# EXTENDED SYSTEMS MONITOR
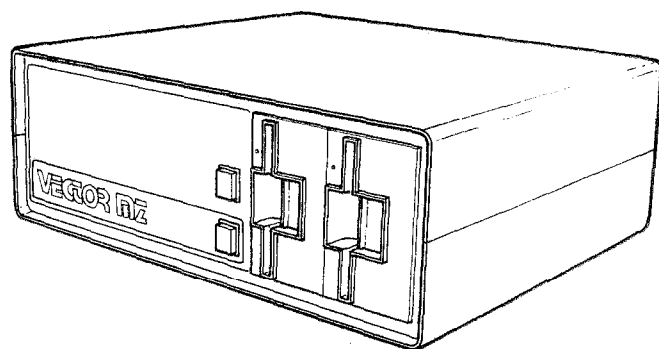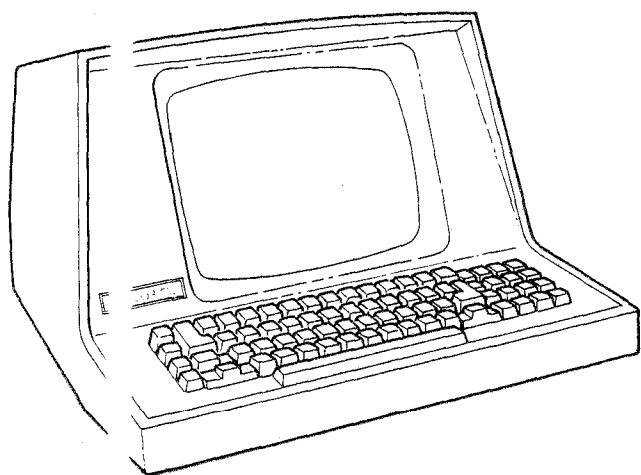# USERS GUIDE

VECTOR GRAPHIC INC.

EXTENDED SYSTEMS MONITOR

Version 4.0

USERS MANUAL

Revision A

October 15, 1979

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.

Extended Systems Monitor User's Manual

## TABLE OF CONTENTS

## GENERAL DESCRIPTION

The Version 4.0 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. Since version 4.0 does not work with serial terminals, use the 2.1 version for this purpose.

Version 4.0 differs from 3.1 in the following key ways:

1) Improved video driver, with a number of additional commands that software can use to manipulate the screen, and easier X-Y positioning.

2) A breakpoint executive command, for quick debugging.

3) A communications program that allows you to use a Vector Graphic system as a dumb terminal. (Executive command "E").

4) Space to include up to 15 keyboard conversions for foreign language keyboards.

5) Much easier to use Executive commands.

6) Control-Q no longer can be used to return to the Monitor Executive.

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 an asterisk (*).

There are 24 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 4.0 Monitor is located at address E000 – E7FF in new Vector Graphic systems. If it is purchased for another location, address references in this manual should be understood accordingly.

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:

$$\begin{array}{r} 10 \\ 40)\overline{\;400} \end{array} \qquad \text{Hex (trivial)}$$

## COMMAND FORMAT

### *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 the WORD MANAGEMENT SYSTEM. ASCII strings embedded in object code are easy to recognize.

### *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 MDOS.

### *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.

### *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.

### *E — EXTERNAL COMMUNICATIONS

The monitor will output anything typed on the keyboard through port 4 on 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 o 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.

### *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 C700, 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.

### *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" (C00B) 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.

### *H - JUMP TO HI RAM

This command jumps to FC00 which is the start of the 1K scratchpad RAM. This is a useful area for small machine language programs.

### *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.

### *J - JUMP TO LOADED DOS

This command permits easy return to the MDOS disk operating system at 0437H, or if not present, jump will be 0437H, which is the MDOS entry location and is subject to change in future releases.

## *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 8, the CPU will jump to the monitor routine that dumps the register contents. The instruction replaced with FF will also be restored. Entry of the monitor at E000 will clear the breakpoint, as will pressing the RESET switch.

## *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.

## *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.

## *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, 0 06 41 will cause an "A" to be printed since 41 is the hex ASCII code for "A".

## *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.

## *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.

## *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.

## *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).

6

### *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.

### *U — JUMP TO 2B00

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

### *W <ADR1> <ADR2> — WIDE ASCII DUMP

This is similar to the "A" command, but with a full screen width display.

### *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).

### *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.

## *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 Bitstreamter I addressed starting at port 4 or the Bistreamer II addressed starting at port 2.  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. E009 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:

    (B)    HOME THE CURSOR
    (D)    CLEAR THE SCREEN AND HOME CURSOR
    (E)    DISPLAY THE CODE IN B REGISTER
    (H)    DESTRUCTIVE BACKSPACE (also BACKSPACE key)
    (I)    TAB OVER TO THE NEXT 8 MULTIPLE (also TAB)
    (J)    LINEFEED (also LF Key)
    (M)    CARRIAGE RETURN (also RETURN key)
    (N)    TOGGLE CURSOR
    (P)    CLEAR TO END OF SCREEN
    (Q)    CLEAR TO END OF LINE
    (R)    CURSOR DOWN (also ↓ )
    (T)    TOGGLE REVERSE VIDEO
    (U)    CURSOR UP (also ↑ )
    (W)    CURSOR LEFT (also ← )
    (X)    CLEAR TO START OF LINE
    (Z)    CURSOR RIGHT (also → )
    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 desireable 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 (Ø5) in A.  Enter the folowing machine code at FCØØ and execute it to demonstrate this feature:

    at FCØØ  Ø6 Ø2 3E Ø5 Ø4 CD Ø9 EØ CD ØC EØ C3 Ø2 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 Ø, Ø. The sequence 1B Ø8 4Ø would cause the cursor to move to line 8, character position 64 on the screen. This feature can 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:           CCNTL    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 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 AHEO 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.

```
Addr  B1 B2 B3 B4  E  Line   Label    Opcod  Operand

0000  E000 =          0000   BASE     EQU    0E000H       ;ASSEMBLY ADDRESS
0000  E000 =          0010   PR       EQU    0E000H       ;PROM/RAM ADDRESS
                      0020            LINK   'M6'
0000  0000           0000   ******** VECTOR MZ MONITOR - VERSION 4.0 ********
0000                 0010   *         R. S. HARP  7/16/79              *
0000                 0020   *         SYSTEM 3 VERSION                 *
0000                 0030   *         ASSEMBLES AT C OR E              *
0000                 0040   *         ONLY FOR FLASHWRITER II          *
0000                 0050   *******************************************
0000                 0060   *
0000                 0070   * SYSTEM EQUATES
0000  0000 =         0080   CONS     EQU    0            ;CONS STATUS PRT
0000  0001 =         0090   COND     EQU    1            ;CONS DATA PORT
0000  0040 =         0100   RCA      EQU    40H          ;RECEIVE FLAG
0000  0000 =         0110   STPOL    EQU    0            ;STATUS POLARITY
0000  FFF0 =         0120   SPTR     EQU    PR+01FF0H    ;STACK POINTER
0000                 0130   *
0000                 0140   ************** COMMAND FORMAT **************
0000                 0150   * A SSSS FFFF ASCII DUMP OF MEMORY       *
0000                 0160   * B JUMP TO BOOTSTRAP LOADER             *
0000                 0170   * C SSSS FFFF CCCC COMPARE BLOCKS        *
0000                 0180   * D SSSS FFFF DUMP MEMORY IN HEX & ASCII *
0000                 0190   * E EXTERNAL COMMUNICATIONS              *
0000                 0200   * F SSSS FFFF DD TWO BYTE SEARCH         *
0000                 0210   * G SSSS GO TO AND EXECUTE               *
0000                 0220   * H JUMP TO HIGH RAM AT FC00             *
0000                 0230   * I PP INPUT FROM PORT                   *
0000                 0240   * J JUMP TO DOS                          *
0000                 0250   * K LLLL SET A BREAKPOINT                *
0000                 0255   * L JUMP TO LOW RAM AT 0                 *
0000                 0260   * M SSSS FFFF DDDD MOVE BLOCK            *
0000                 0270   * N NON DESTRUCTIVE MEMORY TEST          *
0000                 0280   * O PP DD OUTPUT TO PORT                 *
0000                 0290   * P LLLL PROGRAM MEMORY                  *
0000                 0300   * Q SSSS FFFF COMPUTE CHECKSUM           *
0000                 0310   * R DUMP Z-80 REGISTERS                  *
0000                 0320   * S SSSS FFFF DD SEARCH FOR SINGLE BYTE  *
0000                 0330   * T SSSS FFFF TEST MEMORY                *
0000                 0340   * U JUMP TO USER AREA AT 2800            *
0000                 0350   * X SSSS FFFF DDDD EXCHANGE BLOCK        *
0000                 0360   * Y KEYBOARD ECHO                        *
0000                 0370   * Z SSSS FFFF DD ZERO OR FILL MEMORY     *
0000                 0380   *
0000                 0390   *******************************************
0000                 0400            ORG    BASE
0000                 0410   * JUMP TABLE OF ENTRY POINTS
E000  C3 15 E0       0420   MONIT    JMP    INIT         ;INITIALIZE ALL
E003  C3 3C E1       0430   KEYSTT   JMP    KEYST        ;TEST KEYBOARD
E006  C3 41 E1       0440   CONVERT  JMP    CONVDATA     ;INPUT KEYBOARD
E009  C3 3A E3       0450   CRT      JMP    VIDEO        ;OUTPUT TO SCREEN
E00C  C3 2E E1       0455   ESC      JMP    ESCAPE       ;KEYBOARD INPUT
                     0460   *
E00F  D3 03          0470   OTPRTS   OUT    3
E011  D3 05          0480            OUT    5
E013  D3 07          0490            OUT    7
E015  C9             0500            RET
```

Page 1

```
Addr B1 B2 B3 B4 E Line Label          Opcd    Operand

E016 31 D0 FF      0510 INIT           LXI     SP,SPTR      ;INIT STACK
E019 CD 2F E1      0520                CALL    ESCAPE       ;DUMP LATCH
E01C AF            0530                XRA     A
E01D 32 EA FF      0535                STA     XYFLAG
E020 CD 0F E0      0540                CALL    OTPRTS
E023 CD 0F E0      0550                CALL    OTPRTS
E026 CD 0F E0      0560                CALL    OTPRTS
E029 3E 40         0570                MVI     A,40H        ;RESET USART
E02B CD 0F E0      0580                CALL    OTPRTS
E02E 3E CE         0590                MVI     A,0CEH       ;MODE
E030 CD 0F E0      0600                CALL    OTPRTS;
E033 3E 27         0610                MVI     A,27H        ;COMMAND
E035 CD 0F E0      0620                CALL    OTPRTS
E038              0630 * PATCH RST 7
E038 3E C3         0640                MVI     A,0C3H       ;JUMP
E03A 32 38 00      0650                STA     38H          ;RST 7
E03D 21 DB E6      0660                LXI     H,DUMPREGS
E040 22 39 00      0670                SHLD    39H
E043              0680 * DISPLAY SIGN ON
E043 CD DC E4      0690                CALL    SIGN
E046              0700 * CLEAR BREAKPOINT
E046 2A E7 FF      0710 CLRBRK         LHLD    BKPTLOC
E049 11 E9 FF      0720                LXI     D,BRKCODE
E04C ED 53 E7 FF   0730                SDED    BKPTLOC
E050 1A            0740                LDAX    D
E051 77            0750                MOV     M,A
E052 31 D0 FF      0755 START          LXI     SP,SPTR      ;INITIALIZE STACK
E055 CD 38 E5      0760                CALL    PROMPT
E058 CD 2F E1      0770                CALL    ESCAPE       ;READ KEYBOARD
E05B 28 FD         0780                JRZ     START+6
E05D E6 5F         0790                ANI     5FH          ;UPPER AND LOWER
E05F 21 52 E0      0800                LXI     H,START
E062 E5            0810                PUSH    H
E063 FE 04         0820                CPI     'D'-64
E065 CC 8A E3      0830                CZ      VIDEO        ;ECHO CLEARSCN
E068 FE 41         0840                CPI     'A'
E06A D8            0850                RC                   ;TOO SMALL
E06B FE 5B         0860                CPI     05BH
E06D D0            0870                RNC                  ;TOO LARGE
E06E 21 F9 E0      0880                LXI     H,CMDTB+7EH
E071 F5            0890                PUSH    PSW
E072 87            0900                ADD     A
E073 85            0910                ADD     L
E074 6F            0920                MOV     L,A
E075 5E            0930                MOV     E,M
E076 23            0940                INX     H
E077 56            0950                MOV     D,M
E078 EB            0960                XCHG
E079 F1            0970                POP     PSW
E07A E9            0980                PCHL                 ;AWAY WE GO
E07B              0990 * COMMAND TABLE
E07B 47 E5         1000 CMDTB          DW      WASCII       ;A
E07D 47 E2         1010                DW      BOOT         ;B
E07F F1 E2         1020                DW      COMPR        ;C
E081 CB E5         1030                DW      HEXRUL       ;D
E083 E0 E7         1040                DW      EXTCOM       ;E
```

```
Addr B1 B2 B3 B4 E Line Label          Opcd    Operand

E085 14 E3          1050              DW      FIND           ;F
E087 AF E0          1060              DW      EXEC           ;G
E089 65 E2          1070              DW      RAM            ;H
E08B 62 E3          1080              DW      PINPT          ;I
E08D 96 E1          1090              DW      WARM           ;J
E08F C5 E7          1100              DW      SETBRK         ;K
E091 71 E2          1110              DW      LORAM          ;L
E093 A5 E2          1120              DW      MOVEB          ;M
E095 CD E2          1130              DW      NDMT           ;N
E097 74 E3          1140              DW      POUTP          ;O
E099 18 E6          1150              DW.     PROGRAM        ;P
E09B 79 E1          1160              DW      CHKSM          ;Q
E09D CF E6          1170              DW      DREGS          ;R
E09F 21 E3          1180              DW      SRCH           ;S
E0A1 C3 E1          1190              DW      TMEM           ;T
E0A3 56 E2          1200              DW      USER           ;U
E0A5 52 E0          1210              DW      START          ;V
E0A7 52 E0          1220              DW      START          ;W
E0A9 96 E2          1230              DW      EXCHG          ;X
E0AB AE E1          1240              DW      ECHO           ;Y
E0AD 7D E2          1250              DW      ZEROM          ;Z
E0AF               1260  *
E0AF               1270  *** EXECUTE THE PROGRAM AT THE ADDRESS ***
E0AF               1280  *
E0AF CD D1 E4       1290  EXEC        CALL    PTSTNG
E0B2 47 4F 20 54    1300              DTH     'GO TO '
E0B6 4F A0
E0B8 CD BD E0       1310              CALL    AHEX           ;READ ADD FROM KB
E0BB EB             1320              XCHG
E0BC E9             1330              PCHL                   ;JUMP TO IT
E0BD               1340  *
E0BD               1350  *** CONVERT UP TO 4 HEX DIGITS TO BIN
E0BD               1360  *
E0BD 0E 04          1370  AHEX        MVI     C,4            ;COUNT OF 4 DIGIT!
E0BF 21 00 00       1380  AHE0        LXI     H,0            ;16 BIT ZERO
E0C2 CD 2F E1       1390  AHE1        CALL    ESCAPE
E0C5 FE 20          1400              CPI     ' '            ;SPACE?
E0C7 CA E8 E0       1410              JZ      SPCOVR
E0CA CD ED E0       1420              CALL    HEX            ;CHECK VALUE
E0CD 33 F3          1430              JRC     AHE1
E0CF 29             1440              DAD     H              ;MULT H*16
E0D0 29             1450              DAD     H
E0D1 29             1460              DAD     H
E0D2 29             1470              DAD     H
E0D3 85             1480              ADD     L
E0D4 6F             1490              MOV     L,A
E0D5 0D             1500              DCR     C              ;4 DIGITS?
E0D6 C2 C2 E0       1510              JNZ     AHE1           ;KEEP READING
E0D9 EB             1520              XCHG
E0DA 3E 20          1530  SPCE        MVI     A,20H          ;PRINT SPACE
E0DC C3 3A E3       1540  PTCN        JMP     VIDEO
E0DF 3E 0D          1550  CRLF        MVI     A,0DH          ;PRINT CR
E0E1 CD DC E0       1560              CALL    PTCN
E0E4 3E 0A          1570              MVI     A,0AH
E0E6 18 F4          1580              JR      PTCN
E0E8               1590  *
```

```
Addr B1 B2 B3 B4 E Line Label            Opcd    Operand

E0E8 CD 8A E3      1600 SPCOVR           CALL    VIDEO
E0EB 18 EC         1610                  JR      SPCE-1
E0ED               1620 *
E0ED               1630 * CHECK FOR HEX VALUE, CONVERT
E0ED FE 30         1640 HEX              CPI     30H          ;<0
E0EF D8            1650                  RC
E0F0 FE 3A         1660                  CPI     ':'          ;>9
E0F2 38 09         1670                  JRC     NUM
E0F4 E6 5F         1675                  ANI     5FH          ;UPPER & LOWER CAS
E0F6 FE 41         1680                  CPI     'A'          ;<A
E0F8 D8            1690                  RC                   ;
E0F9 FE 47         1700                  CPI     'G'          ;>F
E0FB 3F            1710                  CMC
E0FC D8            1720                  RC
E0FD CD 8A E3      1730 NUM              CALL    VIDEO
E100 D6 30         1740                  SUI     48           ;ASCII BIAS
E102 FE 0A         1750                  CPI     10           ;DIGIT 0-10
E104 38 02         1760                  JRC     ALFA
E106 D6 07         1770                  SUI     7            ;ALPHA BIAS
E108 A7            1780 ALFA             ANA     A            ;CLEAR CY
E109 C9            1790                  RET                  ;WITH CY CLEAR
E10A               1800 *
E10A               1810 * READ 2 DIGITS FROM THE CONSOLE
E10A 0E 02         1820 AHE2             MVI     C,2
E10C 18 B1         1830                  JR      AHE0
E10E               1840 *
E10E               1850 * SHORT ROUTINE TO SAVE CODE
E10E CD BD E0      1860 TAHEX            CALL    AHEX
E111 18 AA         1870                  JR      AHEX
E113               1880 *
E113               1890 *** READ FROM CONSOLE TO REG A ***
E113               1900 *
E113 CD 2F E1      1910 RDCN             CALL    ESCAPE       ;READ KEYBOARD
E116 28 FB         1920                  JRZ     RDCN
E118 FE 60         1930                  CPI     60H
E11A 38 C0         1940                  JRC     PTCN
E11C E6 5F         1950                  ANI     5FH
E11E 18 BC         1960                  JR      PTCN
E120               1970 *
E120 CD 2F E1      1980 PAUSE            CALL    ESCAPE
E123 FE 20         1990                  CPI     20H
E125 C0            2000                  RNZ
E126 CD 2F E1      2010 PLOOP            CALL    ESCAPE
E129 FE 20         2020                  CPI     20H
E12B C2 26 E1      2030                  JNZ     PLOOP
E12E C9            2040                  RET
E12F               2050 *
E12F CD 3C E1      2060 ESCAPE           CALL    KEYSTAT
E132 C8            2070                  RZ
E133 CD 41 E1      2080                  CALL    CONVERT
E136 FE 1B         2090                  CPI     1BH          ;ESCAPE
E138 CA 52 E0      2100                  JZ      START
E13B C9            2110                  RET
E13C               2120 *
E13C DB 00         2130 KEYSTAT          IN      CONS
E13E E6 40         2140                  ANI     RDA
```

```
Addr B1 B2 B3 B4 E Line Label          Opcd    Operand

E140 C9              2150              RET
E141                 2160 *
E141                 2170 * KEYBOARD CODE CONVERSION
E141 DB 01           2180 CONVERT      IN      COND        ;KEYBOARD DATA
E143 E5              2190              PUSH    H
E144 C5              2200              PUSH    B
E145 01 05 00        2210              LXI     B,TABLEND-KTABL/2
E148 21 5B E1        2220              LXI     H,KTABL
E14B ED A1           2230 LOOP         CCI                 ;COMPARE TABLE  -
E14D 28 06           2240              JRZ     FND
E14F 23              2250              INX     H
E150 EA 4B E1        2260              JPE     LOOP        ;CONT LOOKING
E153 18 01           2270              JR      NFND
E155 7E              2280 FND          MOV     A,M         ;NEW CODE
E156 E6 7F           2290 NFND         ANI     7FH         ;MASK DOWN
E158 C1              2300              POP     B
E159 E1              2310              POP     H
E15A C9              2320              RET
E15B                 2330 *
E15B                 2340 * THIS TABLE CAN BE EXTENDED IF DESIRED
E15B E1 5D           2350 KTABL        DD      0E15DH      ;]
E15D F1 5B           2360              DD      0F15BH      ;[
E15F A1 7E           2370              DD      0A17EH      ;~
E161 B1 5C           2380              DD      0B15CH      ;\
E163 60 15           2390              DD      06015H      ;CURSOR UP
E165    E165    =    2400 TABLEND      EQU     $
E165                 2410              ORG     KTABL+30    ;ROOM FOR 15 CONV:
E179                 2420 *
E179                 2430 * CHECKSUM ROUTINE
E179 CD D1 E4        2440 CHKSM        CALL    PTSTNG
E17C 43 48 45 43     2450              DTH     'CHECKSUM '
E180 4B 53 55 4D
E184 A0
E185 CD 0E E1        2460              CALL    TAHEX
E188 06 00           2470              MVI     B,0
E18A 7E              2480 CHKSMLP      MOV     A,M
E18B 30              2490              ADD     B
E18C 47              2500              MOV     B,A
E18D CD 3F E2        2510              CALL    BMP
E190 20 F8           2520              JRNZ    CHKSMLP
E192 73              2530              MOV     A,B
E193 C3 26 E2        2540              JMP     PT2
E196                 2550 *
E196                 2560 * WARM START
E196                 2570 *
E196 CD D1 E4        2580 WARM         CALL    PTSTNG
E199 4A 55 4D 50     2590              DTH     'JUMP TO DOS'
E19D 20 54 4F 20
E1A1 44 4F D3
E1A4 21 E7 04        2600              LXI     H,04E7H     ;MDOS RESTART
E1A7 7E              2610              MOV     A,M
E1A8 FE C3           2620              CPI     0C3H
E1AA C2 00 00        2630              JNZ     0           ;CP/M RESTART
E1AD E9              2640              PCHL                ;MDOS WARM START
E1AE                 2650 *
E1AE                 2660 * KEYBOARD ECHO ROUTINE
```

```
Addr B1 B2 B3 B4 E Line Label              Opcd    Operand

E1AE CD D1 E4       2670 ECHO              CALL    PTSTNG
E1B1 45 43 48 4F    2680                   DTH     'ECHO KEYS '
E1B5 20 4B 45 59
E1B9 53 A0
E1BB CD 2F E1       2690 ECOLP             CALL    ESCAPE          ;LOOK AT KEYBOARD
E1BE C4 DC E0       2700                   CNZ     PTCN            ;PRINT IF KEYPRES.
E1C1 13 F8          2710                   JR      ECOLP           ;CONTINUE LOOPING
E1C3                2720 *
E1C3                2730 *** MEMORY TEST ROUTINE ***
E1C3                2740 *
E1C3 CD D1 E4       2750 TMEM              CALL    PTSTNG
E1C6 54 45 53 54    2760                   DTH     'TEST '
E1CA A0
E1CB CD OE E1       2770                   CALL    TAHEX           ;READ ADDRESSES
E1CE 01 5A 5A       2780                   LXI     B,5A5AH         ;INI B,C
E1D1 CD FD E1       2790 CYCL              CALL    RNDM
E1D4 C5             2800                   PUSH    B               ;KEEP ALL REGS
E1D5 E5             2810                   PUSH    H
E1D6 D5             2820                   PUSH    D
E1D7 CD FD E1       2830 TLOP              CALL    RNDM
E1DA 70             2840                   MOV     M,B             ;WRITE IN MEM
E1DB CD 3F E2       2850                   CALL    BMP
E1DE C2 D7 E1       2860                   JNZ     TLOP            ;REPEAT LOOP
E1E1 D1             2870                   POP     D
E1E2 E1             2880                   POP     H               ;RESTORE ORIG
E1E3 C1             2890                   POP     B               ;VALUES OF
E1E4 E5             2900                   PUSH    H
E1E5 D5             2910                   PUSH    D
E1E6 CD FD E1       2920 RLOP              CALL    RNDM            ;GEN NEW SEQ
E1E9 7E             2930                   MOV     A,M             ;READ MEM
E1EA B8             2940                   CMP     B               ;COMP MEM
E1EB C4 1D E2       2950                   CNZ     ERR             ;CALL ERROR RTN
E1EE CD 3F E2       2960                   CALL    BMP
E1F1 C2 E6 E1       2970                   JNZ     RLOP
E1F4 D1             2980                   POP     D
E1F5 E1             2990                   POP     H
E1F6 3E 2E          3000                   MVI     A,'.'
E1F8 CD 8A E3       3010                   CALL    VIDEO
E1FB 18 D4          3020                   JR      CYCL
E1FD                3030 *** THIS ROUTINE GENERATES RANDOM NOS ***
E1FD CD 20 E1       3040 RNDM              CALL    PAUSE
E200 78             3050                   MOV     A,B             ;LOOK AT B
E201 E6 B4          3060                   ANI     0B4H            ;MASK BITS
E203 A7             3070                   ANA     A               ;CLEAR CY
E204 EA 08 E2       3080                   JPE     PEVE            ;JUMP IF EVEN
E207 37             3090                   STC
E208 79             3100 PEVE              MOV     A,C             ;LOOK AT C
E209 17             3110                   RAL                     ;ROTATE CY IN
E20A 4F             3120                   MOV     C,A             ;RESTORE C
E20B 78             3130                   MOV     A,B             ;LOOK AT B
E20C 17             3140                   RAL                     ;ROTATE CY IN
E20D 47             3150                   MOV     B,A             ;RESTORE B
E20E C9             3160                   RET                     ;RETURN W NEW B,C
E20F                3170 *
E20F                3180 *** ERROR PRINT OUT ROUTINE
E20F                3190 *
```

```
Addr  B1 B2 B3 B4 E Line Label       Opcd    Operand

E20F  CD DF E0       3200 PTAD        CALL    CRLF        ;PRINT CR,LF
E212  CD 20 E1       3210             CALL    PAUSE
E215  7C             3220             MOV     A,H         ;PRINT
E216  CD 26 E2       3230             CALL    PT2         ;ASCII
E219  7D             3240             MOV     A,L         ;CODES
E21A  C3 2F E7       3250             JMP     PT2S        ;FOR ADDRESS
E21D                 3260 *
E21D  F5             3270 ERR         PUSH    PSW         ;SAVE ACC
E21E  CD 0F E2       3280             CALL    PTAD        ;PRINT ADD.
E221  78             3290             MOV     A,B         ;DATA
E222  CD 2F E7       3300             CALL    PT2S        ;WRITTEN
E225  F1             3310             POP     PSW         ;DATA READ
E226  F5             3320 PT2         PUSH    PSW
E227  CD 2D E2       3330             CALL    BINH
E22A  F1             3340             POP     PSW
E22B  18 04          3350             JR      BINL
E22D  1F             3360 BINH        RAR                 ;SHIFT RHT 4 BITS
E22E  1F             3370             RAR
E22F  1F             3380             RAR
E230  1F             3390             RAR
E231  E6 0F          3400 BINL        ANI     0FH         ;LOW 4 BITS
E233  C6 30          3410             ADI     48          ;ASCII BIAS
E235  FE 3A          3420             CPI     58          ;DIGIT 0-9
E237  DA DC E0       3430             JC      PTCN
E23A  C6 07          3440             ADI     7           ;DIGIT A-F
E23C  C3 DC E0       3450             JMP     PTCN
E23F                 3460 *
E23F                 3470 * COMPARE ADDRESSES AND INCREMENT H
E23F  7B             3480 BMP         MOV     A,E
E240  95             3490             SUB     L
E241  20 02          3500             JRNZ    GOON
E243  7A             3510             MOV     A,D
E244  9C             3520             SBB     H
E245  23             3530 GOON        INX     H
E246  C9             3540             RET
E247                 3550 *
E247                 3560 * DISK BOOTSTRAP
E247  CD D1 E4       3570 BOOT        CALL    PTSTNG
E24A  42 4F 4F 54    3580             DTH     'BOOT DISK'
E24E  20 44 49 53
E252  CB
E253  C3 00 F8       3590             JMP     PR+1800H
E256                 3600 *
E256                 3610 * JUMP TO USER RAM
E256  CD D1 E4       3620 USER        CALL    PTSTNG
E259  55 53 45 52    3630             DTH     'USER AREA'
E25D  20 41 52 45
E261  C1
E262  C3 00 2B       3640             JMP     2B00H
E265                 3650 *
E265                 3660 * JUMP TO RAM AT PR+1C00
E265  CD D1 E4       3670 RAM         CALL    PTSTNG
E268  48 49 20 52    3680             DTH     'HI RAM'
E26C  41 CD
E26E  C3 00 FC       3690             JMP     PR+1C00H
E271                 3691 *
```

```
Addr B1 B2 B3 B4 E Line Label           Opcd    Operand

E271                   3692 * JUMP TO RAM AT 0
E271 CD D1 E4          3693 LORAM           CALL    PTSTNG
E274 4C 4F 20 52       3694                 DTH     'LO RAM'
E278 41 CD
E27A C3 00 00          3695                 JMP     0
E27D                   3700 *
E27D                   3710 * ZERO OR FILL MEMORY WITH A CONSTANT
E27D CD D1 E4          3720 ZEROM           CALL    PTSTNG
E280 46 49 4C 4C       3730                 DTH     'FILL '
E284 A0
E285 CD 0E E1          3740                 CALL    TAHEX       ;READ ADDRESSES
E288 E5                3750                 PUSH    H           ;SAVE H
E289 CD 0A E1          3760                 CALL    AHE2        ;READ 2 DIGITS
E28C EB                3770                 XCHG
E28D E3                3780                 XTHL                ;RESTORE H,L
E28E C1                3790                 POP     B
E28F 71                3800 ZLOOP           MOV     M,C         ;WRITE INTO MEM
E290 CD 3F E2          3810                 CALL    BMP         ;COMP ADD, INCR H
E293 C8                3820                 RZ                  ;RETURN IF DONE
E294 18 F9             3830                 JR      ZLOOP       ;CONTINUE TIL DONE
E296                   3840 * EXCHANGE OR MOVE A BLOCK OF MEMORY
E296 47                3850 EXCHG           MOV     B,A
E297 CD D1 E4          3860                 CALL    PTSTNG
E29A 45 58 43 48       3870                 DTH     'EXCHANGE '
E29E 41 4E 47 45
E2A2 A0
E2A3 18 09             3880                 JR      MOVENTR
E2A5 47                3890 MOVEB           MOV     B,A         ;SAVE CODE
E2A6 CD D1 E4          3900                 CALL    PTSTNG
E2A9 4D 4F 56 45       3910                 DTH     'MOVE '
E2AD A0
E2AE CD 0E E1          3920 MOVENTR         CALL    TAHEX       ;READ ADDRESSES
E2B1 E5                3930                 PUSH    H
E2B2 CD BD E0          3940                 CALL    AHEX
E2B5 EB                3950                 XCHG
E2B6 E3                3960                 XTHL                ;BACK TO NORMAL
E2B7 4E                3970 MLOOP           MOV     C,M
E2B8 E3                3980                 XTHL
E2B9 78                3990                 MOV     A,B
E2BA FE 4D             4000                 CPI     'M'
E2BC 28 04             4010                 JRZ     NEXCH
E2BE 7E                4020                 MOV     A,M
E2BF E3                4030                 XTHL
E2C0 77                4040                 MOV     M,A
E2C1 E3                4050                 XTHL
E2C2 71                4060 NEXCH           MOV     M,C
E2C3 23                4070                 INX     H
E2C4 E3                4080                 XTHL
E2C5 CD 3F E2          4090                 CALL    BMP
E2C8 CA 52 E0          4100                 JZ      START
E2CB 18 EA             4110                 JR      MLOOP
E2CD                   4120 * NON DESTRUCTIVE MEMORY TEST
E2CD CD D1 E4          4130 NDMT            CALL    PTSTNG
E2D0 4D 45 4D 20       4140                 DTH     'MEM CHECK'
E2D4 43 48 45 43
E2D8 C8
```

**Rev. 4.0-A  10/15/79**

```
Addr B1 B2 B3 B4 E Line Label            Opcd    Operand

E2D9 21 00 00       4150                 LXI     H,0           ;START AT ZERO
E2DC 4E             4160 NDLOP           MOV     C,M
E2DD 06 FF          4170                 MVI     B,0FFH
E2DF 70             4180                 MOV     M,B
E2E0 7E             4190                 MOV     A,M
E2E1 B8             4200                 CMP     B
E2E2 C2 EA E2       4210                 JNZ     ERRJP         ;PRINT ERROR
E2E5 06 00          4220                 MVI     B,0
E2E7 70             4230                 MOV     M,B
E2E8 7E             4240                 MOV     A,M
E2E9 B8             4250                 CMP     B
E2EA C2 1D E2       4260 ERRJP           JNZ     ERR
E2ED 71             4270                 MOV     M,C
E2EE 23             4280                 INX     H
E2EF 18 EB          4290                 JR      NDLOP
E2F1                4300 * COMPARE TWO BLOCKS OF MEMORY
E2F1 CD D1 E4       4310 COMPR           CALL    PTSTNG
E2F4 43 4F 4D 50    4320                 DTH     'COMPARE '
E2F8 41 52 45 A0
E2FC CD 0E E1       4330                 CALL    TAHEX
E2FF E5             4340                 PUSH    H
E300 CD BD E0       4350                 CALL    AHEX
E303 EB             4360                 XCHG
E304 7E             4370 VMLOP           MOV     A,M
E305 23             4380                 INX     H
E306 E3             4390                 XTHL
E307 BE             4400                 CMP     M
E308 46             4410                 MOV     B,M
E309 C4 1D E2       4420                 CNZ     ERR
E30C CD 3F E2       4430                 CALL    BMP
E30F E3             4440                 XTHL
E310 20 F2          4450                 JRNZ    VMLOP
E312 F1             4460                 POP     PSW
E313 C9             4470                 RET
E314                4480 * SEARCH FOR SPECIFIC CODES
E314 F5             4490 FIND            PUSH    PSW
E315 CD D1 E4       4500                 CALL    PTSTNG
E318 46 49 4E 44    4510                 DTH     'FIND-2 '
E31C 2D 32 A0
E31F 18 0D          4520                 JR      SRCHENT
E321 F5             4530 SRCH            PUSH    PSW
E322 CD D1 E4       4540                 CALL    PTSTNG
E325 53 45 41 52    4550                 DTH     'SEARCH-1 '
E329 43 48 2D 31
E32D A0
E32E CD 0E E1       4560 SRCHENT         CALL    TAHEX
E331 E5             4570                 PUSH    H           ;SAVE H
E332 CD 0A E1       4580                 CALL    AHE2        ;READ 2 DIGITS
E335 EB             4590                 XCHG                ;H=CODE,D=F
E336 45             4600                 MOV     B,L         ;PUT CODE IN B
E337 E1             4610                 POP     H           ;RESTORE H
E338 F1             4620                 POP     PSW
E339 FE 53          4630                 CPI     'S'
E33B F5             4640                 PUSH    PSW
E33C 28 07          4650                 JRZ     CONT
E33E E5             4660                 PUSH    H
```

```
Addr B1 B2 B3 B4 E Line Label          Opcd   Operand

E33F CD 0A E1      4670          CALL   AHE2      ;READ 2 DIGITS
E342 EB            4680          XCHG
E343 4D            4690          MOV    C,L
E344 E1            4700          POP    H
E345 7E            4710 CONT     MOV    A,M       ;READ MEMORY
E346 B8            4720          CMP    B         ;COMPARE TO CODE
E347 20 12         4730          JRNZ   SKP       ;SKIP IF NO COMP
E349 F1            4740          POP    PSW       ;FETCH CONTROL
E34A FE 53         4750          CPI    'S'
E34C F5            4760          PUSH   PSW
E34D 28 06         4770          JRZ    OBCP
E34F 23            4780          INX    H
E350 7E            4790          MOV    A,M
E351 2B            4800          DCX    H
E352 B9            4810          CMP    C
E353 20 06         4820          JRNZ   SKP
E355 23            4830 OBCP     INX    H
E356 7E            4840          MOV    A,M       ;READ NEXT BYTE
E357 2B            4850          DCX    H         ;DECR ADDRESS
E358 CD 1D E2      4860          CALL   ERR       ;PRINT CODES
E35B CD 3F E2      4870 SKP      CALL   BMP       ;CHECK IF DONE
E35E 20 E5         4880          JRNZ   CONT      ;BACK FOR MORE
E360 F1            4890          POP    PSW
E361 C9            4900          RET
E362               4910 *
E362               4920 * INPUT DATA FROM A PORT
E362 CD D1 E4      4930 PINPT    CALL   PTSTNG
E365 49 4E 50 55   4940          DTH    'INPUT '
E369 54 A0
E36B CD 0A E1      4950          CALL   AHE2      ;READ 2 DIGITS
E36E 4B            4960          MOV    C,E
E36F ED 78         4970          INP    A
E371 C3 26 E2      4980          JMP    PT2
E374               4990 *
E374               5000 * OUTPUT TO A PORT
E374 CD D1 E4      5010 POUTP    CALL   PTSTNG
E377 4F 55 54 50   5020          DTH    'OUTPUT '
E37B 55 54 A0
E37E CD 0A E1      5030          CALL   AHE2      ;READ 2 DIGITS
E381 CD 0A E1      5040          CALL   AHE2      ;READ 2 DIGITS
E384 4D            5050          MOV    C,L
E385 ED 59         5060          OUTP   E
E387 C9            5070          RET
E388               5080 *
```

```
Addr B1 B2 B3 B4 E Line Label          Opcd    Operand

E388                5090 *
E388                5100 ******************************************
E388                5110 *                                        *
E388                5120 *     VIDEO DRIVER FOR  FLASHWRITER II    *
E388                5130 *                                        *
E388                5140 ******************************************
E388                5145 *
E388     F000    =  5150 PAGE           EQU     PR+1000H      ;SCREEN LOCATION
E388     0020    =  5160 SPACE          EQU     20H
E388     0004    =  5170 CLRSCRN        EQU     4
E388                5180 *
E388                5190 * COMMANDS:
E388                5200 * (B) HOME CURSOR
E388                5210 * (D) CLEAR SCREEN
E388                5220 * (E) PRINT CONTROL CODE
E388                5230 * (H) BACKSPACE
E388                5240 * (I) TAB
E388                5250 * (J) LINEFEED
E388                5260 * (M) CARRIAGE RETURN
E388                5270 * (N) NO CURSOR
E388                5280 * (P) CLEAR TO END OF SCREEN
E388                5290 * (Q) CLEAR TO END OF LINE
E388                5300 * (R) CURSOR DOWN
E388                5310 * (T) TOGGLE REVERSE VIDEO
E388                5320 * (U) CURSOR UP
E388                5330 * (W) CURSOR LEFT
E388                5340 * (X) CLEAR TO START OF LINE
E388                5350 * (Z) CURSOR RIGHT
E388                5360 * ESC XY POSITION LEAD-IN
E388                5370 *
E388                5380 * VIDEO BOARD PARAMETERS
E388     0050    =  5390 HORIZ          EQU     30            ;NO. OF CHARACTER
E388     0018    =  5400 VERT           EQU     24            ;NO. OF LINES
E388                5410 *
E388  3E 14         5415 TVIDEO         MVI     A,'T'-64      ;TOGGLE VIDEO
E38A                5416 *
E38A  F5            5420 VIDEO          PUSH    PSW
E38B  C5            5430                PUSH    B
E38C  D5            5440                PUSH    D
E38D  E5            5450                PUSH    H
E38E  E6 7F         5460                ANI     07FH
E390  4F            5470                MOV     C,A
E391  3A 00 E8      5480                LDA     BASE+800H
E394  FE C3         5490                CPI     0C3H          ;PROM THERE?
E396  79            5500                MOV     A,C
E397  CC 00 E8      5510                CZ      BASE+800H     ;CALL IT IF SO
E39A  CD 6F E4      5520 DISPL          CALL    LIFTCURS      ;ERASE CURSOR
E39D  3A EA FF      5530                LDA     XYFLAG
E3A0  A7            5540                ANA     A
E3A1  28 0A         5550                JRZ     NOXY
E3A3  3D            5560                DCR     A
E3A4  32 EA FF      5570                STA     XYFLAG
E3A7  CA 3C E4      5580                JZ      YPOS
E3AA  C3 B3 E4      5590                JMP     XPOS
E3AD  79            5600 NOXY           MOV     A,C           ;RECOVER CHARACTE
E3AE  FE 20         5610                CPI     SPACE         ;PRINTING CODE?
```

| Addr | B1 | B2 | B3 | B4 | E | Line | Label | Opcd | Operand | |
|------|----|----|----|----|---|------|-------|------|---------|---|
| E3B0 | F2 | E4 | E3 | | | 5620 | | JP | PRINT | |
| E3B3 | FE | 1C | | | | 5630 | | CPI | PCL-TABL | ;TOO LARGE? |
| E3B5 | F2 | 51 | E4 | | | 5640 | | JP | RET | |
| E3B8 | E5 | | | | | 5650 | | PUSH | H | ;CURSOR IN MEMORY |
| E3B9 | 21 | C7 | E3 | | | 5660 | | LXI | H,TABL | ;TABLE START |
| E3BC | 5F | | | | | 5670 | | MOV | E,A | |
| E3BD | 16 | 00 | | | | 5680 | | MVI | D,0 | |
| E3BF | 19 | | | | | 5690 | | DAD | D | |
| E3C0 | 5E | | | | | 5700 | | MOV | E,M | |
| E3C1 | 21 | E3 | E3 | | | 5710 | | LXI | H,PCL | |
| E3C4 | 19 | | | | | 5720 | | DAD | D | |
| E3C5 | E3 | | | | | 5730 | | XTHL | | ;RECOVER H |
| E3C6 | C9 | | | | | 5740 | | RET | | ;EXECUTE ROUTINE |
| E3C7 | | | | | | 5750 | * CONTROL CHARACTER JUMP TABLE | | | |
| E3C7 | 6E | | | | | 5760 | TABL | DB | RET-PCL | ;@ |
| E3C8 | 6E | | | | | 5770 | | DB | RET-PCL | ;A |
| E3C9 | 63 | | | | | 5780 | | DB | HOME-PCL | ;B HOME CURSOR |
| E3CA | 6E | | | | | 5790 | | DB | RET-PCL | ;C |
| E3CB | 60 | | | | | 5800 | | DB | FORM-PCL | ;D CLEAR SCREEN |
| E3CC | 00 | | | | | 5810 | | DB | PCL-PCL | ;E PRT CONTROL |
| E3CD | 6E | | | | | 5820 | | DB | RET-PCL | ;F |
| E3CE | 6E | | | | | 5830 | | DB | RET-PCL | ;G |
| E3CF | 42 | | | | | 5840 | | DB | DBACKSP-PCL | ;H BACKSPACE |
| E3D0 | 59 | | | | | 5850 | | DB | TAB-PCL | ;I TAB OVER |
| E3D1 | 12 | | | | | 5860 | | DB | LINF-PCL | ;J LINE FEED |
| E3D2 | 6E | | | | | 5870 | | DB | RET-PCL | ;K |
| E3D3 | 6E | | | | | 5880 | | DB | RET-PCL | ;L |
| E3D4 | 6A | | | | | 5890 | | DB | CRET-PCL | ;M CARRIAGE RET |
| E3D5 | 71 | | | | | 5900 | | DB | RET+3-PCL | ;N NO CURSOR |
| E3D6 | 6E | | | | | 5910 | | DB | RET-PCL | ;O |
| E3D7 | A5 | | | | | 5920 | | DB | CLEND-PCL | ;P CLR SCN TO END |
| E3D8 | AA | | | | | 5930 | | DB | CLLINE-PCL | ;Q CLR LINE TO EN |
| E3D9 | 12 | | | | | 5940 | | DB | LINF-PCL | ;R CURSOR DOWN |
| E3DA | 6E | | | | | 5950 | | DB | RET-PCL | ;S |
| E3DB | 76 | | | | | 5960 | | DB | TVIDF-PCL | ;T TOGGLE VIDEO |
| E3DC | 80 | | | | | 5970 | | DB | CURSUP-PCL | ;U CURSOR UP |
| E3DD | 6E | | | | | 5980 | | DB | RET-PCL | ;V |
| E3DE | 50 | | | | | 5990 | | DB | BACKSP-PCL | ;W CURSOR LEFT |
| E3DF | E2 | | | | | 6000 | | DB | CLSTRT-PCL | ;X CLR START OF L |
| E3E0 | 6E | | | | | 6010 | | DB | RET-PCL | ;Y |
| E3E1 | 06 | | | | | 6020 | | DB | EOL-PCL | ;Z CURSOR RIGHT |
| E3E2 | C9 | | | | | 6030 | | DB | LEDIN-PCL | ;[ ESC=XY LEADIN |
| E3E3 | | | | | | 6040 | * | | | |
| E3E3 | | | | | | 6050 | * PRINT CODE IN B REGARDLESS | | | |
| E3E3 | 48 | | | | | 6060 | PCL | MOV | C,B | |
| E3E4 | | | | | | 6070 | * PRINT THE CHARACTER ON THE SCREEN | | | |
| E3E4 | 3A | DD | FF | | | 6080 | PRINT | LDA | VFL | |
| E3E7 | A9 | | | | | 6090 | | XRA | C | |
| E3E8 | 77 | | | | | 6100 | | MOV | M,A | |
| E3E9 | | | | | | 6110 | * EOL CHECKS THE CURS POS FOR END OF LINE | | | |
| E3E9 | 3A | DB | FF | | | 6120 | EOL | LDA | CURPOS | |
| E3EC | 3C | | | | | 6130 | | INR | A | |
| E3ED | FE | 50 | | | | 6140 | | CPI | HORIZ | |
| E3EF | 38 | 5D | | | | 6150 | | JRC | TABRET | |
| E3F1 | AF | | | | | 6160 | | XRA | A | |
| E3F2 | 32 | DB | FF | | | 6170 | | STA | CURPOS | |

```
Addr B1 B2 B3 B4 E Line Label          Opcd    Operand

E3F5                6180 * MOVE DN 1 LINE
E3F5 3A DC FF       6190 LINF           LDA     LINENO
E3F8 FE 17          6200                CPI     VERT-1
E3FA 20 23          6210                JRNZ    NOSCRL
E3FC                6220 * SCROLL UP ONE LINE
E3FC 21 50 00       6230 SCROLL         LXI     H,HORIZ
E3FF ED 5B DF FF    6240                LDED    TOSCN
E403 19             6250                DAD     D
E404 ED A0          6260 SCRL           LDI
E406 ED A0          6270                LDI
E408 7C             6280                MOV     A,H
E409 FE F7          6290                CPI     HORIZ*VERT+PAGE/256
E40B 20 F7          6300                JRNZ    SCRL
E40D 7D             6310                MOV     A,L
E40E FE 80          6320                CPI     HORIZ*VERT+PAGE&0FFH
E410 20 F2          6330                JRNZ    SCRL
E412 3A DC FF       6340                LDA     LINENO
E415                6350 * ERASE BOTTOM LINE
E415 EB             6360 EBOTL          XCHG
E416 06 50          6370                MVI     B,HORIZ
E418 36 20          6380 ELOP           MVI     M,SPACE
E41A 23             6390                INX     H
E41B 05             6400                DCR     B
E41C 20 FA          6410                JRNZ    ELOP
E41E 3D             6420                DCR     A
E41F 3C             6430 NOSCRL         INR     A
E420 32 DC FF       6440                STA     LINENO
E423 18 2C          6450                JR      RET
E425                6460 *
E425                6470 * ERASE BEFORE BACKSPACING
E425 36 20          6480 DBACKSP        MVI     M,20H
E427 3A DB FF       6481                LDA     CURPOS
E42A A7             6482                ANA     A
E42B 28 24          6483                JRZ     RET
E42D 3D             6484                DCR     A
E42E 2B             6485                DCX     H
E42F 36 20          6486                MVI     M,20H
E431 18 1B          6487                JR      TABRET
E433                6510 * MOVE THE CURSOR BACK
E433 3A DB FF       6520 BACKSP         LDA     CURPOS
E436 3D             6530                DCR     A
E437 F2 4E E4       6540                JP      TABRET
E43A 18 11          6550                JR      CRET
E43C                6560 * TAB OVER TO THE NEXT 8 MULTIPLE
E43C 3A DB FF       6570 TAB            LDA     CURPOS
E43F F6 07          6580                ORI     7
E441 18 A9          6590                JR      EOL+3
E443                6600 * CLEAR THE SCREEN AND HOME UP
E443 CD 9A E4       6610 FORM           CALL    CLEAR
E446 AF             6620 HOME           XRA     A
E447 32 DC FF       6630                STA     LINENO
E44A 32 DD FF       6640                STA     VFL         ;CLR VID FLAG
E44D                6650 * CARRIAGE RETURN
E44D AF             6660 CRET           XRA     A
E44E 32 DB FF       6670 TABRET         STA     CURPOS
E451                6680 * RETURN TO THE CALLING ROUTINE
```

```
Addr B1 B2 B3 B4 E Line Label         Opcd  Operand

E451 CD 6F E4       6690 RET           CALL  LIFTCURS
E454 E1             6700               POP   H
E455 D1             6710               POP   D
E456 C1             6720               POP   B
E457 F1             6730               POP   PSW
E458 C9             6740               RET
E459 3A DD FF       6750 TVIDF         LDA   VFL
E45C EE 80          6760               XRI   80H
E45E 32 DD FF       6770               STA   VFL
E461 18 EE          6780               JR    RET
E463                6790 *
E463                6800 * MOVE THE CURSOR UP
E463 3A DC FF       6810 CURSUP        LDA   LINENO
E466 A7             6820               ANA   A
E467 28 E8          6830               JRZ   RET
E469 3D             6840               DCR   A
E46A 32 DC FF       6850 STORLN        STA   LINENO
E46D 18 E2          6860               JR    RET
E46F                6870 * CALCULATE MEM ADD FROM CURS POS
E46F 21 B0 EF       6880 LIFTCURS      LXI   H,PAGE-HORIZ
E472 11 50 00       6890               LXI   D,HORIZ
E475 3A DC FF       6900               LDA   LINENO
E478 3C             6910               INR   A
E479 19             6920 CLOP          DAD   D
E47A 3D             6930               DCR   A
E47B 20 FC          6940               JRNZ  CLOP
E47D ED 5B DB FF    6950 CFIN          LDED  CURPOS
E481 57             6960               MOV   D,A
E482 19             6970               DAD   D
E483                6980 * REVERSE THE VIDEO
E483 7E             6990               MOV   A,M
E484 EE 80          7000               XRI   80H
E486 77             7010               MOV   M,A
E487 C9             7020               RET
E488                7030 * CLEAR TO END OF SCREEN
E488 CD A3 E4       7040 CLEND         CALL  WRSPC
E48B 18 C4          7050               JR    RET
E48D                7060 * CLEAR TO END OF LINE
E48D 3A DB FF       7070 CLLINE        LDA   CURPOS
E490 36 20          7080               MVI   M,20H
E492 23             7090               INX   H
E493 3C             7100               INR   A
E494 FE 50          7110               CPI   50H
E496 20 F8          7120               JRNZ  CLLINE+3
E498 18 B7          7130               JR    RET
E49A                7140 * CLEAR THE SCREEN
E49A 21 00 F0       7150 CLEAR         LXI   H,PAGE
E49D 22 DF FF       7160               SHLD  TOSCN
E4A0 22 EA FF       7170               SHLD  XYFLAG
E4A3 36 20          7180 WRSPC         MVI   M,20H
E4A5 23             7190               INX   H
E4A6 7C             7200               MOV   A,H
E4A7 FE F3          7210               CPI   PAGE+2048/256
E4A9 20 F8          7220               JRNZ  WRSPC
E4AB C9             7230               RET
E4AC                7240 *
```

```
Addr  B1 B2 B3 B4 E Line  Label          Opcd    Operand

E4AC                 7250 * PROCESS LEAD IN CODE
E4AC  3E 02          7260 LEDIN          MVI     A,2
E4AE  32 EA FF       7270                STA     XYFLAG
E4B1  18 9E          7280                JR      RET
E4B3                 7290 *
E4B3                 7300 * SET X AND Y CURSOR POSITIONS
E4B3  79             7310 XPOS           MOV     A,C
E4B4  FE 50          7320                CPI     80
E4B6  38 02          7330                JRC     XINRG
E4B8  3E 4F          7340                MVI     A,79
E4BA  18 92          7350 XINRG          JR      TABRET
E4BC                 7360 *
E4BC  79             7370 YPOS           MOV     A,C
E4BD  FE 18          7380                CPI     24
E4BF  38 02          7390                JRC     YINRG
E4C1  3E 17          7400                MVI     A,23
E4C3  18 A5          7410 YINRG          JR      STORLN
E4C5                 7420 *
E4C5  AF             7430 CLSTRT         XRA     A
E4C6  32 DB FF       7440                STA     CURPOS
E4C9  CD 6F E4       7450                CALL    LIFTCURS
E4CC  18 BF          7460                JR      CLLINE
E4CE      E4CE   =   7470 MSEND          EQU     $
E4CE                 7480 * CURSOR STORAGE LOCATIONS
E4CE                 7490                ORG     SPTR+0BH
FFDB  00             7500 CURPOS         DB      0           ;POS ON LINE
FFDC  00             7510 LINENO         DB      0           ;LINE NUMBER
FFDD  00             7520 VFL            DB      0           ;REVERSE VID FLAG
FFDE  00             7530 WIDTH          DB      0           ;PRINT WIDTH
FFDF  00 F0          7540 TOSCN          DW      PAGE        ;TOP OF SCREEN
FFE1  00 00          7550 TCURPOS        DW      0           ;TEMP POSITION
FFE3                 0030                LINK    'M5'
FFE3                 0000 * ADDITIONS TO 4.0 MONITOR
FFE3                 0010                ORG     MSEND
E4CE                 0020 * PRINT A STRING
E4CE  CD DF E0       0021 RPTSTNG        CALL    CRLF        ;CRLF FIRST
E4D1  E3             0030 PTSTNG         XTHL
E4D2  7E             0040                MOV     A,M
E4D3  23             0050                INX     H
E4D4  E3             0060                XTHL
E4D5  A7             0070                ANA     A
E4D6  CD 8A E3       0030                CALL    VIDEO       ;PRINT IT
E4D9  F8             0090                RM
E4DA  18 F5          0100                JR      PTSTNG
E4DC                 0110 * SIGN ON MESSAGE
E4DC  3E 04          0120 SIGN           MVI     A,4         ;CLEAR SCREEN
E4DE  CD 8A E3       0125                CALL    VIDEO
E4E1  21 50 F1       0130                LXI     H,PAGE+150H
E4E4  E5             0140                PUSH    H
E4E5  11 51 F1       0150                LXI     D,PAGE+151H
E4E8  01 30 00       0160                LXI     B,30H
E4EB  36 A0          0170                MVI     M,0A0H
E4ED  ED B0          0180                LDIR
E4EF  E1             0190                POP     H
E4F0  11 A0 F1       0200                LXI     D,PAGE+1A0H
E4F3  01 80 02       0210                LXI     B,640
```

```
Addr B1 B2 B3 B4 E Line Label              Opcd   Operand

E4F6 ED B0           0220                   LDIR
E4F8 CD D1 E4        0270                   CALL   PTSTNG
E4FB 1B              0272                   DB     27           ;ESC
E4FC 20 07           0274                   DD     2007H        ;X=32 Y=7
E4FE 20 56 45 43     0280                   DT     ' VECTOR GRAPHIC '
E502 54 4F 52 20
E506 47 52 41 50
E50A 48 49 43 20
E50E 1B              0290                   DB     27           ;ESC
E50F 20 08           0300                   DD     2008H        ;X=32 Y=8
E511 20 20 20 20     0310                   DT     '    MONITOR      '
E515 4D 4F 4E 49
E519 54 4F 52 20
E51D 20 20 20 20
E521 1B              0320                   DB     27           ;ESC
E522 20 09           0330                   DD     2009H        ;X=32 Y=9
E524 20 20 56 45     0340                   DT     '  VERSION 4.0   '
E528 52 53 49 4F
E52C 4E 20 34 2E
E530 30 20 20 20
E534 1B              0350                   DB     27           ;ESC
E535 00 3D           0360                   DD     8DH          ;X=0 Y=13
E537 C9              0380                   RET
E538 CD CE E4        0390 PROMPT            CALL   RPTSTNG
E53B 4D 6F 6E 3E     0410                   DTH    'Mon> '
E53F A0
E540 21 00 F0        0420                   LXI    H,PAGE
E543 22 DF FF        0430                   SHLD   TOSCN        ;FULL SCROLL
E546 C9              0440                   RET
E547                 0450 *
E547                 0460 *WIDE ASCII DUMP
E547 CD D1 E4        0470 WASCII            CALL   PTSTNG
E54A 41 53 43 49     0480                   DTH    'ASCII DUMP '
E54E 49 20 44 55
E552 4D 50 A0
E555 CD 0E E1        0490                   CALL   TAHEX
E558 CD 9B E5        0500                   CALL   HOMEC
E55B                 0510 * MAKE A RULER FOR ASCII DUMP
E55B 78              0520 RULELP            MOV    A,B
E55C FE 40           0530                   CPI    64
E55E 28 1A           0540                   JRZ    TERMLIN
E560 E6 0F           0550                   ANI    0FH
E562 28 10           0560                   JRZ    NUMBER
E564 E6 03           0570                   ANI    3
E566 28 03           0580                   JRZ    MARKER
E568 3E 20           0590                   MVI    A,' '
E56A CD 8A E3        0600 REENTR            CALL   VIDEO
E56D 04              0610                   INR    B
E56E 18 EB           0620                   JR     RULELP
E570 3E 6C           0630 MARKER            MVI    A,'1'
E572 18 F6           0640                   JR     REENTR
E574 78              0650 NUMBER            MOV    A,B
E575 CD 2D E2        0660                   CALL   BINH
E578 18 F3           0670                   JR     REENTR+3
E57A                 0680 * TOGGLE REVERSE VIDEO
E57A CD 38 E3        0690 TERMLIN           CALL   TVIDEO
```

```
Addr B1 B2 B3 B4 E Line Label            Opcd    Operand

E57D CD 07 E6      0710 WDMP1            CALL    SETSCRLL
E580 CD 0F E2      0720                  CALL    PTAD
E583 0E 3F         0730                  MVI     C,63
E585 CD 3C E5      0740                  CALL    WDMP2
E588 FA 7D E5      0750                  JM      WDMP1
E58B C8            0760                  RZ
E58C 7E            0770 WDMP2            MOV     A,M
E58D 47            0780                  MOV     B,A
E58E 3E 05         0790                  MVI     A,'E'-64
E590 CD 8A E3      0800                  CALL    VIDEO
E593 CD 3F E2      0810                  CALL    BMP
E596 C8            0820                  RZ
E597 0D            0830                  DCR     C
E598 F8            0840                  RM
E599 18 F1         0850                  JR      WDMP2
E59B               0860 * HOME CURSOR, PRINT "ADDR"
E59B CD CE E4      0870 HOMEC            CALL    RPTSTNG
E59E 14            0890                  DB      'T'-64
E59F 41 44 44 52   0900                  DTH     'ADDR '
E5A3 A0
E5A4 06 00         0910                  MVI     B,0
E5A6 3E 18         0920                  MVI     A,24
E5A8 32 DE FF      0930                  STA     WIDTH
E5AB C9            0940                  RET
E5AC               0950 * MAKE A RULER FOR HEX DUMP
E5AC 78            0960 HEXRULER         MOV     A,B
E5AD FE 10         0970                  CPI     16
E5AF 28 06         0980                  JRZ     HEXRCT
E5B1 CD 2F E7      0990                  CALL    PT2S
E5B4 04            1000                  INR     B
E5B5 18 F5         1010                  JR      HEXRULER
E5B7               1020 * EXTEND FOR ASCII
E5B7 CD DA E0      1030 HEXRCT           CALL    SPCE
E5BA CD DA E0      1040                  CALL    SPCE
E5BD 06 00         1050                  MVI     B,0
E5BF 73            1060 HEXRLP           MOV     A,B
E5C0 FE 10         1070                  CPI     16
E5C2 C8            1080                  RZ
E5C3 E6 0F         1090                  ANI     0FH
E5C5 CD 31 E2      1100                  CALL    BINL
E5C8 04            1110                  INR     B
E5C9 18 F4         1120                  JR      HEXRLP
E5CB               1130 * HEX DUMP ROUTINE
E5CB CD D1 E4      1140 HEXRUL           CALL    PTSTNG
E5CE 48 45 53 20   1150                  DTH     'HEX DUMP '
E5D2 44 55 4D 50
E5D6 A0
E5D7 CD 0E E1      1160                  CALL    TAHEX
E5DA CD 9B E5      1170                  CALL    HOMEC
E5DD CD AC E5      1180                  CALL    HEXRULER
E5E0 CD 38 E3      1200                  CALL    TVIDEO
E5E3 CD 07 E6      1210                  CALL    SETSCRLL
E5E6 CD 0F E2      1220 HLP1             CALL    PTAD
E5E9 E5            1230                  PUSH    H
E5EA D5            1240                  PUSH    D
E5EB 0E 10         1250                  MVI     C,16
```

```
Addr B1 B2 B3 B4 E Line Label          Opcd    Operand

E5ED 7E             1260 HLP2           MOV     A,H
E5EE CD 2F E7       1270                CALL    PT2S
E5F1 23             1280                INX     H
E5F2 0D             1290                DCR     C
E5F3 C2 ED E5       1300                JNZ     HLP2
E5F6 D1             1310                POP     D
E5F7 E1             1320                POP     H
E5F8 0E 0F          1330                MVI     C,15
E5FA CD DA E0       1340                CALL    SPCE
E5FD CD DA E0       1350                CALL    SPCE
E600 CD 3C E5       1360                CALL    WDMP2
E603 FA E3 E5       1370                JM      HLP1-3
E606 C9             1380                RET
E607                1390 * CHECK TO SET SCROLL POINT
E607 3A DE FF       1400 SETSCRLL       LDA     WIDTH
E60A 3D             1410                DCR     A
E60B 32 DE FF       1420                STA     WIDTH
E60E 20 07          1430                JRNZ    CTSCRL
E610 01 50 F0       1440                LXI     B,PAGE+50H  ;2ND LINE
E613 ED 43 DF FF    1450                SBCD    TOSCN       ;SCROLL POINT
E617 C9             1460 CTSCRL         RET
E618                1470 *
E618                1480 *.PROGRAM MEMORY
E618 CD D1 E4       1490 PROGRAM        CALL    PTSTNG
E61B 50 52 4F 47    1500                DTH     'PROGRAM '
E61F 52 41 4D A0
E623 CD BD E0       1510                CALL    AHEX        ;ADDR IN HL
E626 ED 53 E1 FF    1520                SDED    TCURPOS
E62A CD 9B E5       1530                CALL    HOMEC       ;PRINT "ADDR"
E62D CD AC E5       1540                CALL    HEXRULER
E630 CD 33 E3       1560                CALL    TVIDEO
E633 AF             1570                XRA     A
E634 32 DE FF       1580                STA     WIDTH
E637 CD A1 E6       1590                CALL    PRT1LINE    ;PRINT LINE CONT
E63A CD 2F E1       1600 POLLOOP        CALL    ESCAPE
E63D CD ED E0       1610                CALL    HEX
E640 2A E1 FF       1620                LHLD    TCURPOS
E643 30 1A          1630                JRNC    MODMEM
E645                1640 * CONTROL CODE TABLE
E645 FE 20          1650                CPI     ' '
E647 28 46          1660                JRZ     CSRT
E649 FE 08          1670                CPI     8
E64B 28 45          1680                JRZ     CSLT
E64D FE 12          1690                CPI     'R'-64
E64F 28 39          1700                JRZ     CSDN
E651 FE 15          1710                CPI     'U'-64
E653 28 2F          1720                JRZ     CSUP
E655 FE 17          1730                CPI     'W'-64
E657 28 39          1740                JRZ     CSLT
E659 FE 1A          1750                CPI     'Z'-64
E65B 28 32          1760                JRZ     CSRT
E65D 18 DB          1770                JR      POLLOOP
E65F                1780 * MODIFY A MEMORY LOCATION
E65F 2A E1 FF       1790 MODMEM         LHLD    TCURPOS
E662 4F             1800                MOV     C,A
E663 3A DE FF       1810                LDA     WIDTH
```

| Addr | B1 | B2 | B3 | B4 | E | Line | Label | Opcd | Operand |
|------|----|----|----|----|---|------|-------|------|---------|
| E666 | A7 | | | | | 1820 | | ANA | A |
| E667 | 7E | | | | | 1830 | | MOV | A,M |
| E668 | 28 | 0D | | | | 1840 | | JRZ | LSNIBL |
| E66A | E6 | F0 | | | | 1850 | | ANI | 0F0H |
| E66C | B1 | | | | | 1860 | | ORA | C |
| E66D | 77 | | | | | 1870 | REMEM | MOV | M,A |
| E66E | 3A | DE | FF | | | 1880 | | LDA | WIDTH |
| E671 | EE | 01 | | | | 1890 | | XRI | 1 |
| E673 | 20 | 1F | | | | 1900 | | JRNZ | RTRTN+1 |
| E675 | 18 | 18 | | | | 1910 | | JR | CSRT |
| E677 | 17 | | | | | 1920 | LSNIBL | RAL | |
| E678 | 17 | | | | | 1930 | | RAL | |
| E679 | 17 | | | | | 1940 | | RAL | |
| E67A | 17 | | | | | 1950 | | RAL | |
| E67B | E6 | F0 | | | | 1960 | | ANI | 0F0H |
| E67D | B1 | | | | | 1970 | | ORA | C |
| E67E | 0F | | | | | 1980 | | RRC | |
| E67F | 0F | | | | | 1990 | | RRC | |
| E680 | 0F | | | | | 2000 | | RRC | |
| E681 | 0F | | | | | 2010 | | RRC | |
| E682 | 18 | E9 | | | | 2020 | | JR | REMEM |
| E684 | | | | | | 2030 | * MOVE UP ONE LINE | | |
| E684 | 11 | F0 | FF | | | 2040 | CSUP | LXI | D,-16 |
| E687 | 19 | | | | | 2050 | | DAD | D |
| E688 | 18 | 09 | | | | 2060 | | JR | RTRTN |
| E68A | | | | | | 2070 | * MOVE DOWN ONE LINE | | |
| E68A | 11 | 10 | 00 | | | 2080 | CSDN | LXI | D,16 |
| E68D | 18 | F8 | | | | 2090 | | JR | CSUP+3 |
| E68F | | | | | | 2100 | * MOVE RIGHT ONE SPACE | | |
| E68F | 23 | | | | | 2110 | CSRT | INX | H |
| E690 | 18 | 01 | | | | 2120 | | JR | RTRTN |
| E692 | | | | | | 2130 | * MOVE LEFT ONE SPACE | | |
| E692 | 2B | | | | | 2140 | CSLT | DCX | H |
| E693 | | | | | | 2150 | * | | |
| E693 | AF | | | | | 2160 | RTRTN | XRA | A |
| E694 | 32 | DE | FF | | | 2170 | | STA | WIDTH |
| E697 | 22 | E1 | FF | | | 2180 | | SHLD | TCURPOS |
| E69A | 3E | 15 | | | | 2190 | UPAROW | MVI | A,'U'-64 |
| E69C | CD | 8A | E3 | | | 2200 | | CALL | VIDEO |
| E69F | 18 | 96 | | | | 2210 | | JR | POLLOOP-3 |
| E6A1 | | | | | | 2220 | * PRINT A LINE CONTAINING ((H)) | | |
| E6A1 | 2A | E1 | FF | | | 2230 | PRT1LINE | LHLD | TCURPOS |
| E6A4 | E5 | | | | | 2240 | | PUSH | H |
| E6A5 | D1 | | | | | 2250 | | POP | D |
| E6A6 | 7D | | | | | 2260 | | MOV | A,L |
| E6A7 | F6 | 0F | | | | 2270 | | ORI | 0FH |
| E6A9 | 5F | | | | | 2280 | | MOV | E,A |
| E6AA | E6 | F0 | | | | 2290 | | ANI | 0F0H |
| E6AC | 6F | | | | | 2300 | | MOV | L,A |
| E6AD | CD | E6 | E5 | | | 2310 | | CALL | HLP1 |
| E6B0 | | | | | | 2320 | * NOW PUT CURSOR WHERE IT GOES | | |
| E6B0 | CD | 6F | E4 | | | 2330 | | CALL | LIFTCURS |
| E6B3 | 2A | E1 | FF | | | 2340 | | LHLD | TCURPOS |
| E6B6 | 7D | | | | | 2350 | | MOV | A,L |
| E6B7 | E6 | 0F | | | | 2360 | | ANI | 0FH |
| E6B9 | 6F | | | | | 2370 | | MOV | L,A |

```
Addr B1 B2 B3 B4 E Line Label          Opcd    Operand

E6BA 3E 05          2380          MVI    A,5
E6BC 2D             2390 PLOP1    DCR    L
E6BD FA C4 E6       2400          JM     PGCONT
E6C0 C6 03          2410          ADI    3
E6C2 18 F8          2420          JR     PLOP1
E6C4 6F             2430 PGCONT   MOV    L,A
E6C5 3A DE FF       2440          LDA    WIDTH
E6C8 85             2450          ADD    L
E6C9                2460  * A = 5+3*L+W
E6C9 32 DB FF       2470          STA    CURPOS
E6CC C3 6F E4       2480          JMP    LIFTCURS
E6CF                2490  *
E6CF                2500  *
E6CF                2510  * DISPLAY REGISTERS
E6CF CD D1 E4       2520 DREGS    CALL   PTSTNG
E6D2 52 45 47 49    2530          DTH    'REGISTERS'
E6D6 53 54 45 52
E6DA D3
E6DB                2540  * DUMP REGISTERS AFTER ENTRY FROM RST 7
E6DB E3             2550 DUMPREGS XTHL
E6DC F5             2560          PUSH   PSW
E6DD CD 35 E7       2570          CALL   DISPREGS
E6E0 2B             2580          DCX    H          ;GET BREAK ADD
E6E1 CD 0F E2       2590          CALL   PTAD
E6E4 E1             2600          POP    H
E6E5 C5             2610          PUSH   B
E6E6 CD 8A E7       2620          CALL   PRTFLGS
E6E9 C1             2630          POP    B
E6EA CD 12 E2       2640          CALL   PTAD+3     ;PRINT AF
E6ED E1             2650          POP    H
E6EE 22 E3 FF       2660          SHLD   HLTEMP
E6F1 CD AB E7       2670          CALL   PTHREE     ;PRINT B D H
E6F4 DD E5          2680          PUSH   IX
E6F6 E1             2690          POP    H
E6F7 CD 12 E2       2700          CALL   PTAD+3     ;PRINT IX
E6FA FD E5          2710          PUSH   IY
E6FC E1             2720          POP    H
E6FD CD 12 E2       2730          CALL   PTAD+3     ;PRINT IY
E700 21 00 00       2740          LXI    H,0
E703 39             2750          DAD    SP
E704 22 E5 FF       2760          SHLD   SPTEMP
E707 CD 12 E2       2770          CALL   PTAD+3     ;PRINT SP
E70A 08             2780          EXAF
E70B F5             2790          PUSH   PSW
E70C E1             2800          POP    H
E70D CD 12 E2       2810          CALL   PTAD+3
E710 D9             2820          EXX
E711 CD AB E7       2830          CALL   PTHREE
E714 D9             2840          EXX
E715 0A             2850          LDAX   B
E716 CD 2F E7       2860          CALL   PT2S
E719 1A             2870          LDAX   D
E71A CD 2F E7       2880          CALL   PT2S
E71D 2A E3 FF       2890          LHLD   HLTEMP
E720 7E             2900          MOV    A,M
E721 CD 2F E7       2910          CALL   PT2S
```

```
Addr B1 B2 B3 B4 E Line Label            Opcd    Operand

E724 2A E5 FF        2920                LHLD    SPTEMP
E727 F9              2930                SPHL
E728 E1              2940                POP     H
E729 CD 12 E2        2950                CALL    PTAD+3
E72C C3 46 E0        2960                JMP     CLRBRK          ;CLEAR BREAKPOINT
E72F                 2970 *
E72F CD 26 E2        2980 PT2S           CALL    PT2             ;PRINT 2 CHARS
E732 C3 DA E0        2990                JMP     SPCE            ;PRINT SPACE
E735                 3000 * DISPLAY REGISTER HEADER ON SCREEN
E735 CD CE E4        3010 DISPREGS       CALL    RPTSTNG
E738 14              3020                DB      'T'-64
E739 41 44 44 52     3040                DT      'ADDR FLAGS  AF   BC   DE'
E73D 20 46 4C 41
E741 47 53 20 20
E745 41 46 20 20
E749 20 42 43 20
E74D 20 20 44 45
E751 20 20 20 48     3050                DT      '   HL   IX   IY   SP '
E755 4C 20 20 20
E759 49 53 20 20
E75D 20 49 59 20
E761 20 20 53 50
E765 20
E766 20 20 41 46     3060                DT      '  AF'
E76A 27              3070                DB      27H              ' ;'
E76B 20 20 42 43     3080                DT      '  BC'
E76F 27              3090                DB      27H
E770 20 20 44 45     3100                DT      '  DE'
E774 27              3110                DB      27H
E775 20 20 48 4C     3120                DT      '  HL'
E779 27              3130                DB      27H
E77A 20 40 42 20     3140                DT      ' @B @D @H @SP '
E77E 40 44 20 40
E782 48 20 40 53
E786 50 20
E788 94              3150                DB      'T'+64
E789 C9              3160                RET
E78A                 3170 *
E78A                 3180 * PRINT FLAGS
E78A 01 5A 40        3190 PRTFLGS        LXI     B,405AH          ;Z
E78D CD BA E7        3200                CALL    MASKFLG
E790 01 43 01        3210                LXI     B,143H           ;C
E793 CD BA E7        3220                CALL    MASKFLG
E796 01 4D 80        3230                LXI     B,804DH          ;M
E799 CD BA E7        3240                CALL    MASKFLG
E79C 01 45 04        3250                LXI     B,445H           ;E
E79F CD BA E7        3260                CALL    MASKFLG
E7A2 01 48 10        3270                LXI     B,1043H          ;H
E7A5 CD BA E7        3280                CALL    MASKFLG
E7A8 C3 DA E0        3290                JMP     SPCE
E7AB                 3300 *
E7AB                 3310 * PRINT BC DE HL IN ORDER
E7AB E5              3320 PTHREE         PUSH    H
E7AC C5              3330                PUSH    B
E7AD E1              3340                POP     H
E7AE CD 12 E2        3350                CALL    PTAD+3
```

```
Addr B1 B2 B3 B4 E Line Label        Opcd    Operand

E7B1 D5                 3360                  PUSH    D
E7B2 E1                 3370                  POP     H
E7B3 CD 12 E2           3380                  CALL    PTAD+3
E7B6 E1                 3390                  POP     H
E7B7 C3 12 E2           3400                  JMP     PTAD+3
E7BA                    3410  *
E7BA 7D                 3420  MASKFLG         MOV     A,L
E7BB A0                 3430                  ANA     B
E7BC 3E 20              3440                  MVI     A,20H
E7BE CA 3A E3           3450                  JZ      VIDEO
E7C1 79                 3460                  MOV     A,C
E7C2 C3 3A E3           3470                  JMP     VIDEO
E7C5                    3480  *
E7C5                    3490  * SET BREAKPOINT
E7C5 CD D1 E4           3500  SETBRK          CALL    PTSTNG
E7C8 42 52 45 41        3510                  DTH     'BREAK AT '
E7CC 4B 20 41 54
E7D0 A0
E7D1 CD 3D E0           3520                  CALL    AHEX
E7D4 1A                 3530                  LDAX    D
E7D5 32 E9 FF           3540                  STA     BRKCODE
E7D8 ED 53 E7 FF        3550                  SDED    BKPTLOC
E7DC 3E FF              3560                  MVI     A,0FFH          ;RST 7
E7DE 12                 3570                  STAX    D
E7DF C9                 3580                  RET
E7E0                    3590  *
E7E0                    3600  * EXTERNAL COMMUNICATIONS
E7E0 CD D1 E4           3610  EXTCOM          CALL    PTSTNG
E7E3 45 58 54 20        3620                  DTH     'EXT COM '
E7E7 43 4F 4D A0
E7EB DB 05              3630  RECEIVE         IN      5
E7ED E6 02              3640                  ANI     2
E7EF 23 05              3650                  JRZ     NEXCHR
E7F1 DB 04              3660                  IN      4
E7F3 CD 3A E3           3670                  CALL    VIDEO
E7F6 CD 2F E1           3680  NEXCHR          CALL    ESCAPE
E7F9 28 F0              3690                  JRZ     RECEIVE
E7FB D3 04              3700                  OUT     4
E7FD 18 EC              3710                  JR      RECEIVE
E7FF                    3720  *
E7FF                    3730  * TEMPORARY STORAGE LOCATIONS FOR REGISTERS,ETC.
E7FF                    3740                  ORG     TCURPOS+2
FFE3 00 00              3750  HLTEMP          DW      0
FFE5 00 00              3760  SPTEMP          DW      0
FFE7 E9 FF              3770  BKPTLOC         DW      BRKCODE         ;BREAKPT LOCATION
FFE9 00                 3780  BRKCODE         DB      0               ;CODE AT BREAKPT
FFEA 00                 3790  XYFLAG          DB      0               ;CURSOR XY FLAG
```