output to Port
P - Program Memory

Q - Compute Checksum....6
R - Register Dump
S - Search for Single Byte
T - Test Memory

U - Jump to 2B00H....7
V - Jump to Bootstrap Loader-8" disk drives
W - Jump to Bootstrap Loader-8" Winchester
X - Exchange Memory Blocks
Y - Keyboard Echo
Z - Zero or Fill Memory

Entry Points....8

Video Driver....9

Cursor X Y Positioning....10

Keyboard Code Conversion for Vector Graphic Keyboards....10

Using the I/O Routines....11

Other Useful Monitor Routines....12

Monitor Listing....

GENERAL DESCRIPTION

The Version 4.1 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 4.1 was designed to be used with the Flashwriter II board. Use Version 4.0C for serial terminals.

Version 4.1 differs from 4.0 in the following key ways:

1) A new command has been added to jump directly to the bootstrap loader for Vector 8" floppy disk drives. (Executive command "V".)

2) A new command has been added to jump directly to the bootstrap loader for the Vector Winchester technology hard disk drive. (Executive command "W".)

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 ouput 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 softare, 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 4.1 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:

```
      10H          Hex Trivial)
40H) 400H
```

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 the lineditor, BASIC or MEMORITE. ASCII strings embedded in object code are easy to recognize.

Mon>B - JUMP TO BOOTSTRAP LOADER

Typing this command will cause a jump to location F800H which is the disk bootstrap loader. This will cause the disk operating system disk to be loaded into memory and transfer control to CP/M.

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" (C00BH) and if the user routine at ADR1 ends in "RET", program execution will return to the Monitor. Virtually unlimited stack space is available (up to 1K), 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 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 6000 through 63FF to be repeated 8 times from 6000 through 7FFF, since by the time location 6400 is read, it has been overwritten with data from 6000. This is useful for bank programming of proms, or for creating repeating instruciton sequences for test purposes.

Mon>N - NON-DESTRUCTIVE MEMORY TEST

Memory locations starting at 0000 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 4000. 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 instrucion "OUT PORT" is executed. This command is useful for checking our 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 preceeding and following instructions to be viewed. Advancing to the next instruction is accomplished by typing space or cursor right ( ). Backspace or cursor left ( ) 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 <ADRI1> <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 FFD0 on the 2708 PROM/RAM board.

Mon>U - JUMP TO 2B00

This command permits easy return to programs in the user application area of MDOS.

Mon>V - 8" DRIVE BOOT

Typing this command will cause a jump to E800H (contained on the Disk Boot #3 PROM) 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 the Disk Boot #3 PROM) 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 and the ZCB addressed starting at port 5. 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 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 control (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 (also)
20	14	(©T)	TOGGLE REVERSE VIDEO
21	15	(©U)	CURSOR UP (also)
23	17	(©W)	CURSOR LEFT (also)
24	18	(©X)	CLEAR TO START OF LINE
26	1A	(©Z)	CURSOR RIGHT (also)
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 folowing machine code at FC00H and execute it to demonstrate this feature:

```
at FC00  06 02 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. This may be done through assembly language or a higher level language such as Basic. 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). 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 either 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 th accumulator, or may require different flag conditions or accumulator content. 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 AHEX 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.

```
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 4.1      *
0000            *       R. S. HARP  7/16/79  MODIFIED 6/1/80   *
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            *
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 2B00                 *
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 C318E0     MONIT           JMP     INIT            ;INITIALIZE ALL
E003 C33CE1     KEYTST          JMP     KEYSTAT         ;TEST KEYBOARD
E006 C341E1     KEYDATA         JMP     CONVERT         ;INPUT KEYBOARD
E009 C38AE3     CRT             JMP     VIDEO           ;OUTPUT TO SCREEN
E00C C32FE1     ESC             JMP     ESCAPE          ;KEYBOARD INPUT
E00F 00                         NOP
E010 00                         NOP
```



```
E011 00                         NOP
E012            *
E012            * TABLE OF COMMANDS FOR USART
E012 00000040   INITABLE        DB      0,0,0,40H,0CEH,27H
E016 CE27
E018            *
E018 31D0FF     INIT            LXI     SP,SPTR         ;INIT STACK
E01B CD2FE1                     CALL    ESCAPE          ;DUMP LATCH
E01E AF                         XRA     A
E01F 32EAFF                     STA     XYFLAG
E022            * INITIALIZE USARTS AT PORTS 3,5,7
E022 0E03                       MVI     C,3             ;STARTING PORT
E024 0606       INILOOP         MVI     B,6             ;NO OF COMMANDS
E026 2112E0                     LXI     H,INITABLE
E029 EDB3                       OUTIR                   ;BLOCK OUTPUT
E02B 0C                         INR     C
E02C 0C                         INR     C
E02D 79                         MOV     A,C
E02E FE09                       CPI     9               ;DO 3 PORTS
E030 20F2                       JRNZ    INILOOP
E032            * PATCH RST 7
E032 3EC3                       MVI     A,0C3H          ;JUMP
E034 323800                     STA     38H             ;RST 7
E037 21D7E6                     LXI     H,DUMPREGS
E03A 223900                     SHLD    39H
E03D            * DISPLAY SIGN ON
E03D CDBBE4                     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 CD3AE5                     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 CC8AE3                     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 43E5           CMDTB   DW      WASCII          ;A
E07D 47E2                   DW      BOOT            ;B
E07F F1E2                   DW      COMPR           ;C
E081 C7E5                   DW      HEXRUL          ;D
E083 DCE7                   DW      EXTCOM          ;E
E085 14E3                   DW      FIND            ;F
E087 AFE0                   DW      EXEC            ;G
E089 65E2                   DW      RAM             ;H
E08B 62E3                   DW      PINPT           ;I
E08D 96E1                   DW      WARM            ;J
E08F C1E7                   DW      SETBRK          ;K
E091 71E2                   DW      LORAM           ;L
E093 A5E2                   DW      MOVEB           ;M
E095 CDE2                   DW      NDMT            ;N
E097 74E3                   DW      POUTP           ;O
E099 14E6                   DW      PROGRAM         ;P
E09B 79E1                   DW      CHKSM           ;Q
E09D CBE6                   DW      DREGS           ;R
E09F 21E3                   DW      SRCH            ;S
E0A1 C3E1                   DW      TMEM            ;T
E0A3 56E2                   DW      USER            ;U
E0A5 00E8                   DW      DSBOOT          ;V
E0A7 02E8                   DW      MSBOOT          ;W
E0A9 96E2                   DW      EXCHG           ;X
E0AB ARE1                   DW      ECHO            ;Y
E0AD 7DE2                   DW      ZEROM           ;Z
E0AF                *
E0AF                *** EXECUTE THE PROGRAM AT THE ADDRESS ***
E0AF                *
E0AF CDD3E4         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 C38AE3                 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 CD8AE3         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 CD8AE3         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 CDD3E4             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 CDD3E4             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 CDD3E4             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 CDD3E4             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 CD8AE3                     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 C32BE7                     JMP     PT2S            ;FOR ADDRESS
E21D                    *
E21D F5         ERR             PUSH    PSW             ;SAVE ACC
E21E CDDFE2                     CALL    PTAD            ;PRINT ADD,
E221 7B                         MOV     A,B             ;DATA
E222 CD2BE7                     CALL    PT2S            ;WRITTEN
E225 F1                         POP     PSW             ;DATA READ
E226 F5         PT2             PUSH    PSW
E227 CD2DE2                     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                    * DISK BOOTSTRAP
E247 CDD3E4     BOOT            CALL    PTSTNG
E24A 424F4F54                   DTH     'BOOT DISK'
E24E 20444953
E252 CB
E253 C300F8                     JMP     PR+1800H
E256                    *
E256                    * JUMP TO USER RAM
E256 CDD3E4     USER            CALL    PTSTNG
E259 55534552                   DTH     'USER AREA'
E25D 20415245
E261 C1
E262 C30001                     JMP     0100H
E265                    *
E265                    * JUMP TO RAM AT PR+1C00
E265 CDD3E4     RAM             CALL    PTSTNG
E268 48492052                   DTH     'HI RAM'
```



```
E26C 41CD
E26E C300FC                     JMP     PR+1C00H
E271                    *
E271                    * JUMP TO RAM AT 0
E271 CDD3E4     LORAM           CALL    PTSTNG
E274 4C4F2052                   DTH     'LO RAM'
E278 41CD
E27A C30000                     JMP     0
E27D                    *
E27D                    * ZERO OR FILL MEMORY WITH A CONSTANT
E27D CDD3E4     ZEROM           CALL    PTSTNG
E280 46494C4C                   DTH     'FILL '
E284 A0
E285 CD0EE1                     CALL    TAHEX           ;READ ADDRESSES
E288 E5                         PUSH    H               ;SAVE H
E289 CD0AE1                     CALL    AHE2            ;READ 2 DIGITS
E28C EB                         XCHG
E28D E3                         XTHL                    ;RESTORE H,L
E28E C1                         POP     B
E28F 71         ZLOOP           MOV     M,C             ;WRITE INTO MEM
E290 CD3FE2                     CALL    BMP             ;COMP ADD, INCR H
E293 C8                         RZ                      ;RETURN IF DONE
E294 18F9                       JR      ZLOOP           ;CONTINUE TIL DONE
E296                    * EXCHANGE OR MOVE A BLOCK OF MEMORY
E296 47         EXCHG           MOV     B,A
E297 CDD3E4                     CALL    PTSTNG
E29A 45584348                   DTH     'EXCHANGE '
E29E 414E4745
E2A2 A0
E2A3 1809                       JR      MOVENTR
E2A5 47         MOVEB           MOV     B,A             ;SAVE CODE
E2A6 CDD3E4                     CALL    PTSTNG
E2A9 4D4F5645                   DTH     'MOVE '
E2AD A0
E2AE CD0EE1     MOVENTR         CALL    TAHEX           ;READ ADDRESSES
E2B1 E5                         PUSH    H
E2B2 CDBDE0                     CALL    AHEX
E2B5 EB                         XCHG
E2B6 E3                         XTHL                    ;BACK TO NORMAL
E2B7 4E         MLOOP           MOV     C,M
E2B8 E3                         XTHL
E2B9 78                         MOV     A,B
E2BA FE4D                       CPI     'M'
E2BC 2804                       JRZ     NEXCH
E2BE 7E                         MOV     A,M
E2BF E3                         XTHL
E2C0 77                         MOV     M,A
E2C1 E3                         XTHL
E2C2 71         NEXCH           MOV     M,C
E2C3 23                         INX     H
E2C4 E3                         XTHL
E2C5 CD3FE2                     CALL    BMP
E2C8 CA4CE0                     JZ      START
E2CB 18EA                       JR      MLOOP
E2CD                    * NON DESTRUCTIVE MEMORY TEST
E2CD CDD3E4     NDMT            CALL    PTSTNG
E2D0 4D454D20                   DTH     'MEM CHECK'
```

```
E2D4 43484543
E2D8 CB
E2D9 210000             LXI     H,0             ;START AT ZERO
E2DC 4E         NDLOP   MOV     C,M
E2DD 06FF               MVI     B,0FFH
E2DF 70                 MOV     M,B
E2E0 7E                 MOV     A,M
E2E1 B8                 CMP     B
E2E2 C2EAE2             JNZ     ERRJP           ;PRINT ERROR
E2E5 0600               MVI     B,0
E2E7 70                 MOV     M,B
E2E8 7E                 MOV     A,M
E2E9 B8                 CMP     B
E2EA C21DE2     ERRJP   JNZ     ERR
E2ED 71                 MOV     M,C
E2EE 23                 INX     H
E2EF 18EB               JR      NDLOP
E2F1            * COMPARE TWO BLOCKS OF MEMORY
E2F1 CDD3E4     COMPR   CALL    PTSTNG
E2F4 434F4D50           DTH     'COMPARE '
E2F8 415245A0
E2FC CD0EE1             CALL    TAHEX
E2FF E5                 PUSH    H
E300 CDBDE0             CALL    AHEX
E303 EB                 XCHG
E304 7E         VMLOP   MOV     A,M
E305 23                 INX     H
E306 E3                 XTHL
E307 BE                 CMP     M
E308 46                 MOV     B,M
E309 C41DE2             CNZ     ERR
E30C CD3FE2             CALL    BMP
E30F E3                 XTHL
E310 20F2               JRNZ    VMLOP
E312 F1                 POP     PSW
E313 C9                 RET
E314            * SEARCH FOR SPECIFIC CODES
E314 F5         FIND    PUSH    PSW
E315 CDD3E4             CALL    PTSTNG
E318 46494E44           DTH     'FIND-2 '
E31C 2D32A0
E31F 180D               JR      SRCHENT
E321 F5         SRCH    PUSH    PSW
E322 CDD3E4             CALL    PTSTNG
E325 53454152           DTH     'SEARCH-1 '
E329 43482D31
E32D A0
E32E CD0EE1     SRCHENT CALL    TAHEX
E331 E5                 PUSH    H               ;SAVE H
E332 CD0AE1             CALL    AHE2            ;READ 2 DIGITS
E335 EB                 XCHG                    ;H=CODE,D=F
E336 45                 MOV     B,L             ;PUT CODE IN B
E337 E1                 POP     H               ;RESTORE H
E338 F1                 POP     PSW
E339 FE53               CPI     'S'
E33B F5                 PUSH    PSW
E33C 2807               JRZ     CONT
```



```
E33E E5                 PUSH    H
E33F CD0AE1             CALL    AHE2            ;READ 2 DIGITS
E342 EB                 XCHG
E343 4D                 MOV     C,L
E344 E1                 POP     H
E345 7E         CONT    MOV     A,M             ;READ MEMORY
E346 B8                 CMP     B               ;COMPARE TO CODE
E347 2012               JRNZ    SKP             ;SKIP IF NO COMP
E349 F1                 POP     PSW             ;FETCH CONTROL
E34A FE53               CPI     'S'
E34C F5                 PUSH    PSW
E34D 2806               JRZ     OBCP
E34F 23                 INX     H
E350 7E                 MOV     A,M
E351 2B                 DCX     H
E352 B9                 CMP     C
E353 2006               JRNZ    SKP
E355 23         OBCP    INX     H
E356 7E                 MOV     A,M             ;READ NEXT BYTE
E357 2B                 DCX     H               ;DECR ADDRESS
E358 CD1DE2             CALL    ERR             ;PRINT CODES
E35B CD3FE2     SKP     CALL    BMP             ;CHECK IF DONE
E35E 20E5               JRNZ    CONT            ;BACK FOR MORE
E360 F1                 POP     PSW
E361 C9                 RET
E362            *
E362            * INPUT DATA FROM A PORT
E362 CDD3E4     PINPT   CALL    PTSTNG
E365 494E5055           DTH     'INPUT '
E369 54A0
E36B CD0AE1             CALL    AHE2            ;READ 2 DIGITS
E36E 4B                 MOV     C,E
E36F ED78               INP     A
E371 C326E2             JMP     PT2
E374            *
E374            * OUTPUT TO A PORT
E374 CDD3E4     POUTP   CALL    PTSTNG
E377 4F555450           DTH     'OUTPUT '
E37B 5554A0
E37E CD0AE1             CALL    AHE2            ;READ 2 DIGITS
E381 CD0AE1             CALL    AHE2            ;READ 2 DIGITS
E384 4D                 MOV     C,L
E385 ED59               OUTP    E
E387 C9                 RET
E388            *
```

```
E388                    *
E388                    ********************************************
E388                    *                                          *
E388                    *     VIDEO DRIVER FOR  FLASHWRITER II     *
E388                    *                                          *
E388                    ********************************************
E388                    *
E388    F000 =          PAGE            EQU     PR+1000H        ;SCREEN LOCATION
E388    0020 =          SPACE           EQU     20H
E388    0004 =          CLRSCRN         EQU     4
E388                    ********************************************
E388                    *                                          *
E388                    * CONTROL CODE COMMANDS:                   *
E388                    *   (B) HOME CURSOR                        *
E388                    *   (D) CLEAR SCREEN                       *
E388                    *   (E) PRINT CONTROL CODE                 *
E388                    *   (H) BACKSPACE                          *
E388                    *   (I) TAB                                *
E388                    *   (J) LINEFEED                           *
E388                    *   (M) CARRIAGE RETURN                    *
E388                    *   (N) NO CURSOR                          *
E388                    *   (P) CLEAR TO END OF SCREEN             *
E388                    *   (Q) CLEAR TO END OF LINE               *
E388                    *   (R) CURSOR DOWN                        *
E388                    *   (T) TOGGLE REVERSE VIDEO               *
E388                    *   (U) CURSOR UP                          *
E388                    *   (W) CURSOR LEFT                        *
E388                    *   (X) CLEAR TO START OF LINE             *
E388                    *   (Z) CURSOR RIGHT                       *
E388                    *   ESC XY POSITION LEAD-IN                *
E388                    *                                          *
E388                    ********************************************
E388                    *
E388                    * VIDEO BOARD PARAMETERS
E388    0050 =          HORIZ           EQU     80              ;NO. OF CHARACTERS
E388    0018 =          VERT            EQU     24              ;NO. OF LINES
E388                    *
E388 3E14               TVIDEO          MVI     A,'T'-64        ;TOGGLE VIDEO
E38A                    *
E38A F5                 VIDEO           PUSH    PSW
E38B C5                                 PUSH    B
E38C D5                                 PUSH    D
E38D E5                                 PUSH    H
E38E E67F                               ANI     07FH
E390 4F                                 MOV     C,A
E391 3A00E8                             LDA     BASE+800H
E394 FEC3                               CPI     0C3H            ;PROM THERE?
E396 79                                 MOV     A,C
E397 CC00E8                             CZ      BASE+800H       ;CALL IT IF SO
E39A CD6FE4             DISPL           CALL    LIFTCURS        ;ERASE CURSOR
E39D 3AEAFF                             LDA     XYFLAG
E3A0 A7                                 ANA     A
E3A1 280A                               JRZ     NOXY
E3A3 3D                                 DCR     A
E3A4 32EAFF                             STA     XYFLAG
E3A7 CABEE4                             JZ      YPOS
E3AA C3B5E4                             JMP     XPOS
```



```
E3AD 79                 NOXY            MOV     A,C             ;RECOVER CHARACTER
E3AE FE20                               CPI     SPACE           ;PRINTING CODE?
E3B0 F2E4E3                             JP      PRINT
E3B3 FE1C                               CPI     PCL-TABL        ;TOO LARGE?
E3B5 F251E4                             JP      RET
E3B8 E5                                 PUSH    H               ;CURSOR IN MEMORY
E3B9 21C7E3                             LXI     H,TABL          ;TABLE START
E3BC 5F                                 MOV     E,A
E3BD 1600                               MVI     D,0
E3BF 19                                 DAD     D
E3C0 5E                                 MOV     E,M
E3C1 21E3E3                             LXI     H,PCL
E3C4 19                                 DAD     D
E3C5 E3                                 XTHL                    ;RECOVER H
E3C6 C9                                 RET                     ;EXECUTE ROUTINE
E3C7                    * CONTROL CHARACTER JUMP TABLE
E3C7 6E                 TABL            DB      RET-PCL         ;@
E3C8 6E                                 DB      RET-PCL         ;A
E3C9 63                                 DB      HOME-PCL        ;B HOME CURSOR
E3CA 6E                                 DB      RET-PCL         ;C
E3CB 60                                 DB      FORM-PCL        ;D CLEAR SCREEN
E3CC 00                                 DB      PCL-PCL         ;E PRT CONTROL
E3CD 6E                                 DB      RET-PCL         ;F
E3CE 6E                                 DB      RET-PCL         ;G
E3CF 42                                 DB      DBACKSP-PCL     ;H BACKSPACE
E3D0 59                                 DB      TAB-PCL         ;I TAB OVER
E3D1 12                                 DB      LINF-PCL        ;J LINE FEED
E3D2 6E                                 DB      RET-PCL         ;K
E3D3 6E                                 DB      RET-PCL         ;L
E3D4 6A                                 DB      CRET-PCL        ;M CARRIAGE RET
E3D5 71                                 DB      RET+3-PCL       ;N NO CURSOR
E3D6 6E                                 DB      RET-PCL         ;O
E3D7 A7                                 DB      CLEND-PCL       ;P CLR SCN TO END
E3D8 AC                                 DB      CLLINE-PCL      ;Q CLR LINE TO END
E3D9 12                                 DB      LINF-PCL        ;R CURSOR DOWN
E3DA 6E                                 DB      RET-PCL         ;S
E3DB 76                                 DB      TVIDF-PCL       ;T TOGGLE VIDEO
E3DC 80                                 DB      CURSUP-PCL      ;U CURSOR UP
E3DD 6E                                 DB      RET-PCL         ;V
E3DE 50                                 DB      BACKSP-PCL      ;W CURSOR LEFT
E3DF E4                                 DB      CLSTRT-PCL      ;X CLR START OF LN
E3E0 6E                                 DB      RET-PCL         ;Y
E3E1 06                                 DB      EOL-PCL         ;Z CURSOR RIGHT
E3E2 CB                                 DB      LEDIN-PCL       ;[ ESC=XY LEADIN
E3E3                    *
E3E3                    * PRINT CODE IN B REGARDLESS
E3E3 48                 PCL             MOV     C,B
E3E4                    * PRINT THE CHARACTER ON THE SCREEN
E3E4 3ADDFF             PRINT           LDA     VFL
E3E7 A9                                 XRA     C
E3E8 77                                 MOV     M,A
E3E9                    * EOL CHECKS THE CURS POS FOR END OF LINE
E3E9 3ADBFF             EOL             LDA     CURPOS
E3EC 3C                                 INR     A
E3ED FE50                               CPI     HORIZ
E3EF 385D                               JRC     TABRET
E3F1 AF                                 XRA     A
```

```
E3F2 32DBFF                    STA     CURPOS
E3F5            * MOVE DN 1 LINE
E3F5 3ADCFF     LINF           LDA     LINENO
E3F8 FE17                      CPI     VERT-1
E3FA 2023                      JRNZ    NOSCRL
E3FC            * SCROLL UP ONE LINE
E3FC 215000     SCROLL         LXI     H,HORIZ
E3FF ED5BDFFF                  LDED    TOSCN
E403 19                        DAD     D
E404 EDA0       SCRL           LDI
E406 EDA0                      LDI
E408 7C                        MOV     A,H
E409 FEF7                      CPI     HORIZ*VERT+PAGE/256
E40B 20F7                      JRNZ    SCRL
E40D 7D                        MOV     A,L
E40E FE80                      CPI     HORIZ*VERT+PAGE&0FFH
E410 20F2                      JRNZ    SCRL
E412 3ADCFF                    LDA     LINENO
E415            * ERASE BOTTOM LINE
E415 EB         EBOTL          XCHG
E416 0650                      MVI     B,HORIZ
E418 3620       ELOP           MVI     M,SPACE
E41A 23                        INX     H
E41B 05                        DCR     B
E41C 20FA                      JRNZ    ELOP
E41E 3D                        DCR     A
E41F 3C         NOSCRL         INR     A
E420 32DCFF                    STA     LINENO
E423 182C                      JR      RET
E425            *
E425            * ERASE BEFORE BACKSPACING
E425 3620       DBACKSP        MVI     M,20H
E427 3ADBFF                    LDA     CURPOS
E42A A7                        ANA     A
E42B 2824                      JRZ     RET
E42D 3D                        DCR     A
E42E 2B                        DCX     H
E42F 3620                      MVI     M,20H
E431 181B                      JR      TABRET
E433            * MOVE THE CURSOR BACK
E433 3ADBFF     BACKSP         LDA     CURPOS
E436 3D                        DCR     A
E437 F24EE4                    JP      TABRET
E43A 1811                      JR      CRET
E43C            * TAB OVER TO THE NEXT 8 MULTIPLE
E43C 3ADBFF     TAB            LDA     CURPOS
E43F F607                      ORI     7
E441 18A9                      JR      EOL+3
E443            * CLEAR THE SCREEN AND HOME UP
E443 CD9CE4     FORM           CALL    CLEAR
E446 AF         HOME           XRA     A
E447 32DCFF                    STA     LINENO
E44A 32DDFF                    STA     VFL             ;CLR VID FLAG
E44D            * CARRIAGE RETURN
E44D AF         CRET           XRA     A
E44E 32DBFF     TABRET         STA     CURPOS
E451            * RETURN TO THE CALLING ROUTINE
```



```
E451 CD6FE4     RET            CALL    LIFTCURS
E454 E1                        POP     H
E455 D1                        POP     D
E456 C1                        POP     B
E457 F1                        POP     PSW
E458 C9                        RET
E459 3ADDFF     TVIDF          LDA     VFL
E45C EE80                      XRI     80H
E45E 32DDFF                    STA     VFL
E461 18EE                      JR      RET
E463            *
E463            * MOVE THE CURSOR UP
E463 3ADCFF     CURSUP         LDA     LINENO
E466 A7                        ANA     A
E467 28E8                      JRZ     RET
E469 3D                        DCR     A
E46A 32DCFF     STORLN         STA     LINENO
E46D 18E2                      JR      RET
E46F            * CALCULATE MEM ADD FROM CURSOR POSITION
E46F 2180F7     LIFTCURS       LXI     H,HORIZ*VERT+PAGE
E472 11B0FF                    LXI     D,-HORIZ
E475 3ADCFF                    LDA     LINENO
E478 3C         CLOP           INR     A
E479 19                        DAD     D
E47A FE18                      CPI     VERT
E47C 20FA                      JRNZ    CLOP                    ;OPTIMIZED AT
E47E ED5BDBFF   CPIN           LDED    CURPOS
E482 1600                      MVI     D,0
E484 19                        DAD     D
E485            * REVERSE THE VIDEO
E485 7E                        MOV     A,M
E486 EE80                      XRI     80H
E488 77                        MOV     M,A
E489 C9                        RET
E48A            * CLEAR TO END OF SCREEN
E48A CDA5E4     CLEND          CALL    WRSPC
E48D 18C2                      JR      RET
E48F            * CLEAR TO END OF LINE
E48F 3ADBFF     CLLINE         LDA     CURPOS
E492 3620                      MVI     M,20H
E494 23                        INX     H
E495 3C                        INR     A
E496 FE50                      CPI     50H
E498 20F8                      JRNZ    CLLINE+3
E49A 18B5                      JR      RET
E49C            * CLEAR THE SCREEN
E49C 2100F0     CLEAR          LXI     H,PAGE
E49F 22DFFF                    SHLD    TOSCN
E4A2 22EAFF                    SHLD    XYFLAG
E4A5 3620       WRSPC          MVI     M,20H
E4A7 23                        INX     H
E4A8 7C                        MOV     A,H
E4A9 FEF8                      CPI     PAGE+2048/256
E4AB 20F8                      JRNZ    WRSPC
E4AD C9                        RET
E4AE            *
E4AE            * PROCESS LEAD IN CODE
```

```
E4AE 3E02       LEDIN           MVI     A,2
E4B0 32EAFF                     STA     XYFLAG
E4B3 189C                       JR      RET
E4B5            *
E4B5            * SET X AND Y CURSOR POSITIONS
E4B5 79         XPOS            MOV     A,C
E4B6 FE50                       CPI     80
E4B8 3802                       JRC     XINRG
E4BA 3E4F                       MVI     A,79
E4BC 1890       XINRG           JR      TABRET
E4BE            *
E4BE 79         YPOS            MOV     A,C
E4BF FE18                       CPI     24
E4C1 3802                       JRC     YINRG
E4C3 3E17                       MVI     A,23
E4C5 18A3       YINRG           JR      STORLN
E4C7            *
E4C7 AF         CLSTRT          XRA     A
E4C8 32DBFF                     STA     CURPOS
E4CB CD6FE4                     CALL    LIFTCURS
E4CE 18BF                       JR      CLLINE
E4D0   E4D0 =   MSEND           EQU     $
E4D0            * CURSOR STORAGE LOCATIONS
E4D0                            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
E4D0            * PRINT A STRING
E4D0 CDDFE0     RPTSTNG         CALL    CRLF            ;CRLF FIRST
E4D3 E3         PTSTNG          XTHL
E4D4 7E                         MOV     A,M
E4D5 23                         INX     H
E4D6 E3                         XTHL
E4D7 A7                         ANA     A
E4D8 CD8AE3                     CALL    VIDEO           ;PRINT IT
E4DB F8                         RM
E4DC 18F5                       JR      PTSTNG
E4DE            * SIGN ON MESSAGE
E4DE 3E04       SIGN            MVI     A,4             ;CLEAR SCREEN
E4E0 CD8AE3                     CALL    VIDEO
E4E3 2150F1                     LXI     H,PAGE+150H
E4E6 E5                         PUSH    H
E4E7 1151F1                     LXI     D,PAGE+151H
E4EA 013000                     LXI     B,30H
E4ED 3612                       MVI     M,12H           ;GRAPHIC CHARACTER
E4EF EDB0                       LDIR
E4F1 E1                         POP     H
E4F2 11A0F1                     LXI     D,PAGE+1A0H
E4F5 018002                     LXI     B,640
E4F8 EDB0                       LDIR
E4FA CDD3E4                     CALL    PTSTNG
```



```
E4FD 1B                         DB      27              ;ESC
E4FE 2007                       DD      2007H           ;X=32 Y=7
E500 20564543                   DT      ' VECTOR GRAPHIC '
E504 544F5220
E508 47524150
E50C 48494320
E510 1B                         DB      27              ;ESC
E511 2008                       DD      2008H           ;X=32 Y=8
E513 20202020                   DT      '     MONITOR     '
E517 4D4F4E49
E51B 544F5220
E51F 20202020
E523 1B                         DB      27              ;ESC
E524 2009                       DD      2009H           ;X=32 Y=9
E526 20205645                   DT      '  VERSION 4.1   '
E52A 5253494F
E52E 4E20342E
E532 31202020
E536 1B                         DB      27              ;ESC
E537 008D                       DD      8DH             ;X=0 Y=13
E539 C9                         RET
E53A CDD0E4     PROMPT          CALL    RPTSTNG
E53D 4D6F6E3E                   DTH     'Mon> '
E541 A0
E542 C9                         RET
E543            *
E543            *WIDE ASCII DUMP
E543 CDD3E4     WASCII          CALL    PTSTNG
E546 41534349                   DTH     'ASCII DUMP '
E54A 49204455
E54E 4D50A0
E551 CD0EE1                     CALL    TAHEX
E554 CD97E5                     CALL    HOMEC
E557            * MAKE A RULER FOR ASCII DUMP
E557 78         RULELP          MOV     A,B
E558 FE40                       CPI     64
E55A 281A                       JRZ     TERMLIN
E55C E60F                       ANI     0FH
E55E 2810                       JRZ     NUMBER
E560 E603                       ANI     3
E562 2808                       JRZ     MARKER
E564 3E20                       MVI     A,' '
E566 CD8AE3     REENTR          CALL    VIDEO
E569 04                         INR     B
E56A 18EB                       JR      RULELP
E56C 3E6C       MARKER          MVI     A,'l'
E56E 18F6                       JR      REENTR
E570 78         NUMBER          MOV     A,B
E571 CD2DE2                     CALL    BINH
E574 18F3                       JR      REENTR+3
E576            * TOGGLE REVERSE VIDEO
E576 CD88E3     TERMLIN         CALL    TVIDEO
E579 CD03E6     WDMP1           CALL    SETSCRLL
E57C CD0FE2                     CALL    PTAD
E57F 0E3F                       MVI     C,63
E581 CD88E5                     CALL    WDMP2
E584 FA79E5                     JM      WDMP1
```

```
E587 C8                         RZ
E588 7E         WDMP2           MOV     A,M
E589 47                         MOV     B,A
E58A 3E05                       MVI     A,'E'-64
E58C CD8AE3                     CALL    VIDEO
E58F CD3FE2                     CALL    BMP
E592 C8                         RZ
E593 0D                         DCR     C
E594 F8                         RM
E595 18F1                       JR      WDMP2
E597            * HOME CURSOR, PRINT "ADDR"
E597 CDD0E4     HOMEC           CALL    RPTSTNG
E59A 14                         DB      'T'-64
E59B 41444452                   DTH     'ADDR '
E59F A0
E5A0 0600                       MVI     B,0
E5A2 3E18                       MVI     A,24
E5A4 32DEFF                     STA     WIDTH
E5A7 C9                         RET
E5A8            * MAKE A RULER FOR HEX DUMP
E5A8 78         HEXRULER        MOV     A,B
E5A9 FE10                       CPI     16
E5AB 2806                       JRZ     HEXRCT
E5AD CD2BE7                     CALL    PT2S
E5B0 04                         INR     B
E5B1 18F5                       JR      HEXRULER
E5B3            * EXTEND FOR ASCII
E5B3 CDDAE0     HEXRCT          CALL    SPCE
E5B6 CDDAE0                     CALL    SPCE
E5B9 0600                       MVI     B,0
E5BB 78         HEXRLP          MOV     A,B
E5BC FE10                       CPI     16
E5BE C8                         RZ
E5BF E60F                       ANI     0FH
E5C1 CD31E2                     CALL    BINL
E5C4 04                         INR     B
E5C5 18F4                       JR      HEXRLP
E5C7            * HEX DUMP ROUTINE
E5C7 CDD3E4     HEXRUL          CALL    PTSTNG
E5CA 48455820                   DTH     'HEX DUMP '
E5CE 44554D50
E5D2 A0
E5D3 CD0EE1                     CALL    TAHEX
E5D6 CD97E5                     CALL    HOMEC
E5D9 CDA8E5                     CALL    HEXRULER
E5DC CD88E3                     CALL    TVIDEO
E5DF CD03E6                     CALL    SETSCRLL
E5E2 CD0FE2     HLP1            CALL    PTAD
E5E5 E5                         PUSH    H
E5E6 D5                         PUSH    D
E5E7 0E10                       MVI     C,16
E5E9 7E         HLP2            MOV     A,M
E5EA CD2BE7                     CALL    PT2S
E5ED 23                         INX     H
E5EE 0D                         DCR     C
E5EF C2E9E5                     JNZ     HLP2
E5F2 D1                         POP     D
```



```
E5F3 E1                         POP     H
E5F4 0E0F                       MVI     C,15
E5F6 CDDAE0                     CALL    SPCE
E5F9 CDDAE0                     CALL    SPCE
E5FC CD88E5                     CALL    WDMP2
E5FF FADFE5                     JM      HLP1-3
E602 C9                         RET
E603            * CHECK TO SET SCROLL POINT
E603 3ADEFF     SETSCRLL        LDA     WIDTH
E606 3D                         DCR     A
E607 32DEFF                     STA     WIDTH
E60A 2007                       JRNZ    CTSCRL
E60C 0150F0                     LXI     B,PAGE+50H      ;2ND LINE
E60F ED43DFFF                   SBCD    TOSCN           ;SCROLL POINT
E613 C9         CTSCRL          RET
E614            *
E614            * PROGRAM MEMORY
E614 CDD3E4     PROGRAM         CALL    PTSTNG
E617 50524F47                   DTH     'PROGRAM '
E61B 52414DA0
E61F CDBDE0                     CALL    AHEX            ;ADDR IN HL
E622 ED53E1FF                   SDED    TCURPOS
E626 CD97E5                     CALL    HOMEC           ;PRINT "ADDR"
E629 CDA8E5                     CALL    HEXRULER
E62C CD88E3                     CALL    TVIDEO
E62F AF                         XRA     A
E630 32DEFF                     STA     WIDTH
E633 CD9DE6                     CALL    PRTiLINE        ;PRINT LINE CONT H
E636 CD2FE1     POLLOOP         CALL    ESCAPE
E639 CDEDE0                     CALL    HEX
E63C 2AE1FF                     LHLD    TCURPOS
E63F 301A                       JRNC    MODMEM
E641            * CONTROL CODE TABLE
E641 FE20                       CPI     ' '
E643 2846                       JRZ     CSRT
E645 FE08                       CPI     8
E647 2845                       JRZ     CSLT
E649 FE12                       CPI     'R'-64
E64B 2839                       JRZ     CSDN
E64D FE15                       CPI     'U'-64
E64F 282F                       JRZ     CSUP
E651 FE17                       CPI     'W'-64
E653 2839                       JRZ     CSLT
E655 FE1A                       CPI     'Z'-64
E657 2832                       JRZ     CSRT
E659 18DB                       JR      POLLOOP
E65B            * MODIFY A MEMORY LOCATION
E65B 2AE1FF     MODMEM          LHLD    TCURPOS
E65E 4F                         MOV     C,A
E65F 3ADEFF                     LDA     WIDTH
E662 A7                         ANA     A
E663 7E                         MOV     A,M
E664 280D                       JRZ     LSN1BL
E666 E6F0                       ANI     0F0H
E668 B1                         ORA     C
E669 77         REMEM           MOV     M,A
E66A 3ADEFF                     LDA     WIDTH
```

```
E66D EE01                       XRI     1
E66F 201F                       JRNZ    RTRTN+1
E671 1818                       JR      CSRT
E673 17         LSNIBL          RAL
E674 17                         RAL
E675 17                         RAL
E676 17                         RAL
E677 E6F0                       ANI     0F0H
E679 B1                         ORA     C
E67A 0F                         RRC
E67B 0F                         RRC
E67C 0F                         RRC
E67D 0F                         RRC
E67E 18E9                       JR      REMEM
E680            * MOVE UP ONE LINE
E680 11F0FF     CSUP            LXI     D,-16
E683 19                         DAD     D
E684 1809                       JR      RTRTN
E686            * MOVE DOWN ONE LINE
E686 111000     CSDN            LXI     D,16
E689 18F8                       JR      CSUP+3
E68B            * MOVE RIGHT ONE SPACE
E68B 23         CSRT            INX     H
E68C 1801                       JR      RTRTN
E68E            * MOVE LEFT ONE SPACE
E68E 2B         CSLT            DCX     H
E68F            *
E68F AF         RTRTN           XRA     A
E690 32DEFF                     STA     WIDTH
E693 22E1FF                     SHLD    TCURPOS
E696 3E15       UPAROW          MVI     A,'U'-64
E698 CD8AE3                     CALL    VIDEO
E69B 1896                       JR      POLLOOP-3
E69D            * PRINT A LINE CONTAINING ((H))
E69D 2AE1FF     PRTLINE         LHLD    TCURPOS
E6A0 E5                         PUSH    H
E6A1 D1                         POP     D
E6A2 7D                         MOV     A,L
E6A3 F60F                       ORI     0FH
E6A5 5F                         MOV     E,A
E6A6 E6F0                       ANI     0F0H
E6A8 6F                         MOV     L,A
E6A9 CDE2E5                     CALL    HLP1
E6AC            * NOW PUT CURSOR WHERE IT GOES
E6AC CD6FE4                     CALL    LIFTCURS
E6AF 2AE1FF                     LHLD    TCURPOS
E6B2 7D                         MOV     A,L
E6B3 E60F                       ANI     0FH
E6B5 6F                         MOV     L,A
E6B6 3E05                       MVI     A,5
E6B8 2D         PLOP1           DCR     L
E6B9 FACOE6                     JM      PGCONT
E6BC C603                       ADI     3
E6BE 18F8                       JR      PLOP1
E6C0 6F         PGCONT          MOV     L,A
E6C1 3ADEFF                     LDA     WIDTH
E6C4 85                         ADD     L
```



```
E6C5            * A = 5+3*L+W
E6C5 32DBFF                     STA     CURPOS
E6C8 C36FE4                     JMP     LIFTCURS
E6CB            *
E6CB            *
E6CB            * DISPLAY REGISTERS
E6CB CDD3E4     DREGS           CALL    PTSTNG
E6CE 52454749                   DTH     'REGISTERS'
E6D2 53544552
E6D6 D3
E6D7            * DUMP REGISTERS AFTER ENTRY FROM RST 7
E6D7 E3         DUMPREGS        XTHL
E6D8 F5                         PUSH    PSW
E6D9 CD31E7                     CALL    DISPREGS
E6DC 2B                         DCX     H               ;GET BREAK ADD
E6DD CD0FE2                     CALL    PTAD
E6E0 E1                         POP     H
E6E1 C5                         PUSH    B
E6E2 CD86E7                     CALL    PRTFLGS
E6E5 C1                         POP     B
E6E6 CD12E2                     CALL    PTAD+3          ;PRINT AF
E6E9 E1                         POP     H
E6EA 22B3FF                     SHLD    HLTEMP
E6ED CDA7E7                     CALL    PTHREE          ;PRINT B D H
E6F0 DDE5                       PUSH    IX
E6F2 E1                         POP     H
E6F3 CD12E2                     CALL    PTAD+3          ;PRINT IX
E6F6 FDE5                       PUSH    IY
E6F8 E1                         POP     H
E6F9 CD12E2                     CALL    PTAD+3          ;PRINT IY
E6FC 210000                     LXI     H,0
E6FF 39                         DAD     SP
E700 22E5FF                     SHLD    SPTEMP
E703 CD12E2                     CALL    PTAD+3          ;PRINT SP
E706 08                         EXAF
E707 F5                         PUSH    PSW
E708 E1                         POP     H
E709 CD12E2                     CALL    PTAD+3
E70C D9                         EXX
E70D CDA7E7                     CALL    PTHREE
E710 D9                         EXX
E711 0A                         LDAX    B
E712 CD2BE7                     CALL    PT2S
E715 1A                         LDAX    D
E716 CD2BE7                     CALL    PT2S
E719 2AB3FF                     LHLD    HLTEMP
E71C 7E                         MOV     A,M
E71D CD2BE7                     CALL    PT2S
E720 2AE5FF                     LHLD    SPTEMP
E723 F9                         SPHL
E724 E1                         POP     H
E725 CD12E2                     CALL    PTAD+3
E728 C340E0                     JMP     CLRBRK          ;CLEAR BREAKPOINT
E72B            *
E72B CD26E2     PT2S            CALL    PT2             ;PRINT 2 CHARS
E72E C3DAE0                     JMP     SPCE            ;PRINT SPACE
E731            * DISPLAY REGISTER HEADER ON SCREEN
```

```
E731 CDD0E4     DISPREGS        CALL    RPTSTNG
E734 14                         DB      'T'-64
E735 41444452                   DT      'ADDR FLAGS  AF   BC   DE'
E739 20464C41
E73D 47532020
E741 41462020
E745 20424320
E749 20204445
E74D 20202048                   DT      '   HL   IX   IY   SP '
E751 4C202020
E755 49582020
E759 20495920
E75D 20205350
E761 20                         DB      20H
E762 20204146                   DT      '  AF'
E766 27                         DB      27H             ;'
E767 20204243                   DT      '  BC'
E76B 27                         DB      27H
E76C 20204445                   DT      '  DE'
E770 27                         DB      27H
E771 2020484C                   DT      '  HL'
E775 27                         DB      27H
E776 20404220                   DT      ' @B @D @H @SP '
E77A 40442040
E77E 48204053
E782 5020
E784 94                         DB      'T'+64
E785 C9                         RET
E786                *
E786                * PRINT FLAGS
E786 015A40     PRTFLGS         LXI     B,405AH         ;Z
E789 CDB6E7                     CALL    MASKFLG
E78C 014301                     LXI     B,143H          ;C
E78F CDB6E7                     CALL    MASKFLG
E792 014D80                     LXI     B,804DH         ;M
E795 CDB6E7                     CALL    MASKFLG
E798 014504                     LXI     B,445H          ;E
E79B CDB6E7                     CALL    MASKFLG
E79E 014810                     LXI     B,1048H         ;H
E7A1 CDB6E7                     CALL    MASKFLG
E7A4 C3DAE0                     JMP     SPCE
E7A7                *
E7A7                * PRINT BC DE HL IN ORDER
E7A7 E5         PTHREE          PUSH    H
E7A8 C5                         PUSH    B
E7A9 E1                         POP     H
E7AA CD12E2                     CALL    PTAD+3
E7AD D5                         PUSH    D
E7AE E1                         POP     H
E7AF CD12E2                     CALL    PTAD+3
E7B2 E1                         POP     H
E7B3 C312E2                     JMP     PTAD+3
E7B6                *
E7B6 7D         MASKFLG         MOV     A,L
E7B7 A0                         ANA     B
E7B8 3E20                       MVI     A,20H
E7BA CABAE3                     JZ      VIDEO
```



```
E7BD 79                         MOV     A,C
E7BE C38AE3                     JMP     VIDEO
E7C1                *
E7C1                * SET BREAKPOINT
E7C1 CDD3E4     SETBRK          CALL    PTSTNG
E7C4 42524541                   DTH     'BREAK AT '
E7C8 4B204154
E7CC A0
E7CD CDBDE0                     CALL    AHEX
E7D0 1A                         LDAX    D
E7D1 32E9FF                     STA     BRKCODE
E7D4 ED53E7FF                   SDED    BKPTLOC
E7D8 3EFF                       MVI     A,0FFH          ;RST 7
E7DA 12                         STAX    D
E7DB C9                         RET
E7DC                *
E7DC                * EXTERNAL COMMUNICATIONS
E7DC CDD3E4     EXTCOM          CALL    PTSTNG
E7DF 45585420                   DTH     'EXT COM '
E7E3 434F4DA0
E7E7 DB05       RECEIVE         IN      5
E7E9 E602                       ANI     2
E7EB 2805                       JRZ     NEXCHR
E7ED DB04                       IN      4
E7EF CD8AE3                     CALL    VIDEO
E7F2 CD2FE1     NEXCHR          CALL    ESCAPE
E7F5 28F0                       JRZ     RECEIVE
E7F7 D304                       OUT     4
E7F9 18EC                       JR      RECEIVE
E7FB                *
E7FB                * TEMPORARY STORAGE LOCATIONS FOR REGISTERS,ETC.
E7FB                            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
```