VECTOR 5.0
EXECUTIVE

User's Manual

VECTOR 5.0 EXECUTIVE

PROGRAM

USERS MANUAL

Revision A

January 29, 1982

P/N 7100-0250-00-00

Disclaimer

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

Revision Numbers

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

TABLE OF CONTENTS

GENERAL DESCRIPTION

The Version 5.3 executive is a complete systems executive, designed to support the new Vector Extended CP/M Operating System. It also drives the Flashwriter II (80 X 24) video display board, and the Vector Graphic serial and parallel keyboards. Thus it is recommended for use with the Vector Mindless Terminal.

Because of the nature of the relationship between Extended CP/M and the 5.0 Executive, it is strongly suggested that any user writing conventional machine language programs use the facilities present in the operating system rather than those present in the Executive. It is further suggested that the user does not implement input and output commands directly to hardware devices. See the BIOS section of the Extended CP/M manual for information on how to interface to most I/O devices (including the keyboard and screen of the Mindless Terminal). See the BDOS section of the Extended CP/M manual to find out how to interface to the Disk Drive(s) in your system.

The 5.0 Executive differs so significantly from previous versions of the Extended Systems Monitor that it was given a different name. The changes made were necessary in order to accomodate the single/multiple user features of the Vector Operating System. In addition, disk boot driver routines have been included. Previously these were incorporated into a separate disk boot PROM.

This program includes an extensive command executive, a compactly written program designed to facilitate manipulation and display of memory data. The "prompt" which indicates that the Executive is waiting for operator entry is "Exec>".

If you are operating a terminal in a Multi-User system and are not quite sure what you are doing, be aware that Executive commands may have undesirable effects on other users of your system. It is suggested that you gain experience when there are no other active users on the system.

There are 23 commands which are entered as a single letter followed by up to four hexadecimal data fields. After each field is entered, a space is automatically output as a prompt. Either upper or lower case alpha characters may be used, but lower case characters will be converted to upper case, and any non-hex characters will be ignored. Allowable hex characters are 0-9, A-F. Address fields are four digits long; other fields are two digits long.

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 by depressing the space bar and continued by pressing the space bar again. The ESCape key will abort a display or command entry.

The 5.0 Executive is located at address E000H - EBFFH in Vector Graphic systems. The physical implementation of this program may vary according to the system involved. In current systems as shipped from the factory, the Executive occupies the lower three quarters of the address space on a 2732 EPROM (or equivalent). The upper quarter of the address space on that EPROM is not used. The ZCB board in these systems has been modified to not respond to any memory address in the EC00H to EFFFH range. This allows these addresses to be used by other memory boards in the system.

HEXADECIMAL NUMBERS

The hexadecimal number system may seem confusing if you are not familiar with it, but 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). An explanation of hexadecimal and other number systems used in microcomputers may be found in virtually any introductory microcomputer book.

HEX NUMBER	DECIMAL VALUE	JARGON	BINARY BITS
0	0		1
1	1		1
2	2		2
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

COMMAND FORMAT

Exec>B - BOOT FLOPPY

Typing this command causes a jump to location E800H which is located in the disk boot section of the Executive. This will cause the disk operating system to be loaded into memory and transfer control to CP/M. This is designed to be used with a Vector system using the DualMode or FD controller board. The use of a Micropolis Disk Controller board is incompatible with this system.

Exec>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.

Exec>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.

Exec>E - EXTERNAL COMMUNICATIONS

The Executive 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.

Exec>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.

Exec>G <ADR1> - GO TO AND EXECUTE

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

Exec>H - DISPLAY MEMORY BANKS

This command displays the bank number of resident Video and Ram memory boards found in the system.

Exec>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.

Exec>J - COLD BOOT

This command first checks to see which operating system is present in the system and then jumps to F800H. This will perform a cold boot of the operating system.

Exec>R - 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 Executive 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 Executive at E000H will clear the breakpoint, as will pressing the RESET switch.

Exec>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.

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

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

Exec>N - NON-DESTRUCTIVE MEMORY TEST

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

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

The two hex digits "DATA" are loaded into the accumulator and the instruction "OUT PORT" is executed. This command is useful for checking out peripheral equipment. For example, if a printer is connected to I/O port 6, O 06 41 will cause an "A" to be printed since 41 is the hex ASCII code for "A". If there are other users on the system, be careful that you do not output to the port address of their memory boards as this may cause loss of data.

Exec>P <ADR1> - PROGRAM MEMORY

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

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

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

Exec>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.

Exec>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).

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

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

Exec>U - JUMP TO 0100H

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

Exec>W - WINCHESTER DRIVE BOOT

Typing this command will cause a jump to E802H which is in the Disk Boot section of the Executive and contains 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.

Exec>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 Executive.

Exec>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. Be careful with this command if there are other users on the system.

VIDEO DRIVER

Version 5.0 of the Executive contains an elaborate video driver. 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 Executive prompt, and any keyboard generated code will be echoed to the video driver. The following control codes are interpreted as special functions, while all others are ignored:

Decimal Value	Hex Value	Control Code	Description
2	2	(^B)	HOME THE CURSOR
4	4	(^D)	CLEAR THE SCREEN AND HOME CURSOR
5	5	(^E)	DISPLAY THE CODE IN B REGISTER
8	8	(^H)	DESTRUCTIVE BACKSPACE (also BACKSPACE key)
9	9	(^I)	TAB OVER TO THE NEXT 8 MULTIPLE (also TAB)
10	A	(^J)	LINEFEED (also LF Key)
13	D	(^M)	CARRIAGE RETURN (also RETURN key)
14	E	(^N)	TOGGLE CURSOR
16	10	(^P)	CLEAR TO END OF SCREEN
17	11	(^Q)	CLEAR TO END OF LINE
18	12	(^R)	CURSOR DOWN
20	14	(^T)	TOGGLE REVERSE VIDEO
21	15	(^U)	CURSOR UP
23	17	(^W)	CURSOR LEFT
24	18	(^X)	CLEAR TO START OF LINE
26	1A	(^Z)	CURSOR RIGHT
_7	1B	ESC	CURSOR XY POSITION LEAD-IN or TOP OF SCREEN LEAD-IN

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

In some cases it is desirable to print the symbol for a control code on the screen. This can be done in assembly language programs by putting the code for the symbol in the B register and calling the video driver with Control E (05) in A. Enter the following machine code at FC00H and execute it to demonstrate this feature: 06 01 3E 05 04 CD 09 E0 CD 0C E0 C3 02 FC

CURSOR X Y POSITIONING

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

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.

SETTING TOP OF SCREEN

The logical top of screen can be set by sending the appropriate codes (escape sequences) to the Executive program. To set the top of screen send: ESC DEL (line number) to the Video driver. The line number must be expressed in hexadecimal in the range of 0H to 16H (0 to 22 decimal).

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 Executive'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. It is essential that software utilize the Executive conversion routine for this reason.

```
0000                ;*************************************************
0000                ;                                                *
0000                ;       VECTOR EXECUTIVE - VERSION 5.0           *
0000                ;       02/01/82                                 *
0000                ;                                                *
0000                ;*************************************************
0000                ;
0000  E000 =  BASE        EQU     0E000H          ;ASSEMBLY ADDRESS
0000  E000 =  PR          EQU     0E000H          ;PROM/RAM ADDRESS
0000  0000 =  CONS        EQU     0               ;CONS STATUS PRT
0000  0001 =  COND        EQU     1               ;CONS DATA PORT
0000  001B =  ESCP        EQU     1BH
0000  0004 =  CURSCRN     EQU     4
0000  0040 =  RDA         EQU     40H             ;RECEIVE FLAG
0000  0040 =  PORT        EQU     40H             ;RAM ENABLE PORT
0000  0000 =  STPOL       EQU     0               ;STATUS POLARITY
0000  0100 =  TPA         EQU     0100H           ;CP/M USER AREA
0000  FB00 =  SPTR        EQU     0FB00H          ;STACK POINTER
0000  FFFD =  TEMPSPTR    EQU     0FFFDH          ;TEMPORARY STACK POINTER
0000  E800 =  FLBOOT      EQU     0E800H          ;FLOPPY BOOTSTRAP
0000  E802 =  MSBOOT      EQU     0E802H          ;MEGASTOR BOOTSTRAP
0000  FF10 =  DBUSY       EQU     0FF10H          ;CONTROLLER BUSY
0000  F800 =  SCREEN      EQU     PR+1000H        ;SCREEN LOCATION
0000  FFFE =  VIDEOFLAG   EQU     0FFFEH          ;VIDEOS BANKS
0000  FFFF =  RAMFLAG     EQU     0FFFFH          ;RAM BANKS
0000  F800 =  COLDSTART   EQU     0F800H          ;EXTENDED CP/M ENTRY POINT
0000                ;
0000                ;*************************************************
0000                ;                                                *
0000                ;       COMMANDS:                                *
0000                ;                                                *
0000                ;       A                                        *
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 PRINT SYSTEM STATUS                    *
0000                ;       I PP INPUT FROM PORT                     *
0000                ;       J CP/M COLD BOOT                         *
0000                ;       K LLLL SET A BREAKPOINT                  *
0000                ;       L JUMP TO LOW RAM AT 0                   *
0000                ;       M SSSS FFFF DDDD MOVE BLOCK              *
0000                ;       N NON DESTRUCTIVE MEMORY TEST            *
0000                ;       O PP DD OUTPUT TO PORT                   *
0000                ;       P LLLL PROGRAM MEMORY                    *
0000                ;       Q SSSS FFFF COMPUTE CHECKSUM             *
0000                ;       R DUMP Z-80 REGISTERS                    *
0000                ;       S SSSS FFFF DD SEARCH FOR SINGLE BYTE    *
0000                ;       T SSSS FFFF TEST MEMORY                  *
0000                ;       U JUMP TO USER AREA AT 100H              *
0000                ;       V                                        *
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                ;
0000                        ORG     BASE
E000                ;
E000                ;       JUMP TABLE OF ENTRY POINTS
E000                ;
E000  C320E0    BEGIN:      JMP     INIT            ;INITIALIZE ALL
E003  C30FE2    GETSTAT:    JMP     KEYSTAT         ;FETCH KEYBOARD STATUS
E006  C314E2    GETDATA:    JMP     KEYDATA         ;FETCH KEYBOARD DATA (NO CONVERT)
E009  C318E4    PUTDATA:    JMP     VIDEO           ;OUTPUT TO SCREEN
E00C  C302E2    GETCONV:    JMP     ESCAPE          ;GET AND CONVERT DATA
E00F  C319E2    CONVDATA:   JMP     CONVERT         ;CHARACTER CONVERSION
E012  C307E0    WARMENTRY:  JMP     START           ;WARM ENTRY POINT
E015                ;
E015                ;       TABLE OF COMMANDS FOR USART
E015                ;
E015  00000040  INITABLE    DB      0,0,0,40H,0CEH,27H
E019  CE27
E01B                ;
E01B                ;       INIALIZATION SUBROUTINE FOR USARTS (PORT ADDRESS IN A)
E01B                ;
E01B  060G      INILOOP     MVI     B,6             ;NO OF COMMAND BYTES
E01D  2115E0                LXI     H,INITABLE      ;START OF BYTE TABLE
E020  4F                    MOV     C,A
E021  EDA3      OUTLOOP     OUTI                    ;OUTPUT A BYTE
E023  E3                    XTHL                    ;DELAY FOR 6 MHZ.
E024  E3                    XTHL
E025  20FA                  JRNZ    OUTLOOP         ;SEND NEXT BYTE
E027  C9                    RET
E028                ;
E028  F3        INIT        DI                      ;MAKE SURE INT DOES
E029  31FDFF                LXI     SP,TEMPSPTR     ;SET STACK TO TOP OF COMMON
E02C                ;
E02C                ;       INITIALIZE USARTS AT PORTS 0,3,5,7
E02C                ;
E02C  3E07                  MVI     A,7             ;HIGHEST PORT ADDRESS
E02E  CD1BE0    INITPORT:   CALL    INILOOP         ;INITIALIZES PORT
E031  D602                  SUI     2               ;DECREMENT TO NEXT ADDRESS
E033  FE01                  CPI     1               ;CHECK FOR END
E035  20F7                  JRNZ    INITPORT
E037  D340                  OUT     PORT            ;ENABLE BANK 1
E039  3D                    DCR     A
E03A  3210FF                STA     DBUSY           ;CLEAR CONTROLLER FLAG
E03D                ;
E03D                ;       CHECK MEMORY BANKS AND PATCH RST 5
E03D                ;       INITIALIZE PARAMETERS/FLAGS IN ALL BANKS
E03D                ;       CHECK VIDEO BANKS AND WRITE PROMPT
E03D                ;
E03D  0E40      USERCHECK:  MVI     C,PORT
E03F  0601                  MVI     B,1             ;START WITH BANK 1
E041  11FEFF                LXI     D,VIDEOFLAG     ;POINT TO VIDEOFLAG
E044  12                    STAX    D               ;ZERO VIDEOFLAG
E045  13                    INX     D
E046  12                    STAX    D               ;ZERO RAMFLAG
E047  ED41      USERCHECK10:OUTP    B               ;ENABLE BANK
E049  210000                LXI     H,0             ;CHECK LOW RAM
```

```
E04C 7E                 MOV     A,M             ;GET BYTE FROM MEMORY
E04D 35                 DCR     M               ;CHANGE RAM
E04E BE                 CMP     M               ;CHECK IF SAME
E04F 77                 MOV     M,A             ;RESTORE RAM
E050 2811               JRZ     USERCHECK20     ;SKIP IF NOT ENABLED
E052 22EAFB             SHLD    XYFLAG          ;ZEROS FLAGS (ORDER DEPENDENT)
E055 1A                 LDAX    D               ;GET RAM FLAG
E056 B0                 ORA     B               ;SET BANK BIT
E057 12                 STAX    D               ;SAVE NEW RAM FLAG
E058 3EC3               MVI     A,0C3H          ;JUMP
E05A 322800             STA     28H             ;RST 5
E05D 21D7E6             LXI     H,DUMPREGS      ;JUMP ADDRESS FOR RESTART
E060 222900             SHLD    29H             ;SAVE IN MEMORY AT RST 5
E063 2100F0     USERCHECK20:    LXI     H,SCREEN        ;CHECK VIDEO RAM
E066 7E                 MOV     A,M             ;GET BYTE FROM MEMORY
E067 35                 DCR     M               ;CHANGE RAM
E068 BE                 CMP     M               ;CHECK IF SAME
E069 77                 MOV     M,A             ;RESTORE RAM
E06A 2812               JRZ     USERCHECK30     ;SKIP IF NOT ENABLED
E06C 1B                 DCX     D               ;(DE) = VIDEO FLAG
E06D 1A                 LDAX    D
E06E B0                 ORA     B               ;SET BIT
E06F 12                 STAX    D               ;SAVE NEW VIDEO FLAG
E070 13                 INX     D               ;(DE) = RAMFLAG
E071 1A                 LDAX    D               ;GET RAM FLAGS
E072 A0                 ANA     B               ;MASK BANK WITH RAM BITS
E073 2809               JRZ     USERCHECK30     ;SKIP IF NOT BOTH VIDEO AND RAM
E075 D9                 EXX                     ;SAVE ALL REGS
E076 CDFEE0             CALL    SIGN.ON         ;DISPLAY SIGN-ON
E079 AF                 XRA     A               ;USE PORT 0
E07A CD18E0             CALL    INILOOP         ;INITIALIZES KEYBOARD PORT
E07D D9                 EXX                     ;RETRIEVE ALL REGS
E07E CB10       USERCHECK30:    RALR    B               ;ROTATE BIT UP
E080 30C5               JRNC    USERCHECK10     ;LOOPS THRU ALL BANKS
E082 CB10               RALR    B
E084 ED41               OUTP    B               ;TURN ON BANK 1
E086 3E0E               MVI     A,14
E088 CD10E4             CALL    VIDEO           ;TOGGLE CURSOR ON
E08B
E08B 2A67FB     CLRBRK  LHLD    BKPTLOC         ;HL = ADDRESS OF BREAKPOINT
E08E 116FFB             LXI     D,BRKCODE       ;DE = INSTRUCTION STORAGE
E091 ED5367FB           SDED    BKPTLOC         ;SAVE DE AS BREAKPOINT ADDRESS
E095 1A                 LDAX    D               ;GET INSTRUCTION
E096 77                 MOV     M,A             ;PUT BACK IN MEMORY
E097                    ;
E097 31D0FB     START   LXI     SP,STTB         ;INITIALIZE STACK
E09A 2100F0             LXI     H,SCREEN        ;INITIALIZE TOP OF SCREEN
E09D 22DFFB             SHLD    TOSCN
E0A0 CD82E5             CALL    PROMPT
E0A3 CD02E2     KEYPOL  CALL    ESCAPE          ;READ KEYBOARD
E0A6 28FB               JRZ     KEYPOL          ;LOOP IF NO INPUT
E0A8 E65F               ANI     5FH             ;CHANGE TO UPPER CASE
E0AA 2197E0             LXI     H,START         ;PUSH RETURN ADDRESS
E0AD E5                 PUSH    H
E0AE FE04               CPI     'D'-64          ;CHECK FOR CURSON
E0B0 CC10E4             CZ      VIDEO           ;ECHO CLEARSON
E0B3 FE41               CPI     'A'             ;RANGE CHECK
```



```
E0B5 D8                 RC                      ;TOO SMALL
E0B6 FE5B               CPI     05BH
E0B8 D0                 RNC                     ;TOO LARGE
E0B9 21CAE0             LXI     H,CMDTB         ;HL = START OF COMMAND TABLE
E0BC F5                 PUSH    PSW             ;SAVE CHARACTER
E0BD D641               SUI     'A'             ;SUBTRACT ASCII BIAS
E0BF 87                 ADD     A               ;DOUBLE FOR 2 BYTE ADDRESS
E0C0 5F                 MOV     E,A
E0C1 1600               MVI     D,0             ;DE = INDEX
E0C3 19                 DAD     D               ;(HL) = JUMP ADDRESS
E0C4 5E                 MOV     E,M             ;GET LSBYTE
E0C5 23                 INX     H
E0C6 56                 MOV     D,M             ;GET MSBYTE
E0C7 EB                 XCHG                    ;HL = JUMP ADDRESS
E0C8 F1                 POP     PSW
E0C9 E9                 PCHL                    ;AWAY WE GO
E0CA            ;
E0CA            ;       COMMAND TABLE
E0CA            ;
E0CA 97E0       CMDTB   DW      START           ;A
E0CC 00E8               DW      FLBOOT          ;B
E0CE 83E3               DW      COMPR           ;C
E0D0 C8E5               DW      HEXRUL          ;D
E0D2 DCE7               DW      EXTCOM          ;E
E0D4 A2E3               DW      FIND            ;F
E0D6 5DE1               DW      EXEC            ;G
E0D8 68E1               DW      SYSTAT          ;H
E0DA EEE3               DW      PINPT           ;I
E0DC 56E2               DW      COLD            ;J
E0DE C1E7               DW      SETBRK          ;K
E0E0 14E3               DW      LORAM           ;L
E0E2 39E3               DW      MOVED           ;M
E0E4 61E3               DW      NEMT            ;N
E0E6 FDE1               DW      POUTP           ;O
E0E8 14E6               DW      PROGRAM         ;P
E0EA 30E2               DW      CHKSM           ;Q
E0EC C8E6               DW      DREGS           ;R
E0EE AFE3               DW      SINCH           ;S
E0F0 87E2               DW      TMEM            ;T
E0F2 00E3               DW      USER            ;U
E0F4 97E0               DW      START           ;V
E0F6 02E8               DW      MSBOOT          ;W
E0F8 97E0               DW      START           ;X
E0FA 77E2               DW      ECHO            ;Y
E0FC 28E3               DW      ZEINCM          ;Z
E0FE            ;
E0FE 3E04       SIGN.ON:        MVI     A,4
E100 CD10E4             CALL    VIDEO
E103
E103 21A3F1             LXI     H,HORIZ*5+SCREEN+19
E106 11072A             LXI     D,2A07H
E109 3E7F               MVI     A,07FH
E10B CD4CE1             CALL    DRBOX
E10E
E10E 21F5F1             LXI     H,HORIZ*6+SCREEN+21
E111 110526             LXI     D,2605H
E114 3EA0               MVI     A,0A0H
```

```
E116 CD4CE1                     CALL    DRBOX
E119
E119 214FF2                     LXI     H,HORIZ*7+SCREEN+23
E11C 110322                     LXI     D,2203H
E11F 3E20                       MVI     A,' '
E121 CD4CE1                     CALL    DRBOX
E124
E124 CD77E5                     CALL    PTSTNG
E127 1B1A08                     DB      ESCP,26,8
E12A 56454354                   DT      'VECTOR GRAPHIC EXECUTIVE 5.0b'
E12E 4F522047
E132 52415048
E136 494320A5
E13A 58454355
E13E 54495645
E142 20352E30
E146 62
E147 1B0000E                    DB      ESCP,0,13,14+80H
E14B C9                         RET
E14C
E14C E5         DRBOX:          PUSH    H
E14D 42                         MOV     B,D
E14E 77         DRBOX10:        MOV     M,A
E14F 23                         INX     H
E150 10FC                       DJNZ    DRBOX10
E152 E1                         POP     H
E153 D5                         PUSH    D
E154 115000                     LXI     D,80
E157 19                         DAD     D
E158 D1                         POP     D
E159 1D                         DCR     E
E15A 20F0                       JRNZ    DRBOX
E15C C9                         RET
E15D            ;
E15D            ;** EXECUTE THE PROGRAM AT THE ADDRESS ***
E15D            ;
E15D CD77E5     EXEC            CALL    PTSTNG
E160 474F20                     DTH     'GO '
E163 CD92E1                     CALL    AHEX            ;READ ADD FROM KB
E166 EB                         XCHG
E167 E9                         PCHL                    ;JUMP TO IT
E168            ;
E168            ;       DISPLAY SYSTEM HARDWARE STATUS
E168            ;
E168 CD74E5     SYSTAT:         CALL    RPTSTNG
E16B 5241CD                     DTH     'RAM'
E16E 3AFFFF                     LDA     RAMFLAG         ;POINT TO LOW RAM FLAG
E171 CD7FE1                     CALL    CHECKUSER       ;CHECK AND PRINT ACTIVE
E174 CD74E5                     CALL    RPTSTNG         ;BANKS
E177 56494445                   DTH     'VIDEO'
E17B CF
E17C 3AFEFF                     LDA     VIDEOFLAG       ;NOW CHECK VIDEO FLAG
E17F 0608       CHECKUSER:      MVI     B,8             ;TEST 8 BANKS
E181 1F         CHECKUSER10:    RAR                     ;ROTATE BIT INTO CARRY
E182 3008                       JRNC    CHECKUSER20     ;SKIP IF NOT SET
E184 4F                         MOV     C,A             ;SAVE BANK BYTE
E185 CDA0E1                     CALL    SPCE            ;PRINT SPACE
```



```
E188 3E09                       MVI     A,9             ;FORM BANK NUMBER
E18A 90                         SUB     B
E18B CDEAE2                     CALL    PT2             ;PRINT BANK NUMBER
E18E 79                         MOV     A,C             ;RETRIEVE BANK BYTE
E18F 10F0       CHECKUSER20:    DJNZ    CHECKUSER10     ;LOOP THRU ALL BANKS
E191 C9                         RET
E192            ;
E192            ;** CONVERT UP TO 4 HEX DIGITS TO BIN
E192            ;
E192 0E04       AHEX            MVI     C,4             ;COUNT OF 4 DIGITS
E194 210000     AHE0            LXI     H,0             ;16 BIT ZERO
E197 CD02E2     AHE1            CALL    ESCAPE
E19A FE20                       CPI     ' '             ;SPACE?
E19C 281D                       JRZ     SPCOVR
E19E CDC8E1                     CALL    HEX             ;CHECK VALUE
E1A1 38F4                       JRC     AHE1
E1A3 29                         DAD     H               ;MULT H*16
E1A4 29                         DAD     H
E1A5 29                         DAD     H
E1A6 29                         DAD     H
E1A7 85                         ADD     L
E1A8 6F                         MOV     L,A
E1A9 0D                         DCR     C               ;4 DIGITS?
E1AA 20EB                       JRNZ    AHE1            ;KEEP READING
E1AC EB                         XCHG
E1AD 3E20       SPCE            MVI     A,' '           ;PRINT SPACE
E1AF C310E4     PTCN            JMP     VIDEO
E1B2 3E0D       CRLF            MVI     A,0DH           ;PRINT CR
E1B4 CD10E4                     CALL    VIDEO
E1B7 3E0A                       MVI     A,0AH
E1B9 18F4                       JR      PTCN
E1BB            ;
E1BB CD10E4     SPCOVR          CALL    VIDEO
E1BE 18EC                       JR      SPCE-1
E1C0            ;
E1C0            ; CHECK FOR HEX VALUE, CONVERT
E1C0 FE30       HEX             CPI     '0'             ;<0
E1C2 D8                         RC
E1C3 FE3A                       CPI     '9'+1           ;>9
E1C5 3009                       JRC     NUM
E1C7 E65F                       ANI     5FH             ;UPPER & LOWER CASE
E1C9 FE41                       CPI     'A'             ;<A
E1CB D8                         RC
E1CC FE47                       CPI     'G'             ;>F
E1CE 3F                         CMC
E1CF D8                         RC
E1D0 CD10E4     NUM             CALL    VIDEO
E1D3 D630                       SUI     48              ;ASCII BIAS
E1D5 FE0A                       CPI     10              ;DIGIT 0-10
E1D7 3802                       JRC     ALFA
E1D9 D607                       SUI     7               ;ALPHA BIAS
E1DB A7         ALFA            ANA     A               ;CLEAR CY
E1DC C9                         RET                     ;WITH CY CLEAR
E1DD            ;
E1DD            ; READ 2 DIGITS FROM THE CONSOLE
E1DD 0E02       AHE2            MVI     C,2
E1DF 18D3                       JR      AHE0
```

```
01E1                    ;
01E1                    ; SHORT ROUTINE TO SAVE CODE
01E1 CD92E1     TAREX   CALL    AREX
01E4 18AC               JR      AREX
01E6                    ;
01E6                    ;** READ FROM CONSOLE TO REG A ***
01E6                    ;
01E6 CD02E2     RDCN    CALL    ESCAPE          ;READ KEYBOARD
01E9 28FB               JRZ     RDCN
01EB FE60               CPI     60H
01ED 38C0               JRC     PTCN
01EF E65F               ANI     5FH
01F1 18BC               JR      PTCN
01F3                    ;
01F3 CD02E2     PAUSE   CALL    ESCAPE
01F6 FE20               CPI     ' '
01F8 C8                 RNZ
01F9 CD02E2     PLOOP   CALL    ESCAPE
01FC FE20               CPI     ' '
01FE C2F9E1             JNZ     PLOOP
0201 C9                 RET
0202                    ;
0202 CD0FE2     ESCAPE  CALL    KEYSTAT
0205 C8                 RZ
0206 CD17E2             CALL    DATACONV
0209 FE1B               CPI     ESCP            ;ESCAPE
020B CA97E0             JZ      START
020E C9                 RET
020F                    ;
020F DB00       KEYSTAT IN      CONS
0211 E640               ANI     RDA
0213 C9                 RET
0214                    ;
0214                    ; KEYBOARD DATA FETCH
0214                    ;
0214 DB01       KEYDATA:        IN      COND            ;KEYBOARD DATA
0216 C9                 RET
0217                    ;
0217                    ; KEYBOARD FETCH AND CODE CONVERSION
0217                    ;
0217 DB01       DATACONV:       IN      COND
0219 E5         CONVERT:        PUSH    H
021A C5                 PUSH    B
021B 010500             LXI     B,TABLEND-KTABL/2
021E 2131E2             LXI     H,KTABL
0221 EDA1       LOOP    CCI                     ;COMPARE TABLE
0223 2006               JRZ     FND
0225 23                 INX     H
0226 EA21E2             JPE     LOOP            ;CONT LOOKING
0229 1801               JR      NFND
022B 7E         FND     MOV     A,M             ;NEW CODE
022C E67F       NFND    ANI     7FH             ;MASK DOWN
022E C1                 POP     B
022F E1                 POP     H
0230 C9                 RET
0231                    ;
0231                    ; THIS TABLE CAN BE EXTENDED IF DESIRED
```



```
0231 015D       KTABL   DD      0E15DH          ;]
0233 F158               DD      0F158H          ;[
0235 A17E               DD      0A17EH          ;~
0237 B15C               DD      0B15CH          ;\
0239 6015               DD      06015H          ;CURSOR UP
023B   023B =   TABLEND EQU     $
023B                    ;       ORG     KTABL+30        ;ROOM FOR 15 CONVS
023B                    ;
023B                    ; CHECKSUM ROUTINE
023B CD77E5     CHKSM   CALL    PTSTNG
023E 43484B53           DTH     'CHKSUM '
0242 5540A0
0245 CDE1E1             CALL    TAREX
0248 0600               MVI     B,0
024A 7E         CHKSMLP MOV     A,M
024B 80                 ADD     B
024C 47                 MOV     B,A
024D CD03E3             CALL    BHP
0250 20FB               JRNZ    CHKSMLP
0252 78                 MOV     A,B
0253 C3EAE2             JMP     PT2
0256                    ;
0256                    ;       CP/M COLD BOOT
0256                    ;
0256 CD77E5     COLD    CALL    PTSTNG
0259 434F4C44           DTH     'COLD BOOT'
025D 20424F4F
0261 D4
0262 3A00F8             LDA     COLDSTART
0265 FEC3               CPI     0C3H
0267 CA00F8             JZ      COLDSTART
026A CD74E5             CALL    RPTSTNG
026D 4E4F2053           DTH     'NO SYSTEM'
0271 59535445
0275 CD
0276 C9                 RET
0277                    ;
0277                    ; KEYBOARD ECHO ROUTINE
0277                    ;
0277 CD77E5     ECHO    CALL    PTSTNG
027A 4543484F           DTH     'ECHO '
027E A0
027F CD02E2     ECOLP   CALL    ESCAPE          ;LOOK AT KEYBOARD
0282 C410E4             CNZ     VIDEO           ;PRINT IF KEYPRESS
0285 18F8               JR      ECOLP           ;CONTINUE LOOPING
0287                    ;
0287                    ;** MEMORY TEST ROUTINE ***
0287                    ;
0287 CD77E5     TMEM    CALL    PTSTNG
028A 54455354           DTH     'TEST '
028E A0
028F CDE1E1             CALL    TAREX           ;READ ADDRESSES
0292 015A5A             LXI     B,5A5AH         ;INI B,C
0295 CDC1E2     CYCL    CALL    RNDM
0298 C5                 PUSH    B               ;KEEP ALL REGS
0299 E5                 PUSH    H
029A D5                 PUSH    D
```

```
E29B CDC1E2     TLOP    CALL    RNDM
E29E 78                 MOV     M,B             ;WRITE IN MEM
E29F CD03E3             CALL    BMP
E2A2 C29BE2             JNZ     TLOP            ;REPEAT LOOP
E2A5 D1                 POP     D
E2A6 E1                 POP     H               ;RESTORE ORIG
E2A7 C1                 POP     B               ;VALUES OF
E2A8 E5                 PUSH    H
E2A9 D5                 PUSH    D
E2AA CDC1E2     RLOP    CALL    RNDM            ;GEN NEW SEQ
E2AD 7E                 MOV     A,M             ;READ MEM
E2AE B8                 CMP     B               ;COMP MEM
E2AF C4E1E2             CNZ     ERR             ;CALL ERROR RTN
E2B2 CD03E3             CALL    BMP
E2B5 C2AAE2             JNZ     RLOP
E2B8 D1                 POP     D
E2B9 E1                 POP     H
E2BA 3E2E               MVI     A,'.'
E2BC CD10E4             CALL    VIDEO
E2BF 18D4               JR      CYCL
E2C1            ;** THIS ROUTINE GENERATES RANDOM NOS ***
E2C1 CDF3E1     RNDM    CALL    PAUSE
E2C4 78                 MOV     A,B             ;LOOK AT B
E2C5 E6B4               ANI     0B4H            ;MASK BITS
E2C7 A7                 ANA     A               ;CLEAR CY
E2C8 EACCE2             JPE     PEVE            ;JUMP IF EVEN
E2CB 37                 STC
E2CC 79         PEVE    MOV     A,C             ;LOOK AT C
E2CD 17                 RAL                     ;ROTATE CY IN
E2CE 4F                 MOV     C,A             ;RESTORE C
E2CF 78                 MOV     A,B             ;LOOK AT B
E2D0 17                 RAL                     ;ROTATE CY IN
E2D1 47                 MOV     B,A             ;RESTORE B
E2D2 C9                 RET                     ;RETURN W NEW B,C
E2D3            ;
E2D3            ;** ERROR PRINT OUT ROUTINE
E2D3            ;
E2D3 CDB2E1     PTAD    CALL    CRLF            ;PRINT CR,LF
E2D6 CDF3E1             CALL    PAUSE
E2D9 7C                 MOV     A,H             ;PRINT
E2DA CDEAE2             CALL    PT2             ;ASCII
E2DD 7D                 MOV     A,L             ;CODES
E2DE C32BE7             JMP     PT2S            ;FOR ADDRESS
E2E1            ;
E2E1 F5         ERR     PUSH    PSW             ;SAVE ACC
E2E2 CDD3E2             CALL    PTAD            ;PRINT ADD.
E2E5 78                 MOV     A,B             ;DATA
E2E6 CD2BE7             CALL    PT2S            ;WRITTEN
E2E9 F1                 POP     PSW             ;DATA READ
E2EA F5         PT2     PUSH    PSW
E2EB CDF1E2             CALL    BINH
E2EE F1                 POP     PSW
E2EF 18B4               JR      BINL
E2F1 1F         BINH    RAR                     ;SHIFT HIT 4 BITS
E2F2 1F                 RAR
E2F3 1F                 RAR
E2F4 1F                 RAR
```



```
E2F5 E60F       BINL    ANI     0FH             ;LOW 4 BITS
E2F7 C630               ADI     48              ;ASCII BIAS
E2F9 FE3A               CPI     58              ;DIGIT 0-9
E2FB DA10E4             JC      VIDEO
E2FE C607               ADI     7               ;DIGIT A-F
E300 C310E4             JMP     VIDEO
E303            ;
E303            ; COMPARE ADDRESSES AND INCREMENT H
E303 7B         BMP     MOV     A,E
E304 95                 SUB     L
E305 2002               JNZ     GOON
E307 7A                 MOV     A,D
E308 9C                 SBB     H
E309 23         GOON    INX     H
E30A C9                 RET
E30B            ;
E30B            ; JUMP TO USER RAM
E30B CD77E5     USER    CALL    PTSTNG
E30E 5450C1             DTH     'TPA'
E311 C30001             JMP     TPA
E314            ;
E314            ; JUMP TO RAM AT B
E314 CD77E5     LORAM   CALL    PTSTNG
E317 4C4F2052           DTH     'LO RAM'
E31B 41CD
E31D C30000             JMP     0
E320            ;
E320            ; ZERO OR FILL MEMORY WITH A CONSTANT
E320 CD77E5     ZERO    CALL    PTSTNG
E323 46494C4C           DTH     'FILL '
E327 A0
E328 CDE1E1             CALL    TWOEX           ;READ ADDRESSES
E32B E5                 PUSH    H               ;SAVE H
E32C CDDDE1             CALL    ARE2            ;READ 2 DIGITS
E32F EB                 XCHG
E330 E3                 XTHL                    ;RESTORE H,L
E331 C1                 POP     B
E332 71         ZLOOP   MOV     M,C             ;WRITE DATA MEM
E333 CD03E3             CALL    BMP             ;COMP ADD, INCR H
E336 C8                 RZ                      ;RETURN IF DONE
E337 18F9               JR      ZLOOP           ;CONTINUE TIL DONE
E339            ;
E339 47         MOVEB   MOV     B,A             ;SAVE CODE
E33A CD77E5             CALL    PTSTNG
E33D 4D4F5645           DTH     'MOVE '
E341 A0
E342 CDE1E1     MOVENTR CALL    TWOEX           ;READ ADDRESSES
E345 E5                 PUSH    H
E346 CD92E1             CALL    AHEX
E349 EB                 XCHG
E34A E3                 XTHL                    ;BACK TO NORMAL
E34B 4E         MLOOP   MOV     C,M
E34C E3                 XTHL
E34D 78                 MOV     A,B
E34E FE4D               CPI     'M'
E350 2804               JRZ     NEXTH
E352 7E                 MOV     A,M
```

```
E353 E3                         XTHL
E354 77                         MOV     M,A
E355 E3                         XTHL
E356 71             NEXCH       MOV     M,C
E357 23                         INX     H
E358 E3                         XTHL
E359 CD03E3                     CALL    BMP
E35C CA97E0                     JZ      START
E35F 18EA                       JR      MLOOP
E361                ; NON DESTRUCTIVE MEMORY TEST
E361 CD77E5         MEMT        CALL    PTSTNG
E364 4D454D20                   DTH     'MEM TOP'
E368 544F50
E368 210000                     LXI     H,0             ;START AT ZERO
E36E 4E             NDLOP       MOV     C,M
E36F 06FF                       MVI     B,0FFH
E371 70                         MOV     M,B
E372 7E                         MOV     A,M
E373 B8                         CMP     B
E374 C27CE3                     JNZ     ERRJP           ;PRINT ERROR
E377 0600                       MVI     B,0
E379 70                         MOV     M,B
E37A 7E                         MOV     A,M
E37B B8                         CMP     B
E37C C2E1E2         ERRJP       JNZ     ERR
E37F 71                         MOV     M,C
E380 23                         INX     H
E381 18EB                       JR      NDLOP
E383                ; COMPARE TWO BLOCKS OF MEMORY
E383 CD77E5         COMPR       CALL    PTSTNG
E386 434D5020                   DTH     'CMP '
E38A CDE1E1                     CALL    TAHEX
E38D E5                         PUSH    H
E38E CD92E1                     CALL    AHEX
E391 EB                         XCHG
E392 7E             VMLOP       MOV     A,M
E393 23                         INX     H
E394 E3                         XTHL
E395 BE                         CMP     M
E396 46                         MOV     B,M
E397 C4E1E2                     CNZ     ERR
E39A CD03E3                     CALL    BMP
E39D E3                         XTHL
E39E 20F2                       JRNZ    VMLOP
E3A0 F1                         POP     PSW
E3A1 C9                         RET
E3A2                ; SEARCH FOR SPECIFIC CODES
E3A2 F5             FIND        PUSH    PSW
E3A3 CD77E5                     CALL    PTSTNG
E3A6 46494E44                   DTH     'FIND-2 '
E3AA 2D3220
E3AD 1809                       JR      SRCHENT
E3AF F5             SRCH        PUSH    PSW
E3B0 CD77E5                     CALL    PTSTNG
E3B3 46494E44                   DTH     'FIND-1 '
E3B7 2D3120
E3BA CDE1E1         SRCHENT     CALL    TAHEX
```



```
E3BD E5                         PUSH    H               ;SAVE H
E3BE CDDDE1                     CALL    AHE2            ;READ 2 DIGITS
E3C1 EB                         XCHG                    ;H=CODE,D=F
E3C2 45                         MOV     B,L             ;PUT CODE IN B
E3C3 E1                         POP     H               ;RESTORE H
E3C4 F1                         POP     PSW
E3C5 FE53                       CPI     'S'
E3C7 F5                         PUSH    PSW
E3C8 2807                       JRZ     CONT
E3CA E5                         PUSH    H
E3CB CDDDE1                     CALL    AHE2            ;READ 2 DIGITS
E3CE EB                         XCHG
E3CF 4D                         MOV     C,L
E3D0 E1                         POP     H
E3D1 7E             CONT        MOV     A,M             ;READ MEMORY
E3D2 B8                         CMP     B               ;COMPARE TO CODE
E3D3 2812                       JRNZ    SKP             ;SKIP IF NO COMP
E3D5 F1                         POP     PSW             ;FETCH CONTROL
E3D6 FE53                       CPI     'S'
E3D8 F5                         PUSH    PSW
E3D9 2806                       JRZ     OBCP
E3DB 23                         INX     H
E3DC 7E                         MOV     A,M
E3DD 2B                         DCX     H
E3DE B9                         CMP     C
E3DF 2006                       JRNZ    SKP
E3E1 23             OBCP        INX     H
E3E2 7E                         MOV     A,M             ;READ NEXT BYTE
E3E3 2B                         DCX     H               ;DECR ADDRESS
E3E4 CDE1E2                     CALL    ERR             ;PRINT CODES
E3E7 CD03E3         SKP         CALL    BMP             ;CHECK IF DONE
E3EA 20E5                       JRNZ    CONT            ;BACK FOR MORE
E3EC F1                         POP     PSW
E3ED C9                         RET
E3EE                ;
E3EE                ; INPUT DATA FROM A PORT
E3EE CD77E5         PINPT       CALL    PTSTNG
E3F1 494E5A0                    DTH     'IN '
E3F4 CDDDE1                     CALL    AHE2            ;READ 2 DIGITS
E3F7 4B                         MOV     C,E
E3F8 ED78                       INP     A
E3FA C3EAE2                     JMP     PT2
E3FD                ;
E3FD                ; OUTPUT TO A PORT
E3FD CD77E5         POUTP       CALL    PTSTNG
E400 4F5554A0                   DTH     'OUT '
E404 CDDDE1                     CALL    AHE2            ;READ 2 DIGITS
E407 CDDDE1                     CALL    AHE2            ;READ 2 DIGITS
E40A 4D                         MOV     C,L
E40B ED59                       OUTP    E
E40D C9                         RET
E40E                ;
```

```
E40E                    ;
E40E                    ;*****************************************
E40E                    ;                                         *
E40E                    ;     VIDEO DRIVER FOR  PLAINWRITER II    *
E40E                    ;                                         *
E40E                    ;*****************************************
E40E                    ;
E40E                    ;*****************************************
E40E                    ;                                         *
E40E                    ; CONTROL CODE COMMANDS:                  *
E40E                    ;    (B) HOME CURSOR                      *
E40E                    ;    (D) CLEAR SCREEN                     *
E40E                    ;    (E) PRINT CONTROL CODE               *
E40E                    ;    (H) BACKSPACE                        *
E40E                    ;    (I) TAB                              *
E40E                    ;    (J) LINEFEED                         *
E40E                    ;    (M) CARRIAGE RETURN                  *
E40E                    ;    (N) NO CURSOR                        *
E40E                    ;    (P) CLEAR TO END OF SCREEN           *
E40E                    ;    (Q) CLEAR TO END OF LINE             *
E40E                    ;    (R) CURSOR DOWN                      *
E40E                    ;    (T) TOGGLE REVERSE VIDEO             *
E40E                    ;    (U) CURSOR UP                        *
E40E                    ;    (V) CURSOR LEFT                      *
E40E                    ;    (X) CLEAR TO START OF LINE           *
E40E                    ;    (Z) CURSOR RIGHT                     *
E40E                    ;   ESC XY POSITION LEAD-IN               *
E40E                    ;                                         *
E40E                    ;*****************************************
E40E                    ;
E40E                    ; VIDEO BOARD PARAMETERS
E40E   0050 =   HORIZ   EQU     80              ;NO. OF CHARACTERS
E40E   0018 =   VERT    EQU     24              ;NO. OF LINES
E40E                    ;
E40E 3E14       TVIDEO  MVI     A,'T'-64        ;TOGGLE VIDEO
E410                    ;
E410 F5         VIDEO   PUSH    PSW
E411 C5                 PUSH    B
E412 D5                 PUSH    D
E413 E5                 PUSH    H
E414 E67F               ANI     07FH            ;MASK OFF MSBIT
E416 4F                 MOV     C,A             ;PUT CHAR IN C
E417 CD6CE4     DISPL   CALL    LIFTCURS        ;ERASE CURSOR
E41A 3AEAFB             LDA     XYFLAG          ;GET POSITIONING FLAG
E41D A7                 ANA     A               ;CHECK IF TRUE
E41E 280A               JRZ     NOXY            ;SKIP IF FALSE
E420 3D                 DCR     A               ;DECREMENT FLAG
E421 32EAFB             STA     XYFLAG          ;SAVE NEW VALUE
E424 CA3CE5             JZ      YPOS            ;Y IF SECOND VALUE
E427 C32AE5             JMP     XPOS            ;ELSE X
E42A 79         NOXY    MOV     A,C             ;RECOVER CHARACTER
E42B FE20               CPI     ' '             ;PRINTING CODE?
E42D F26124             JP      PRINT
E430 FE1C               CPI     PCL-TABL        ;TOO LARGE?
E432 F2CEE4             JP      RET
E435 E5                 PUSH    H               ;CURSOR IN MEMORY
E436 2144E4             LXI     H,TABL          ;TABLE START
```



```
E439 5F                 MOV     E,A
E43A 1600               MVI     D,0
E43C 19                 DAD     D
E43D 5E                 MOV     E,M
E43E 2160E4             LXI     H,PCL
E441 19                 DAD     D
E442 E3                 XTHL                    ;RECOVER H
E443 C9                 RET                     ;EXECUTE ROUTINE
E444                ; CONTROL CHARACTER JUMP TABLE
E444 6E         TABL    DB      RET-PCL         ;@
E445 6E                 DB      RET-PCL         ;A
E446 63                 DB      HOME-PCL        ;B HOME CURSOR
E447 6E                 DB      RET-PCL         ;C
E448 68                 DB      FORM-PCL        ;D CLEAR SCREEN
E449 08                 DB      ICL-PCL         ;E PRT CONTROL
E44A 6E                 DB      RET-PCL         ;F
E44B 6E                 DB      RET-PCL         ;G
E44C 42                 DB      IBACKSP-PCL     ;H BACKSPACE
E44D 59                 DB      TAB-PCL         ;I TAB OVER
E44E 12                 DB      LINF-PCL        ;J LINE FEED
E44F 6E                 DB      RET-PCL         ;K
E450 6E                 DB      RET-PCL         ;L
E451 6A                 DB      CRET-PCL        ;M CARRIAGE RET
E452 71                 DB      RETN1-PCL       ;N NO CURSOR
E453 6E                 DB      RET-PCL         ;O
E454 9E                 DB      CLEND-PCL       ;P CLR SCN TO END
E455 A3                 DB      CLLINE-PCL      ;Q CLR LINE TO END
E456 12                 DB      LINF-PCL        ;R CURSOR DOWN
E457 6E                 DB      RET-PCL         ;S
E458 76                 DB      TVIDF-PCL       ;T TOGGLE VIDEO
E459 80                 DB      CURUP-PCL       ;U CURSOR UP
E45A 6E                 DB      RET-PCL         ;V
E45B 58                 DB      BACKSP-PCL      ;W CURSOR LEFT
E45C B8                 DB      CLSTRT-PCL      ;X CLR START OF LN
E45D 6E                 DB      RET-PCL         ;Y
E45E 86                 DB      EOL-PCL         ;Z CURSOR RIGHT
E45F C3                 DB      LEDIN-PCL       ;[ ESC=XY LEADIN
E460                    ;
E460                    ; PRINT CODE IN B REGARDLESS
E460 48         PCL     MOV     C,B
E461                    ; PRINT THE CHARACTER ON THE SCREEN
E461 3AD8FB     PRINT   LDA     VFL
E464 A9                 XRA     C
E465 77                 MOV     M,A
E466                    ; EOL CHECKS THE CURS POS FOR END OF LINE
E466 3AD0FB     EOL     LDA     CURPOS
E469 3C                 INR     A
E46A FE50               CPI     HORIZ
E46C D05D               JRC     TABRET
E46E AF                 XRA     A
E46F 32D0FB             STA     CURPOS
E472                    ; MOVE IN 1 LINE
E472 3AD1FB     LINF    LDA     LINENO
E475 FE17               CPI     VERT-1
E477 2023               JRNZ    NOSCRL
E479                    ; SCROLL UP ONE LINE
E479 2150FB     SCROLL  LXI     H,HORIZ
```

```
E47C ED58DFFB                   LDED    TOSCN
E480 19                         DAD     D
E481 EDA0       SCRL            LDI
E483 EDA0                       LDI
E485 7C                         MOV     A,H
E486 FEF7                       CPI     HORIZ*VERT+SCREEN/256
E488 20F7                       JRNZ    SCRL
E48A 7D                         MOV     A,L
E48B FE00                       CPI     HORIZ*VERT+SCREEN&0FFH
E48D 20F2                       JRNZ    SCRL
E48F 3ADCFB                     LDA     LINENO
E492            ; ERASE BOTTOM LINE
E492 EB         EBOTL           XCHG
E493 0650                       MVI     B,HORIZ
E495 3620       ELOP            MVI     M,' '
E497 23                         INX     H
E498 05                         DCR     B
E499 20FA                       JRNZ    ELOP
E49B 3D                         DCR     A
E49C 3C         NOSCRL          INR     A
E49D 32DCFB                     STA     LINENO
E4A0 182C                       JR      RET
E4A2            ;
E4A2            ; ERASE BEFORE BACKSPACING
E4A2 3620       DBACKSP         MVI     M,20H
E4A4 3AD8FB                     LDA     CURPOS
E4A7 A7                         ANA     A
E4A8 2824                       JRZ     RET
E4AA 3D                         DCR     A
E4AB 2B                         DCX     H
E4AC 3620                       MVI     M,' '
E4AE 181B                       JR      TABRET
E4B0            ; MOVE THE CURSOR BACK
E4B0 3AD8FB     BACKSP          LDA     CURPOS
E4B3 3D                         DCR     A
E4B4 F2CBE4                     JP      TABRET
E4B7 1811                       JR      CRET
E4B9            ; TAB OVER TO THE NEXT 8 MULTIPLE
E4B9 3AD8FB     TAB             LDA     CURPOS
E4BC F607                       ORI     7
E4BE 18A9                       JR      EOL+3
E4C0            ; CLEAR THE SCREEN AND HOME UP
E4C0 CD18E5     FORM            CALL    CLEAR
E4C3 AF         HOME            XRA     A
E4C4 32DCFB                     STA     LINENO
E4C7 32D0FB                     STA     VFL             ;CLR VID FLAG
E4CA            ; CARRIAGE RETURN
E4CA AF         CRET            XRA     A
E4CB 32D8FB     TABRET          STA     CURPOS
E4CE            ; RETURN TO THE CALLING ROUTINE
E4CE CDCCE4     RET             CALL    LIFTCURS
E4D1 E1                         POP     H
E4D2 D1                         POP     D
E4D3 C1                         POP     B
E4D4 F1                         POP     PSW
E4D5 C9                         RET
E4D6 3AD0FB     TVLINE          LDA     VFL
```



```
E4D9 EE80                       XRI     80H
E4DB 32D0FB                     STA     VFL
E4DE 18EE                       JR      RET
E4E0            ;
E4E0            ; MOVE THE CURSOR UP
E4E0 3ADCFB     CURSUP          LDA     LINENO
E4E3 A7                         ANA     A
E4E4 28EB                       JRZ     RET
E4E6 3D                         DCR     A
E4E7 32DCFB     STORLN          STA     LINENO
E4EA 18E2                       JR      RET
E4EC            ;
E4EC            ; CALCULATE MEM ADD FROM CURSOR POSITION
E4EC            ;
E4EC 3ADCFB     LIFTCURS        LDA     LINENO          ;GET CURRENT LINE
E4EF CD67E5                     CALL    CALCULATELINE   ;GET MEMORY ADDRESS FOR LINE
E4F2 ED58D8FB                   LDED    CURPOS          ;GET POSITION ON LINE
E4F6 1600                       MVI     D,0             ;DE = X POSITION
E4F8 19                         DAD     D               ;HL = CURSOR POSITION
E4F9 7E                         MOV     A,M             ;GET CHARACTER
E4FA EE80                       XRI     80H             ;REVERSE VIDEO
E4FC 77                         MOV     M,A             ;PUT CHAR BACK
E4FD C9                         RET
E4FE            ;
E4FE            ; CLEAR TO END OF SCREEN
E4FE            ;
E4FE CD1AE5     CLEND           CALL    WRSPC
E501 18CB                       JR      RET
E503            ; CLEAR TO END OF LINE
E503 3AD8FB     CLLINE          LDA     CURPOS
E506 3620                       MVI     M,' '
E508 23                         INX     H
E509 3C                         INR     A
E50A FE50                       CPI     HORIZ
E50C 20F8                       JRNZ    CLLINE+3
E50E 18BE                       JR      RET
E510            ; CLEAR THE SCREEN
E510 2100F0     CLEAR           LXI     H,SCREEN
E513 22DFFB                     SHLD    TOSCN           ;SET TOP OF SCREEN
E516 AF                         XRA     A
E517 32EAFB                     STA     XYFLAG          ;ZERO XY LEAD-IN FLAG
E51A 3620       WRSPC           MVI     M,' '
E51C 23                         INX     H
E51D 7C                         MOV     A,H
E51E FEF8                       CPI     SCREEN+2048/256
E520 20F8                       JRNZ    WRSPC
E522 C9                         RET
E523            ;
E523            ; PROCESS LEAD IN CODE
E523 3E02       LEDIN           MVI     A,2
E525 32EAFB                     STA     XYFLAG
E528 18A4                       JR      RET
E52A            ; SET X AND Y CURSOR POSITIONS
E52A 79         XPOS            MOV     A,C             ;GET X POSITION
E52B FE7F                       CPI     7FH             ;CHECK FOR 07FH
E52D 2005                       JRNZ    XPOS10          ;SKIP IF NOT
E52F 32D1FB                     STA     TOPFLAG         ;SET TOPFLAG
```

```
E532 189A                       JR      RET             ;RETURN
E534 FE50       XPOS10:         CPI     HORIZ
E536 3893                       JRC     TABRET
E538 3E4F                       MVI     A,HORIZ-1       ;SET TO MAX
E53A 188F                       JR      TABRET
E53C            ;
E53C 3A3BFB     YPOS            LDA     TOPFLAG         ;GET TOP SCREEN SET FLAG
E53F B7                         ORA     A               ;CHECK IF TRUE
E540 2012                       JRNZ    TOPSCREEN       ;SET TOP OF SCREEN
E542 79                         MOV     A,C
E543 FE18                       CPI     VERT
E545 3BA8                       JRC     STORLN
E547 3E17                       MVI     A,VERT-1        ;SET TO MAX
E549 189C                       JR      STORLN
E54B            ;
E54B AF         CLSTRT          XRA     A
E54C 32D3FB                     STA     CURPOS
E54F CDECE4                     CALL    LIFTCURS
E552 18AF                       JR      CLLINE
E554            ;
E554            ;       SET TOP OF SCREEN TO LINE SPECIFIED BY A
E554            ;
E554 AF         TOPSCREEN:      XRA     A
E555 32EBFB                     STA     TOPFLAG         ;RESET FLAG
E558 79                         MOV     A,C             ;GET LINE NUMBER
E559 FE17                       CPI     VERT-1          ;RANGE CHECK
E55B D2CEE4                     JNC     RET             ;RETURN IF LINE >= VERT
E55E CD67E5                     CALL    CALCULATELINE
E561 22D3FB                     SHLD    TOSCN           ;SAVE NEW TOP OF SCREEN
E564 C3CEE4                     JMP     RET
E567            ;
E567            ;       LINE ADDRESS CALCULATION (LINE IN A) RETURNS ADDRESS IN HL
E567            ;       OPTIMIZED AT BOTTOM
E567            ;
E567 2180F7     CALCULATELINE:  LXI     H,HORIZ*VERT*SCREEN     ;ASSUME LAST LINE
E56A 1180FF                     LXI     D,-HORIZ                ;DE = -(CHAR/LINE)
E56D 3C         CALCLOOP:       INR     A                       ;INCR LINE NUMBER
E56E 19                         DAD     D                       ;SUBTRACT 1 LINE
E56F FE18                       CPI     VERT                    ;CHECK IF DONE
E571 20FA                       JRNZ    CALCLOOP                ;LOOP IF NOT
E573 C9                         RET                             ;RETURN WITH ADDRESS
E574            ;
E574            ;       PRINT A STRING
E574            ;
E574 CD82E1     RPTSTNG         CALL    CRLF            ;CRLF FIRST
E577 E3         PTSTNG          XTHL                    ;GET STRING POINTER
E578 7E                         MOV     A,M             ;GET CHAR
E579 23                         INX     H               ;INCR POINTER
E57A E3                         XTHL                    ;PUT POINTER BACK
E57B A7                         ANA     A               ;ZERO CARRY
E57C CD10E4                     CALL    VIDEO           ;PRINT IT
E57F F8                         RM                      ;RETURN IF NEGATIVE
E580 18F5                       JR      PTSTNG          ;REPEAT IF NOT
E582            ;
E582 CD74E5     PROMPT          CALL    RPTSTNG
E585 45786563                   DFB     'Exec> '
E589 BEA0
```



```
E588 C9                         RET
E58C            ;
E58C 7E         WDMP2           MOV     A,M
E58D 47                         MOV     B,A
E58E 3E05                       MVI     A,'E'-64
E590 CD10E4                     CALL    VIDEO
E593 CD81E3                     CALL    DMP
E596 C8                         RZ
E597 0D                         DCR     C
E598 F8                         RM
E599 18F1                       JR      WDMP2
E59B            ; HOME CURSOR, PRINT "ADDR"
E59B CD74E5     HOMEC           CALL    RPTSTNG
E59E 14                         DB      'T'-64
E59F 41444452                   DFB     'ADDR '
E5A3 A0
E5A4 0600                       MVI     B,0
E5A6 3E18                       MVI     A,24
E5A8 32DEFB                     STA     WIDTH
E5AB C9                         RET
E5AC            ; MAKE A RULER FOR HEX DUMP
E5AC 78         HEXRULER        MOV     A,B
E5AD FE10                       CPI     16
E5AF 2806                       JRZ     HEXRCT
E5B1 CD2DE7                     CALL    PT2S
E5B4 04                         INR     B
E5B5 18F5                       JR      HEXRULER
E5B7            ; EXTEND FOR ASCII
E5B7 CDA1E1     HEXRCT          CALL    SPCE
E5BA CDA1E1                     CALL    SPCE
E5BD 0600                       MVI     B,0
E5BF 78         HEXRLP          MOV     A,B
E5C0 FE10                       CPI     16
E5C2 C8                         RZ
E5C3 E60F                       ANI     0FH
E5C5 CDF5E2                     CALL    BINL
E5C8 04                         INR     B
E5C9 18F4                       JR      HEXRLP
E5CB            ; HEX DUMP ROUTINE
E5CB CD77E5     HEXRUL          CALL    PTSTNG
E5CE 44554D50                   DFB     'DUMP '
E5D2 A0
E5D3 CDE1E1                     CALL    TABEX
E5D6 CD9BE5                     CALL    HOMEC
E5D9 CDACE5                     CALL    HEXRULER
E5DC CD9EE4                     CALL    TVIDEO
E5DF CD03E6                     CALL    SETCHLL
E5E2 CD03E2     HLP1            CALL    PTAD
E5E5 E5                         PUSH    H
E5E6 D5                         PUSH    D
E5E7 0E10                       MVI     C,16
E5E9 7E         HLP2            MOV     A,M
E5EA CD2DE7                     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 CDADE1             CALL    SPCE
E5F9 CDADE1             CALL    SPCE
E5FC CDACE5             CALL    HEMP2
E5FF FADFE5             JM      HLP1-3
E602 C9                 RET
E603            ; CHECK TO SET SCROLL POINT
E603 3ADEFB     SETSCRLL        LDA     WIDTH
E606 3D                 DCR     A
E607 32DEFB             STA     WIDTH
E60A 2007               JNZ     CTSCRL
E60C 0150F0             LXI     B,SCREEN+50H    ;2ND LINE
E60F ED43DFFB           SBCD    TOSCN           ;SCROLL POINT
E613 C9         CTSCRL  RET
E614            ;
E614            ; PROGRAM MEMORY
E614 CD77E5     PROGRAM         CALL    PTSTNG
E617 50524F47                   DTH     'PROGRAM '
E61B 52414DA0
E61F CD92E1                     CALL    AHEX            ;ADDR IN HL
E622 ED53E1FB                   SDED    TCURPOS
E626 CD90E5                     CALL    HOMEC           ;PRINT "ADDR"
E629 CDACE5                     CALL    HEXRULER
E62C CD0CE4                     CALL    TVIDEO
E62F AF                         XRA     A
E630 32DEFB                     STA     WIDTH
E633 CD90E6                     CALL    PRTLINE         ;PRINT LINE CONT H
E636 CD42E2     POLLOOP         CALL    ESCAPE
E639 CDC0E1                     CALL    HEX
E63C 2AE1FB                     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 2819                       JRZ     CSLT
E655 FE1A                       CPI     'Z'-64
E657 2832                       JRZ     CSRT
E659 18DB                       JR      POLLOOP
E65B            ; MODIFY A MEMORY LOCATION
E65B 2AE1FB     MODMEM          LHLD    TCURPOS
E65E 4F                         MOV     C,A
E65F 3ADEFB                     LDA     WIDTH
E662 A7                         ANA     A
E663 7E                         MOV     A,M
E664 280D                       JRZ     LSNIBL
E666 E6F0                       ANI     0F0H
E668 B1                         ORA     C
E669 77         REMEM           MOV     M,A
E66A 3ADEFB                     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 32DEFB                     STA     WIDTH
E693 22E1FB                     SHLD    TCURPOS
E696 3E15       UPAROW          MVI     A,'U'-64
E698 CD10E4                     CALL    VIDEO
E69B 1896                       JR      POLLOOP-3
E69D            ; PRINT A LINE CONTAINING (H)
E69D 2AE1FB     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 CDECE4                     CALL    LEFTCURS
E6AF 2AE1FB                     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 FAC0E6                     JM      PGCONT
E6BC C603                       ADI     3
E6BE 18F8                       JR      PLOP1
E6C0 6F         PGCONT          MOV     L,A
E6C1 3ADEFB                     LDA     WIDTH
E6C4 85                         ADD     L
```

```
E6C5            ; A = 5+3*LIN
E6C5 3200FB             STA     CURPOS
E6C8 C3CCE4             JMP     LIFTCURS
E6CB            ;
E6CB            ;
E6CB            ; DISPLAY REGISTERS
E6CB CD77E5     DRGS    CALL    PTSTNG
E6CE 52454749           DFM     'REGISTERS'
E6D2 53544552
E6D6 03
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 CDD3E2             CALL    PTAD
E6E0 E1                 POP     H
E6E1 C5                 PUSH    B
E6E2 CDB6E7             CALL    PRTFLGS
E6E5 C1                 POP     B
E6E6 CDD6E2             CALL    PTAD+3          ;PRINT AF
E6E9 E1                 POP     H
E6EA 22E3FB             SHLD    HLTEMP
E6ED CDA7E7             CALL    PTHREE          ;PRINT B D H
E6F0 DDE5               PUSH    IX
E6F2 E1                 POP     H
E6F3 CDD6E2             CALL    PTAD+3          ;PRINT IX
E6F6 FDE5               PUSH    IY
E6F8 E1                 POP     H
E6F9 CDD6E2             CALL    PTAD+3          ;PRINT IY
E6FC 210000             LXI     H,0
E6FF 39                 DAD     SP
E700 22E5FB             SHLD    SPTEMP
E703 CDD6E2             CALL    PTAD+3          ;PRINT SP
E706 08                 EXAF
E707 F5                 PUSH    PSW
E708 E1                 POP     H
E709 CDD6E2             CALL    PTAD+3
E70C D9                 EXX
E70D CDA7E7             CALL    PTHREE
E710 D9                 EXX
E711 0A                 LDAX    B
E712 CD20E7             CALL    PT2S
E715 1A                 LDAX    D
E716 CD20E7             CALL    PT2S
E719 2AE3F9             LHLD    HLTEMP
E71C 7E                 MOV     A,M
E71D CD20E7             CALL    PT2S
E720 2AE5FB             LHLD    SPTEMP
E723 F9                 SPHL
E724 E1                 POP     H
E725 CDD6E2             CALL    PTAD+3
E728 C3BBE0             JMP     CLRBRK          ;CLEAR BREAKPOINT
E72B            ;
E72B CDEAE2     PT2S    CALL    PT2             ;PRINT 2 CHARS
E72E C3ADE1             JMP     SPCE            ;PRINT SPACE
E731            ; DISPLAY REGISTER HEADER ON SCREEN
```



```
E731 CD74E5     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
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 CD86E7             CALL    MASKFLG
E78C 014301             LXI     B,143H          ;C
E78F CD86E7             CALL    MASKFLG
E792 014D80             LXI     B,804DH         ;M
E795 CD86E7             CALL    MASKFLG
E798 014504             LXI     B,445H          ;E
E79B CD86E7             CALL    MASKFLG
E79E 014810             LXI     B,1048H         ;H
E7A1 CD86E7             CALL    MASKFLG
E7A4 C3ADE1             JMP     SPCE
E7A7            ;
E7A7            ; PRINT BC DE HL IN ORDER
E7A7 E5         PTHREE          PUSH    H
E7A8 C5                 PUSH    B
E7A9 E1                 POP     H
E7AA CDD6E2             CALL    PTAD+3
E7AD D5                 PUSH    D
E7AE E1                 POP     H
E7AF CDD6E2             CALL    PTAD+3
E7B2 E1                 POP     H
E7B3 C3D6E2             JMP     PTAD+3
E7B6            ;
E7B6 7D         MASKFLG         MOV     A,L
E7B7 A8                 XRA     B
E7B8 3E20               MVI     A,20H
E7BA CA10E4             JZ      VIDEO
```

```
E7BD 79                         MOV     A,C
E7BE C310E4                     JMP     VIDEO
E7C1                    ;
E7C1                    ;       SET BREAKPOINT
E7C1                    ;
E7C1 CD77E5             SETBRK  CALL    PTSTNG
E7C4 42524541                   DTH     'BREAK AT '
E7C8 4820415420
E7CC A0
E7CD CD92E1                     CALL    NXTEX
E7D0 1A                         LDAX    D
E7D1 32E9FB                     STA     BRKCODE
E7D4 ED53E7FB                   SDED    BKPTLOC
E7D8 3EEF                       MVI     A,0EFH          ;RESTART 5
E7DA 12                         STAX    D
E7DB C9                         RET
E7DC                    ;
E7DC                    ; EXTERNAL COMMUNICATIONS
E7DC CD77E5             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 CD10E4                     CALL    VIDEO
E7F2 CD92E2             NEXCHR  CALL    ESCAPE
E7F5 28F8                       JRZ     RECEIVE
E7F7 D304                       OUT     4
E7F9 18EC                       JR      RECEIVE
E7FB                            PRT     'PROGRAM LENGTH = ',$-BEGIN+1
E7FB                            ORG     BASE+7FFH
E7FF 50                 VERSION:DB      50H
E800                    ;
E800                    ;       CURSOR STORAGE LOCATIONS
E800                    ;
E800                            ORG     SPTR+0BH
FB0B                    CURPOS  DS      1               ;POS ON LINE
FB0C                    LINENO  DS      1               ;LINE NUMBER
FB0D                    VFL     DS      1               ;REVERSE VID FLAG
FB0E                    WIDTH   DS      1               ;PRINT WIDTH
FB0F                    TOPSCN  DS      2               ;TOP OF SCREEN
FB11                    TEMPPOS DS      2               ;TEMP POSITION
FB13                    ;
FB13                    ;       TEMPORARY STORAGE LOCATIONS FOR REGISTERS,ETC.
FB13                    ;
FB13                    BLTEMP  DS      2
FB15                    SPTEMP  DS      7
FB17                    BKPTLOC DS      2               ;BREAKPT LOCATION
FB19                    BRKCODE DS      1               ;CODE AT BREAKPT
FB1A                    XYFLAG  DS      1               ;CURSOR XY FLAG
FB1B                    TOPFLAG DS      1               ;NON-ZERO IF TOPSCREEN SET
```