Z-1 MONITOR

Cromemco
Specialists in computer peripherals
2432 Charleston Rd., Mountain View, CA 94043 • (415) 964-7400

CROMEMCO Z-1 MONITOR

1K Version

The Z-1 Monitor makes it possible to control computers which use the CROMEMCO ZPU™ from a terminal keyboard.

It includes executive commands to examine and change memory, make a binary or an ASCII dump of memory, move and compare blocks of memory, output a byte of data to any port, read and write binary paper tapes, and program 2708 and 2704 proms using the CROMEMCO BYTESAVER.

Transfer of control to a program in memory can be commanded from the keyboard with up to five breakpoints set and with the initial contents of the ZPU registers specified. When a breakpoint is encountered during execution, control is transferred back to the monitor and the contents of all 22 ZPU registers are stored and displayed. These register values can be examined and changed before execution of the program is resumed.

The Z-1 Monitor was designed to be flexible as well as powerful. For example, the monitor does not require the user to address a RAM board at a special place in memory for its own use. Rather, it finds the highest page of RAM active in the machine and places its stack and temporary storage area there. (The top 4AH or 74 bytes of this page should be reserved for system use.)

The monitor is also flexible in its command format. It will accept command words of any length, but it only looks at the first and last characters typed in. This allows the use of either longer expressions for their mnemonic value or shorter expressions for their brevity. For example, any of the following

```
DR
DISPLAYR
DSPR
```

causes the contents of all 22 user-registers to be displayed on the terminal.

Many of the commands apply to a range of memory. For example:

```
MOV 2408 240A B000
```

will move the contents of locations 2408 through 240A to B000 through B003. Another way to accomplish the same thing is by means of the swath-operator. Thus:

```
MOV 2408 S3 B000
```

will move to B000 a swath of 3 bytes starting at 2408.

The format is free-form with respect to spaces.

```
MOV 2408 S  3  B000
  MOV   2408S3   B000
MV 2408S3 B000
```

All have the same effect. Note that at least one space follows the command word and at least one space separates a pair of numeric operands such as '3' and 'B000'.

USE OF THE MONITOR

Set the power-on jump switch on the ZPU card to E (1110 binary). Whenever the computer is reset control will then immediately pass to the monitor.

If the ZPU is installed in the CROMEMCO Z-1™ computer, depress CARRIAGE RETURN from 2 to 3 times. This will set the UART on the serial interface card to the baud rate of the terminal being used.

When used with a serial interface card with baud rate fixed to that of the terminal, simply depress CARRIAGE RETURN twice. The monitor will respond

```
CROMEMCO MON1.0 C.1976
```

followed by a prompt ':'. The monitor is then ready to accept commands from the keyboard.

When entering an address as the operand of a command, only the last four digits typed in are retained. For example:

```
12345
```

is read as

```
2345.
```

Therefore, if a wrong digit is entered, continue typing until the last four digits are correct. There is a hazard with this, however. An extra 'F' inadvertently typed when '1FFF' is desired yields 'FFFF' instead.

When a two digit number, such as a data byte is entered, only the last two digits typed are retained.

If the monitor detects an error condition, the command is aborted, a '?' is printed followed by the prompt ':' for the next command.

Any command may be aborted from the keyboard either when the monitor is requesting further input, or during print-out, by depressing ESCAPE. For teletypes ESCAPE is the same as CONTROL-SHIFT-K. For most other terminals it is CONTROL SEMI-COLON.

Two cautions should be noted. When using the MOV or the PRGM commands, be careful not to overwrite the system stack and temporary storage area which resides in the upper 4AH or 72 bytes of active RAM installed in the machine.

When the computer is reset, the monitor is also reset. If a user program is executing at the time with breakpoints set, then these breakpoints will remain in the user program until removed by hand. (The normal procedure is for the monitor to restore the user program code whenever a breakpoint is executed.)

The monitor assumes that data transfer occurs on I/O port 1. Status flags are transmitted over input port 0. The Data Available Flag is on bit 6 of input port 0. The Transmitter Buffer Empty flag is on bit 7 of input port 0.

COMMANDS

1. DSPM (Start) (Stop) (CARRIAGE RETURN)

Display memory (DM is the short form) starting with location Start and ending with Stop. As with all 2-operand commands, the swath-operator may be used instead.
DSPM (Start) S (Swath width) (CR).

Example: :DSPM 100 S 3
0100: AB 34 7F

2. DSPR (CR)

Display registers (DR). The 22 registers are displayed with the following format:

```
00PC  00SP
0A0F  0B0C  0D0E  0H0L  0I0T  00IX  00IY
AAFF  BBCC  DDEE  HHLL
```

where the placement of a 1-byte register, such as A is indicated by "0A", of a 2-byte register, such as the program counter, PC, by "00PC", and of a primed register, such as A' by "AA". The T register, by the way, contains 1 if interrupts were enabled when the monitor was entered, and 0 otherwise.

3. GO (CR)

Resumes execution at the location contained in the user program counter, PC.

GO (Addr) (CR)

Begins execution at Addr.

4. GO/(bp1) (Bp2)...(CR)

Resumes execution at the location of PC with breakpoints set Bp1, Bp2, ...(Up to 5 breakpoints may be set.)

GO (Addr)/(Bp1)(Bp2)...(CR)

Begins execution at Addr with the indicated breakpoints set. All breakpoints are cleared upon re-entry of the monitor from a breakpoint.

5. MOV (Start) (Stop) (Destination) (CR)

Move (MV) the contents of memory beginning with Start and ending with Stop to Destination. After the move, the monitor verifies that source and destination are the same. This will result in a print-out of discrepancies which are not really errors after certain types of overlapping moves. However, this print-out can be terminated by depressing ESCAPE.

The MOV command can be used to fill a block of memory with a constant. For example, to enter zeros between locations 100 and 10D, use the SBSM command to enter 0 at location 100, and then move 100 through 10C to 101:

MOV 100 10C 101

Care should be taken not to overwrite the system stack which resides in the upper 4AH or 74 bytes of active RAM.

6. OUT (Data Byte) (Port)

Output (OT) Data Byte to Port. One use of this command is

to select banks on CROMEMCO memory boards. When the monitor is first entered on power-up or reset, it selects bank 0 and turns off all other memory banks.

Either a software output or a monitor output to port 40 hex serves to change the bank selection. To select bank n output a byte with bit n high.

Bank	Output byte
0	01
1	02
2	04
3	08
4	10
5	20
6	40
7	80

For example, the following command selects bank 4

```
OUT 10  40
```

7. PRGM (Source) (Source-end) (Destination)

Program (PM) from Source through Source-end into proms beginning at Destination.

If the length of the source is not a multiple of 400H (1024 decimal) or if the destination does not begin at a 400H boundary, the monitor will reject the command. (Multiples of 400H end in 000, 400, 800, or C00.)

8. READ (Start) (Stop) (CR)

Read (RD) input from paper tape reader or console and store in memory beginning at Start.

9. SUBM (Addr) (SPACE)

Substitute memory (SM). Displays the contents of Addr and outputs a DOT, '.', as a prompt for the substituted value. If no change is desired, type another '.'. Otherwise, enter the new value. The monitor accepts hex digits until it gets a delimiter, such as a SPACE or DOT, retaining the last 2 digits entered as the value. After it receives a delimiter, the monitor outputs the contents of the next sequential memory location with a DOT prompt. To terminate, depress ESCAPE.

10. SUBR (Register) (SPACE)

Where Register may be PC, SP, A, F, B, C, D, E, H, L, I, T (Interrupts-enabled state), X (IX), Y (IY), A', B', C', D', E', F', H', or L'.

Substitute register (SR). This works like the Substitute - memory command with two exceptions. (1) When substituting for the value of a 2-byte register, the monitor retains the last 4 digits before the delimiter, and (2) after accepting the substitution value for one register, it awaits entry of the ID of the next register.

11. VRFY (Start) (Stop) (Destination) (CR)

Verify (VY) that the block of memory between Start and Stop contains the same values as the block beginning at Destination. The address and contents are printed for each discrepancy found.

12. WRIT (Start) (Stop) (CR)

Write (WT) the contents of memory locations between Start

and Stop on the console or paper tape punch. This is useful for punching binary or ASCII paper tapes of the contents of memory, and for looking at the ASCII contents of memory on the console.

When punching a paper tape, it is often desirable to punch a series of nulls as a leader. This can be done by filling about 60 hex bytes preceding the desired memory block with zeros (see the MOV command) and writing out the whole thing beginning with the nulls. (Depress CARRIAGE RETURN at the end of the WRIT command before turning on the paper-tape punch in order to avoid punching an extra CARRIAGE RETURN, LINE FEED, RUB-OUT at the beginning of the tape.

CROMEMCO Z-1 MONITOR SOURCE LISTING

```
E000                    0005 *
E000                    0006 PPAGE  EQU    0E1H   ;MUST BE THE HIGHER OF
E000                    0007 * A PAIR OF NON-RAM PAGES.
E000                    0010 PSW    EQU    6
E000                    0015 SP     EQU    6
E000                    0016 PF     EQU    80H    ;PRIME-ABLE REG FLAG
E000                    0017 R2F    EQU    40H    ;2-BYTE REG FLAG
E000                    0020 BELL   EQU    07
E000                    0025 ESC    EQU    1BH
E000                    0030 CR     EQU    0DH
E000                    0035 LF     EQU    0AH
E000                    0040 STAT   EQU    0
E000                    0045 DAV    EQU    40H
E000                    0050 TBE    EQU    80H
E000                    0055 DATA   EQU    1
E000                    0060 TEMPS  EQU    16H    ;ROOM FOR TEMP STORAGE
E000                    0065 RSTLC  EQU    30H    ;RST LOCATION
E000                    0070 CASE   EQU    20H    ;DIFF BETW LOWER & UPPER CA
E000                    0075 *
E000                    0080 * Z80 OP-CODES
E000                    0085 JR     EQU    18H
E000                    0090 JRC    EQU    38H
E000                    0095 JRNC   EQU    30H
E000                    0100 JRZ    EQU    28H
E000                    0105 JRNZ   EQU    20H
E000                    0110 DJNZ   EQU    10H
E000                    0115 EXAF   EQU    08      ;EX AF,AF'
E000                    0120 EXX    EQU    0D9H
E000                    0125 RLD    EQU    0EDH
E000                    0130 RLDI   EQU    6FH
E000                    0135 CPI    EQU    0EDH
E000                    0140 CPII   EQU    0A1H
E000                    0145 CPIR   EQU    0EDH
E000                    0150 CPIRI  EQU    0B1H
E000                    0155 LDI    EQU    0EDH
E000                    0160 LDII   EQU    0A0H
E000                    0165 LDIR   EQU    0EDH
E000                    0170 LDIRI  EQU    0B0H
E000                    0175 LDD    EQU    0EDH
E000                    0180 LDDI   EQU    0A8H
E000                    0185 LDDR   EQU    0EDH
E000                    0190 LDDRI  EQU    0B8H
E000                    0195 SET5A  EQU    0CBH
E000                    0200 ST5AI  EQU    0EFH
E000                    0205 *
E000                    0210 IX     EQU    0DDH
E000                    0215 IY     EQU    0FDH
E000                    0220 *
```

```
E000                    0225 * DISPLACEMENTS FROM IX OF HI BYTE OF REG PAIRS
E000                    0230 DUPC    EQU     0
E000                    0235 DUSP    EQU     -2
E000                    0240 DUAF    EQU     -4
E000                    0245 DUBC    EQU     -6
E000                    0250 DUDE    EQU     -8
E000                    0255 DUHL    EQU     -10
E000                    0260 DUIT    EQU     -12     ;USER I & INTERRUPT ENABLE
E000                    0265 DUIX    EQU     -14
E000                    0270 DUIY    EQU     -16
E000                    0275 DUAF2   EQU     -18
E000                    0280 DUBC2   EQU     -20
E000                    0285 DUDE2   EQU     -22
E000                    0290 DUHL2   EQU     -24
E000                    0295 *
E000                    0300 *.,START
E000                    0305 * ENTER MONITOR FROM RESET
E000                    0310 *
E000                    0315 *
E000                    0320 * ORG 0E000H
E000                    0325 *
E000 3E 01              0330         MVI     A,1
E002 D3 40              0335         OUT     40H     ;SELECT BANK 0
E004                    0340 *
E004                    0345 * PLACE SYS STACK AT HIGHEST PAGE OF
E004                    0350 * AVAILABLE RAM.
E004                    0355 * ALLOW ROOM FOR TEMP STORAGE.
E004                    0360 *
E004 21 EB 00           0365         LXI     H,00FFH-TEMPS+2
E007 25                 0370 INIT    DCR     H
E008 7E                 0385         MOV     A,M
E009 34                 0390         INR     M
E00A BE                 0395         CMP     M       ;DID IT CHANGE?
E00B 28                 0400         DB      JRZ
E00C FA                 0405         DB      INIT-$-1
E00D 35                 0410         DCR     M       ;YES.  RESTORE IT.
E00E                    0415 *
E00E                    0440 * HL NOW POINTS TO BP STACK END
E00E                    0445 *
E00E 36 00              0455         MVI     M,0     ;BP STACK END MARK
E010 7D                 0465         MOV     A,L     ;SAVE
E011 2B                 0470         DCX     H       ;STORAGE FOR BPSP,LO
E012 77                 0475         MOV     M,A     ;STORE BPSP,LO
E013 11 E6 FF           0480         LXI     D,DUHL2-2
E016 19                 0481         DAD     D       ;TO END OF REG STORAGE
E017 F9                 0482         SPHL            ;SYS SP
E018                    0483 *
E018 ED                 0484         DB      0EDH    ;SBC HL,DE; BACK TO UPC,HI
E019 52                 0485         DB      52H     ;(CY WAS SET BY 'DAD D')
E01A E5                 0486         PUSH    H
E01B DD                 0487         DB      IX
E01C E1                 0488         POP     H       ;POP IX;STORAGE PNTR
E01D                    0489 *
E01D 16 E1              0490         MVI     D,PPAGE  ;FORCE USER SP TO
E01F 2B                 0491         DCX     H
E020 2B                 0492         DCX     H
E021 72                 0493         MOV     M,D     ;POINT TO PROM
E022                    0495 *
```

```
E022                    0500 * SET BAUD RATE
E022                    0505 *
E022 3E D8              0510 INIT1  MVI    A,0D8H  ;300 BAUD
E024 CD 36 E0           0515        CALL   BAUD
E027 3E F4              0520        MVI    A,0F4H  ;110 BAUD
E029 C4 36 E0           0525        CNZ    BAUD
E02C 20                 0537        DB     JRNZ
E02D F4                 0538        DB     INIT1-$-1
E02E                    0540 *
E02E 21 94 E3           0545        LXI    H,HEAD  ;HEADING
E031 CD 0C E2           0550        CALL   PMSG
E034                    0555 *
E034 18                 0560        DB     JR
E035 66                 0565        DB     CMND-$-1
E036                    0566 *
E036                    0567 *
E036 D3 00              0568 BAUD   OUT    STAT    ;SET BAUD RATE
E038 CD 41 E1           0569        CALL   GBYTE
E03B CD 41 E1           0570        CALL   GBYTE   ;CAN WE
E03E E6 7F              0571        ANI    7FH     ;READ
E040 FE 0D              0572        CPI    CR      ;A CR?
E042 C9                 0573        RET
E043                    0575 *
E043                    0580 * ENTER MONITOR FROM BRKPT
E043                    0585 *
E043                    0590 * SAVE MACHINE STATE.  SAVES ALL REGS INCLUDING
E043                    0595 * SP, FINDS THE TOP OF RAM INSTALLED IN MACHINE
E043                    0600 * & SWITCHES THE STACK THERE.
E043                    0605 *
E043 E3                 0610 SVMS   XTHL      ; ADJUST BRKPT RET ADDR
E044 2B                 0615        DCX    H
E045 E3                 0620        XTHL
E046                    0625 *
E046 E5                 0630        PUSH   H         ;SAVE
E047 21 04 00           0635        LXI    H,4
E04A 39                 0640        DAD    SP        ;USP (USER-SP)
E04B E3                 0645        XTHL          ;TO STACK
E04C                    0650 *
E04C F5                 0655        PUSH   PSW       ;UAF
E04D C5                 0660        PUSH   B         ;UBC
E04E D5                 0665        PUSH   D         ;UDE
E04F E5                 0670        PUSH   H         ;UHL
E050                    0675 *
E050                    0680 * FIND SYS STACK AGAIN
E050                    0685 *
E050 21 E9 00           0690        LXI    H,00FFH-TEMPS
E053 25                 0695 SVMS1  DCR    H        ;DECRM MEM PAGE
E054 7E                 0710        MOV    A,M
E055 34                 0715        INR    M
E056 BE                 0720        CMP    M       ;DID IT CHANGE?
E057 28                 0725        DB     JRZ
E058 FA                 0730        DB     SVMS1-$-1
```

```
E059 35              0735          DCR     M       ;YES.  RESTORE IT.
E05A                 0740 *
E05A EB              0745          XCHG
E05B 21 0B 00        0750          LXI     H,11
E05E 39              0755          DAD     SP      ;PNTS TO BPRA, HI BYTE
E05F 01 0C 00        0760          LXI     B,12
E062 ED              0765          DB      LDDR    ;TRANSFER TO SYS STACK
E063 B8              0770          DB      LDDR1
E064 13              0775          INX     D       ;DE HAS CURRENT VALUE OF
E065                 0780 * SYS SP AND POINTS TO UR
E065 23              0785          INX     H       ;HL HAS CURRENT VALUE OF
E066                 0790 * USER SP AND ALSO POINTS TO UR
E066 EB              0795          XCHG
E067 F9              0800          SPHL          ;SYS SP
E068                 0805 *
E068 ED              0810          DB      0EDH    ;LD A,I
E069 57              0815          DB      57H
E06A 0E 00           0820          MVI     C,0
E06C E2 70 E0        0825          JPO     SVMS3   ;IFF?
E06F 0C              0830          INR     C
E070                 0835 * C NOW HOLDS USER-IFF
E070 47              0840 SVMS3    MOV     B,A
E071 C5              0845          PUSH    B       ;UIF (USER-I & USER-IFF)
E072                 0850 *
E072 DD              0855          DB      IX
E073 E5              0860          PUSH    H       ;PUSH IX; UIX
E074 FD              0865          DB      IY
E075 E5              0870          PUSH    H       ;PUSH IY; UIY
E076 01 0B 00        0875          LXI     B,DUPC-DUHL+1
E079 09              0880          DAD     B       ;PNTS TO UPC, HI BYTE
E07A E5              0885          PUSH    H
E07B DD              0890          DB      IX
E07C E1              0895          POP     H       ;TO IX (POINTS TO UPC)
E07D                 0900 *
E07D 08              0905          DB      EXAF
E07E F5              0910          PUSH    PSW
E07F D9              0915          DB      EXX
E080 C5              0920          PUSH    B       ;UBC2
E081 D5              0925          PUSH    D       ;UDE2
E082 E5              0930          PUSH    H       ;UHL2
E083                 0935 *
E083                 0940 *
E083 DD              0945          DB      IX
E084 E5              0950          PUSH    H       ;PUSH IX
E085 E1              0955          POP     H
E086 23              0960          INX     H       ;POINTS TO BPSP,LO
E087 6E              0965          MOV     L,M     ;BPSP NOW IN HL
E088                 0970 *
E088                 0975 * CLEAR ALL BRKPTS
E088                 0980 *
```

```
E088 7E          0985 CLBP1  MOV    A,M     ;BP STK EMPTY?
E089 B7          0990        ORA    A
E08A 28          0995        DB     JRZ
E08B 0A          1000        DB     CLBP2-$-1
E08C             1005 *
E08C 2B          1010        DCX    H
E08D 56          1015        MOV    D,M
E08E 2B          1020        DCX    H
E08F 5E          1025        MOV    E,M
E090 2B          1030        DCX    H
E091 7E          1035        MOV    A,M
E092 12          1040        STAX   D       ;RESTORE CONTENTS TO MEM
E093 2B          1045        DCX    H
E094 18          1050        DB     JR
E095 F2          1055        DB     CLBP1-$-1
E096             1060 *
E096 7D          1065 CLBP2  MOV    A,L
E097 2B          1070        DCX    H
E098 77          1075        MOV    M,A     ;ADJUST @PSP
E099             1080 *
E099 CD F5 E0    1120        CALL   DSPR      ;DISPLAY USER REGISTERS
E09C             1125 *
E09C             1130 * GET 1-BYTE COMMAND.
E09C             1135 * RETURNS VALUE IN HL & JUMPS TO THAT ADDR.
E09C             1140 *
E09C CD 0C E1    1145 CMND   CALL   CRLF
E09F 21 B3 E3    1147 CMND1  LXI    H,PRMPT
E0A2 CD 0C E2    1150        CALL   PMSG
E0A5             1155 * HL NOW PNTS TO TBL ADDR
E0A5 CD 1A E2    1160        CALL   GCMND   ;DE GETS LETTER - 'A'
E0A8 EB          1165        XCHG
E0A9 29          1170        DAD    H       ;TIMES 2
E0AA 19          1175        DAD    D       ; + TBL ADDR
E0AB 5E          1180        MOV    E,M
E0AC 23          1185        INX    H
E0AD 56          1190        MOV    D,M
E0AE EB          1195        XCHG
E0AF 11 9F E0    1200        LXI    D,CMND1  ;SET UP RETURN
E0B2 D5          1205        PUSH   D        ;TO CMND
E0B3 79          1207        MOV    A,C     ;A & C HAVE CMND DELIMITER
E0B4 E9          1210        PCHL
E0B5             1215 *
E0B5             1416 *
E0B5             1417 *
E0B5             1418 * REJECTS ALL BUT ALPHABETIC CHARACTERS.
E0B5             1419 * RETURNS THE CHAR LESS THE ASCII VALUE OF 'A'.
E0B5             1420 *
E0B5 D6 61       1421 ABCYZ  SUI    'A'+CASE ;'A' OR ABOVE?
E0B7 38          1422        DB     JRC
E0B8 03          1423        DB     ERROR-$-1
E0B9 FE 19       1424        CPI    25D     ;'Y' OR BELOW?
E0BB D8          1425        RC             ;IF NOT, CONTINUE BELOW
```

```
E0BC                    1426 *
E0BC                    1427 *
E0BC                    1430 * ERROR & ESCAPE.  RETURNS TO CMND WITH SP
E0BC                    1435 * POINTING TO SAVED-REG AREA (UHL2).
E0BC                    1440 *
E0BC CD 09 E2           1445 ERROR  CALL   PSQS      ;PRINT ' ? <BELL>'
E0BF DD                 1450 ESCPE  DB     IX
E0C0 E5                 1455        PUSH   H         ;PUSH IX
E0C1 E1                 1460        POP    H
E0C2 11 E7 FF           1465        LXI    D,DUHL2-1-DUPC
E0C5 19                 1470        DAD    D
E0C6 F9                 1475        SPHL
E0C7 18                 1480        DB     JR
E0C8 D3                 1485        DB     CMND-$-1 ;GET NEW CMND
E0C9                    1490 *
E0C9                    1495 *
E0C9                    1530 * PROGRAM PROMS.  ABORTS IF DESTINATION
E0C9                    1535 * IS NOT ON A 1K (400H) BOUNDARY, SWATH
E0C9                    1540 * WIDTH IS NOT A MULTIPLE OF 1K.
E0C9                    1550 *
E0C9                    1570 *
E0C9 06 B5              1575 PROG   MVI    B,181  ;360 ITERATIONS
E0CB C5                 1580 PROG1  PUSH   B      ;SAVE # OF ITERATIONS
E0CC CD 70 E1           1590        CALL   LD2N   ;SOURCE TO DE,INCRM TO BC,
E0CF F5                 1595        PUSH   PSW    ;SAVE LATEST DELIMITER
E0D0 78                 1600        MOV    A,B      ;IS INCRM A MULT OF 1024?
E0D1 E6 03              1605        ANI    3
E0D3 B1                 1610        ORA    C
E0D4 20                 1615        DB     JRNZ
E0D5 E6                 1620        DB     ERROR-$-1
E0D6 F1                 1623        POP    PSW    ;LAST DELIMITER
E0D7 CD 8E E1           1625        CALL   LINCR     ;SOURCE TO HL, DEST TO DE
E0DA 7A                 1635        MOV    A,D       ;IS DEST A MULT OF 1024?
E0DB E6 03              1640        ANI    3
E0DD B3                 1645        ORA    E
E0DE 20                 1650        DB     JRNZ
E0DF DC                 1655        DB     ERROR-$-1
E0E0                    1660 *
E0E0 F1                 1665 PROG3  POP    PSW    ;ITERATIONS
E0E1 F5                 1670        PUSH   PSW
E0E2 C5                 1675        PUSH   B         ;INCREMENT
E0E3 01 00 04           1685        LXI    B,1024
E0E6 C5                 1687        PUSH   B      ;SAVE
E0E7 CD 50 E2           1690        CALL   MVE      ;MOVE IT
E0EA C1                 1691        POP    B      ;RETRIEVE
E0EB E3                 1730        XTHL          ;INCRM TO HL
E0EC B7                 1737        ORA    A      ;RESET CY
E0ED ED                 1740        DB     0EDH     ;SBC HL,BC
E0EE 42                 1745        DB     42H
E0EF E3                 1750        XTHL          ;SOURCE BACK TO HL
E0F0 C1                 1755        POP    B        ;NEW INCRM
```

```
E0F1 20             1760          DB      JRNZ
E0F2 ED             1765          DB      PROG3-$-1 ;LOOP IF INCRM NOT 0
E0F3 F1             1767          POP     PSW     ;CLEAN UP
E0F4 C9             1775          RET             ;BACK TO CMND
E0F5                1780 *
E0F5                1782 * COMMAND
E0F5                1785 *
E0F5                1800 * DISPLAY THE USER REGISTERS.
E0F5                1805 *
E0F5 CD 0C E1       1810 DSPR     CALL    CRLF
E0F8 DD             1815          DB      IX
E0F9 E5             1820          PUSH    H       ;PUSH IX
E0FA E1             1825          POP     H       ;POINTS TO UPC
E0FB 06 02          1830          MVI     B,2     ;UPC & USP
E0FD CD 07 E1       1835          CALL    PREGS
E100 06 07          1840          MVI     B,7     ;UAF THRU UIY
E102 CD 07 E1       1845          CALL    PREGS
E105 06 04          1850          MVI     B,4     ;UAF2 THRU UHL2
E107 CD 4A E1       1855 PREGS    CALL    P2BMS   ;PRINT 2 BYTES PNTED TO B
E10A 10             1860          DB      DJNZ
E10B FB             1865          DB      PREGS-$-1
E10C                1870 * (CONTINUE BELOW)
E10C                1875 *
E10C                1880 *
E10C                1885 * PRINT CR & LF. PRESERVES ALL REGS BUT A.
E10C                1890 *
E10C 3E 0D          1895 CRLF     MVI     A,CR
E10E                1900 * (CONTINUE BELOW)
E10E                1905 *
E10E                1910 *
E10E                1915 * PRINT THE CHARACTER IN THE A-REGISTER. (CHECKS
E10E                1920 * INPUT FOR ESCAPE.) PRESERVES ALL REGS.
E10E                1925 *
E10E F5             1930 PCHR     PUSH    PSW     ;SAVE THE CHAR
E10F DB 00          1935          IN      STAT
E111 E6 40          1940          ANI     DAV
E113 28             1945          DB      JRZ
E114 08             1950          DB      PCHR2-$-1
E115 DB 01          1955          IN      DATA
E117 E6 7F          1960          ANI     7FH
E119                1965 *
E119 FE 1B          1970 PCHR1    CPI     ESC
E11B 28             1975          DB      JRZ
E11C A2             1980          DB      ESCPE-$-1
E11D                1985 *
E11D DB 00          1990 PCHR2    IN      STAT
E11F E6 80          1995          ANI     TBE
E121 28             2000          DB      JRZ
E122 FA             2005          DB      PCHR2-$-1
E123 F1             2010          POP     PSW
E124 D3 01          2015          OUT     DATA
```

```
E126 F5             2020          PUSH    PSW
E127 E5             2025          PUSH    H
E128 21 80 E3       2030          LXI     H,LFNN
E12B FE 0D          2035          CPI     CR
E12D CC 0C E2       2040          CZ      PMSG
E130 E1             2045          POP     H
E131 F1             2050          POP     PSW
E132 C9             2055          RET
E133                2060 *
E133                2065 *
E133                2070 * GET CHARACTER. RETURNS IT IN A. CONVERTS
E133                2075 * ALPHA CHARS TO LOWER-CASE. ALTERS F.
E133                2080 *
E133 CD 41 E1       2085 GCHR     CALL    GBYTE
E136 E6 7F          2087          ANI     7FH
E138 FE 41          2090          CPI     'A'
E13A 38             2095          DB      JRC
E13B 02             2100          DB      GCHR1-$-1
E13C F6 20          2105          ORI     20H       ;CONVERT TO LOWER-CASSE
E13E F5             2110 GCHR1    PUSH    PSW     ;SAVE THE CHAR
E13F 18             2112          DB      JR
E140 D8             2115          DB      PCHR1-$-1   ;PRINT IT
E141                2116 *
E141                2117 *
E141 DB 00          2118 GBYTE    IN      STAT
E143 E6 40          2119          ANI     DAV
E145 28             2120          DB      JRZ
E146 FA             2121          DB      GBYTE-$-1
E147 DB 01          2122          IN      DATA
E149 C9             2123          RET
E14A                2124 *
E14A                2125 * PRINT 2 BYTES IN (HL) & (HL - 1).
E14A                2130 * DECREMENTS HL BY 2. ALTERS A. PRESERVES OTHERS
E14A                2135 *
E14A CD E5 E1       2140 P2BMS    CALL    PNM
E14D 2B             2145          DCX     H
E14E CD E5 E1       2150          CALL    PNM
E151 2B             2155          DCX     H
E152                2160 *
E152                2165 *
E152                2170 * PRINTS SPACE. PRESERVES ALL REGS BUT A.
E152                2175 *
E152 3E 20          2180 SPACE    MVI     A,20H
E154 18             2185          DB      JR
E155 B8             2190          DB      PCHR-$-1
E156                2195 *
E156                2200 *
E156                2205 * IF HL IS A MULTIPLE OF 16, DO PADDR.
E156                2210 *
E156 3E 0F          2215 CK16B    MVI     A,15
E158                2220 *
E158                2225 *
```

```
E158                    2230 * ENTER WITH A CONTAINING N. IF HL IS A MUL-
E158                    2235 * TIPLE OF N+1, DO PADDR.
E158                    2240 *
E158 A5                 2245 CKBND  ANA     L
E159 C0                 2250        RNZ
E15A                    2255 *
E15A                    2260 *
E15A                    2265 * PRINT THE NUMBER IN HL, FOLLOWED BY A COLON.
E15A                    2270 * PRESERVES ALL REGS EXCEPT A.
E15A                    2275 *
E15A CD 0C E1           2280 PADDR  CALL    CRLF
E15D CD DA E1           2285 PADR1  CALL    PNHL
E160 3E 3A              2290        MVI     A,':'
E162 18                 2295        DB      JR
E163 AA                 2300        DB      PCHR-$-1
E164                    2305 *
E164                    2310 *
E164                    2325 * LOAD TWO NUMBERS. FOLLOW WITH A CRLF.
E164                    2330 *
E164 CD 70 E1           2335 L2NCR  CALL    LD2N
E167                    2340 *
E167                    2345 * SKIP INITIAL SPACES.
E167                    2350 * IF DELIMITER NOT A CR, ERROR
E167                    2355 *
E167 CD D1 E1           2360 SKSGC  CALL    SKSG    ;LOOK FOR A NON-SPACE
E16A FE 0D              2365        CPI     CR      ;CR?
E16C C2 BC E0           2370        JNZ     ERROR
E16F C9                 2375        RET
E170                    2380 *
E170                    2385 *
E170                    2390 * LOAD TWO NUMBERS. LOADS DE WITH THE BEGINNING
E170                    2395 * ADDR, N1. LOADS BC & HL WITH THE INCREMENT
E170                    2400 * N2-N1+1 (OR WITH N2 IF THE OPR IS 'S').
E170                    2405 * RETURNS WITH LAST DELIMITER IN A.
E170                    2420 *
E170                    2440 *
E170 CD 97 E1           2455 LD2N   CALL    GNHL      ;N1 TO HL, DELIMITER TO A
E173 EB                 2460        XCHG              ;SAVE N1 IN DE
E174 CD D1 E1           2465        CALL    SKSG    ;GET NEXT NON-SPACE CHAR
E177 FE 73              2475        CPI     'S'+CASE ;SWATH?
E179 20                 2476        DB      JRNZ
E17A 06                 2477        DB      LD2N1-$-1
E17B                    2478 *
E17B AF                 2480        XRA     A       ;YES
E17C CD 97 E1           2485        CALL    GNHL    ;INCREMENT TO HL
E17F 18                 2505        DB      JR
E180 07                 2510        DB      LD2N2-$-1
E181                    2511 *
E181 CD 97 E1           2512 LD2N1  CALL    GNHL    ;INCREMENT
E184 B7                 2513        ORA     A       ;CLEAR CY
E185 ED                 2515        DB      0EDH      ;SBC HL,DE
```

```
E186 52              2520          DB      52H       ;N2-N1
E187 23              2525          INX     H         ;INCLUDE END POINT
E188 44              2550 LD2N2    MOV     B,H
E189 4D              2555          MOV     C,L       ;BC GETS THE INCRM
E18A C9              2565          RET
E18B                 2570 *
E18B                 2575 *
E18B                 2580 * LOAD 3 OPERANDS. HL GETS SOURCE, DE THE
E18B                 2585 * 3RD OPERAND, BC THE INCREMENT & A THE
E18B                 2590 * LOW BYTE OF THE 3RD OPERAND.
E18B                 2595 *
E18B CD 70 E1        2600 LD3N     CALL    LD2N
E18E                 2605 * (CONTINUE BELOW)
E18E                 2610 *
E18E                 2615 *
E18E                 2620 * TRANSFER DE TO HL. ENTER WITH SPACE OR
E18E                 2625 * 1ST DIGIT OF NUMBER IN A. GET NUMBBR
E18E                 2630 * INTO DE WITH LOW BYTE ALSO TO A.
E18E                 2635 * FINISHES WITH A CRLF.
E18E                 2640 *
E18E CD 97 E1        2645 LINCR    CALL    GNHL      ;SKIP SPACES, LOAD HL
E191 CD 67 E1        2650          CALL    SKSGC     ;WAIT FOR A CR
E194 7D              2655          MOV     A,L
E195 EB              2660          XCHG
E196 C9              2665          RET
E197                 2670 *
E197                 2675 *
E197                 2680 * CLEARS HL. IF ENTERED WITH HEX CHAR IN A,
E197                 2685 * SHIFTS IT INTO HL. O/W, IGNORES LEADING
E197                 2690 * SPACES. FIRST CHAR MUST BE HEX. CONTINUES
E197                 2695 * SHIFT UNTIL A NON-HEX CHAR RECEIVED & THEN
E197                 2700 * RETURNS WITH THE LATTER IN A.
E197                 2715 * PRESERVES B,C,D,E.
E197                 2720 *
E197                 2730 *
E197 C5              2735 GNHL     PUSH    B         ;SAVE
E198 21 00 00        2740 GNHL1    LXI     H,0       ;CLEAR BUFFER
E19B                 2745 * STRIP LEADING SPACES & GET CHAR
E19B CD D1 E1        2750          CALL    SKSG
E19E                 2760 * FIRST CHAR MUST BE HEX
E19E CD AF E1        2765          CALL    HEXSH     ;IF HEX, SHIFT INTO HL
E1A1 DA BC E0        2790          JC      ERROR     ;O/W,RETRY
E1A4 CD 33 E1        2795 GNHL3    CALL    GCHR
E1A7 CD AF E1        2840 GNHL5    CALL    HEXSH     ;IF HEX SHIFT INTO HL
E1AA 78              2845          MOV     A,B       ;RESTORE CHAR
E1AB 30              2850          DB      JRNC
E1AC F7              2855          DB      GNHL3-$-1 ;IF HEX, CONTINUE
E1AD C1              2860          POP     B         ;IF NON-HEX, DONE
E1AE C9              2865          RET
E1AF                 2870 *
```

```
E1AF                    2875 *
E1AF                    2880 * IF A CONTAINS HEX CHAR, SHIFTS BINARY EQUIVALE
E1AF                    2885 * INTO HL. IF NOT HEX, RET WITH CY SET. SAVES
E1AF                    2890 * ORIGINAL CHAR IN B
E1AF                    2895 *
E1AF 47                 2900 HEXSH  MOV     B,A
E1B0 D6 30              2905        SUI     '0'         ; < '0'?
E1B2 D8                 2910        RC
E1B3 C6 C9              2915        ADI     '0'-'G'-CASE
E1B5 D8                 2920        RC
E1B6 D6 FA              2925        SUI     'A'-'G'
E1B8 30                 2930        DB      JRNC        ;OK IF >= 'A'
E1B9 03                 2935        DB      HXSH0-$-1
E1BA C6 27              2940        ADI     'A'-'9'-1+CASE
E1BC D8                 2945        RC
E1BD C6 0A              2950 HXSH0  DW      0AC6H   ;ADI '9'+1-'0'
E1BF                    2955 * THE A-REG NOW CONTAINS THE HEX DIGIT IN BINARY
E1BF                    2960 * (THE HIGH-ORDER NIBBLE OF A IS 0.)
E1BF CD C8 E1           2965 HXSH4  CALL    HXSH1       ;SHIFT 4 BITS INTO HL
E1C2 CD C8 E1           2970        CALL    HXSH1
E1C5 CD C8 E1           2975        CALL    HXSH1
E1C8                    2980 *
E1C8 07                 2985 HXSH1  RLC             ;SHIFT INTO BIT 4
E1C9 29                 2990        DAD     H       ;SHIFT LEFT
E1CA                    2995 * CLEAR CY IN CASE OF RET FROM HEXSH
E1CA B7                 3000        ORA     A
E1CB CB                 3005        DB      0CBH        ;BIT 4,A
E1CC 67                 3010        DB      67H         ;IS IT 0?
E1CD C8                 3015        RZ
E1CE 23                 3020        INX     H
E1CF C9                 3025        RET
E1D0                    3030 *
E1D0                    3035 *
E1D0                    3040 * RETURNS WITH A NON-SPACE IN THE A-REG.
E1D0                    3045 * IF ENTERED WITH A-REG CONTAINING A NULL
E1D0                    3050 * OR A SPACE, GETS NEW CHARS UNTIL FIRST
E1D0                    3055 * NON-SPACE OCCURS. ALTERS AF.
E1D0                    3060 *
E1D0 AF                 3065 SKSGO  XRA     A           ;START WITH A NULL
E1D1                    3070 *
E1D1 B7                 3075 SKSG   ORA     A           ;DOES A CONTAIN NULL?
E1D2 CC 33 E1           3080 SKSG1  CZ      GCHR
E1D5 FE 20              3085        CPI     20H         ;SPACE?
E1D7 28                 3090        DB      JRZ
E1D8 F9                 3095        DB      SKSG1-$-1
E1D9 C9                 3100        RET
E1DA                    3105 *
E1DA                    3110 *
E1DA                    3330 *
E1DA                    3335 *
```

```
E1DA                    3350 * PRINT THE NUMBER IN HL. PRESERVES ALL REGS.
E1DA                    3355 *
E1DA F5                 3360 PNHL   PUSH   PSW
E1DB E5                 3365        PUSH   H          ;TO STACK
E1DC CD EB E1           3370        CALL   P4HEX
E1DF E1                 3375        POP    H
E1E0 F1                 3380        POP    PSW
E1E1 C9                 3385        RET
E1E2                    3390 *
E1E2                    3395 *
E1E2                    3400 * PRINT SPACE FOLLOWED BY THE NUMBER POINTED
E1E2                    3405 * TO BY HL. ALTERS A ONLY.
E1E2                    3410 *
E1E2 CD 52 E1           3415 PSNM   CALL   SPACE
E1E5                    3420 * (CONTINUE BELOW)
E1E5                    3425 *
E1E5                    3430 * PRINTS THE NUMBER POINTED TO BY HL.
E1E5                    3435 * PRESERVES  ALL RDGISTDRS.
E1E5                    3440 *
E1E5 F5                 3445 PNM    PUSH   PSW
E1E6 CD F3 E1           3450        CALL   P2HEX
E1E9 F1                 3455        POP    PSW
E1EA C9                 3460        RET
E1EB                    3465 *
E1EB                    3470 *
E1EB                    3475 * PRINTS 4 HEX CHARS FROM TOP OF STACK.
E1EB                    3480 * ALTERS F,H,L.
E1EB                    3485 *
E1EB 21 03 00           3490 P4HEX  LXI    H,3
E1EE 39                 3495        DAD    SP         ;HL = SP
E1EF CD F3 E1           3505        CALL   P2HEX      ;HIGH BYTE
E1F2 2B                 3510        DCX    H          ;LOW BYTE
E1F3                    3515 *
E1F3                    3520 *
E1F3                    3525 * PRINT THE NUMBER POINTED TO BY HL.
E1F3                    3530 * PRESERVES ALL REGS EXCEPT AF.
E1F3                    3535 *
E1F3 7E                 3540 P2HEX  MOV    A,M        ;GET THE NUMBER
E1F4 0F                 3545        RRC
E1F5 0F                 3550        RRC
E1F6 0F                 3555        RRC
E1F7 0F                 3560        RRC
E1F8 CD FC E1           3565        CALL   P1HEX      ;LEFT NIBBLE
E1FB 7E                 3570        MOV    A,M        ;NOW DO THE RIGHT NIBBLE
E1FC E6 0F              3575 P1HEX  ANI    0FH        ;MASK
E1FE FE 0A              3580        CPI    10         ; <= 9?
E200 38                 3585        DB     JRC
E201 02                 3590        DB     P1HX1-$-1
E202 C6 07              3595        ADI    7          ;A THRU F
E204 C6 30              3600 P1HX1  ADI    30H        ;ASCII BIAS
```

```
E206 C3 0E E1          3605          JMP     PCHR    ;PRINT IT
E209                   3615 *
E209                   3620 *
E209                   3625 * PRINT MESSAGE. ENTER WITH ADDR OF MSG
E209                   3630 * IN HL. MSG IS TERMINATED BY 00 THRU 07.
E209                   3635 * PRESERVES FLAGS, CLEARS A, INCRM HL.
E209                   3640 *
E209                   3645 *
E209                   3650 * PRINT ' ? <BELL>'
E209                   3655 *
E209 21 AD E3          3660 PSQS     LXI     H,SQS
E20C                   3665 *
E20C 3E 00             3670 PMSG     MVI     A,0       ;CLEAR A (FOR GNHL)
E20E F5                3675          PUSH    PSW       ;SAVE FLAGS
E20F 7E                3680 PMSG1    MOV     A,M
E210 23                3685          INX     H
E211 CD 0E E1          3690          CALL    PCHR
E214 E6 F8             3695          ANI     0F8H      ;<NULL> THRU <BELL>?
E216 20                3700          DB      JRNZ
E217 F7                3705          DB      PMSG1-$-1
E218 F1                3710          POP     PSW
E219 C9                3715          RET
E21A                   3720 *
E21A                   3725 *
E21A                   3980 *
E21A                   3985 *
E21A                   3990 * DE GETS THE FIRST ALPHA CHAR - 'A'.
E21A                   3991 * C GETS THE FIRST DELIMITER.
E21A                   3992 * B IS INITIALIZED TO '0' & RETURNS
E21A                   3993 * THE LAST CMND CHARACTER.
E21A                   3994 *
E21A CD D0 E1          3995 GCMND    CALL    SKSGO   ;GET NON-SPACE
E21D CD B5 E0          3996          CALL    ABCYZ   ;ALPHA CHECK
E220 5F                3997          MOV     E,A
E221 16 00             3998          MVI     D,0     ;DE HAS TBL DISPLACEMENT
E223 06 6F             3999          MVI     B,'0'+CASE ;INITIALIZE FOR GO CMND
E225 CD 33 E1          4000 GCMN1    CALL    GCHR    ;GET CHAR
E228 FE 30             4002          CPI     30H     ;DELIMITER ?
E22A 4F                4004          MOV     C,A     ;DELIM STORE
E22B D8                4006          RC              ;IF SO, DONE
E22C 47                4008          MOV     B,A     ;LAST CHAR STORE
E22D 18                4010          DB      JR
E22E F6                4015          DB      GCMN1-$-1
E22F                   4045 *
E22F                   4050 *
E22F                   4065 * COMMAND
E22F                   4070 *
E22F CD 8B E1          4075 VERIF    CALL    LD3N    ;GET 3 OPERANDS
E232                   4080 *
E232                   4085 *
E232                   4090 * COMPARES TWO AREAS OF MEMORY. ENTER WITH
E232                   4095 * SOURCE IN HL. DESTINATION IN DE & COUNT
```

```
E232                    4100 * IN BC. ALTERS ALL REGISTERS.
E232 1A                 4105 VRFY   LDAX   D       ;DESTINATION
E233 ED                 4110        DB     CPI      ;COMPARE TO SOURCE
E234 A1                 4115        DB     CPI1
E235 C4 0C E1           4120        CNZ    CRLF     ;IF NOT SAME, CRLF
E238 2B                 4125        DCX    H        ;(CPI INCRMS HL)
E239 C4 DA E1           4130        CNZ    PNHL     ; & PRINT SOURCE ADDR
E23C C4 E2 E1           4135        CNZ    PSNM     ; & SOURCE CONTENTS
E23F EB                 4140        XCHG
E240 C4 E2 E1           4145        CNZ    PSNM     ; & DEST CONTENTS
E243 EB                 4150        XCHG
E244 23                 4155        INX    H        ;RESTORE HL FOR CPI
E245 13                 4160        INX    D        ;NEXT DEST
E246 E2 0C E1           4165        JPO    CRLF    ;IF BC = 0, DONE
E249 18                 4170        DB     JR
E24A E7                 4175        DB     VRFY-$-1
E24B                    4180 *
E24B                    4185 *
E24B                    4190 * COMMAND
E24B                    4195 *
E24B CD 8B E1           4200 MOVE   CALL   LD3N    ;OPERANDS
E24E 3E 01              4210        MVI    A,1      ;# OF ITERATIONS
E250                    4215 *
E250                    4220 *
E250                    4225 * MOVE FROM ONE LOCATION TO ANOTHER. ENTER
E250                    4230 * WITH SOURCE ADDR IN HL, DEST IN DE, BYTE
E250                    4235 * COUNT IN BC. THE MOVE IS ITERATED N TIMES,
E250                    4240 * WHERE N = TWICE THE CONTENTS OF A, LESS ONE.
E250                    4245 * INCREMENTS HL & DE BY BC. CHECKS RESULT
E250                    4250 * & PRINTS THE ERRORS FOUND.
E250                    4255 *
E250 37                 4260 MVE    STC            ;CY IS USED IN ITERATION COU
E251 E5                 4265 MVE1   PUSH   H        ;SOURCE
E252 D5                 4270        PUSH   D        ;DEST
E253 C5                 4275        PUSH   B        ;BYTE COUNT
E254 F3                 4280        DI              ;FOR PROM PROGRAMMING
E255 ED                 4285        DB     LDIR     ;ONE ITERATION
E256 B0                 4290        DB     LDIR1
E257 FB                 4295        EI
E258 C1                 4300        POP    B
E259 D1                 4305        POP    D
E25A E1                 4310        POP    H
E25B                    4315 * ITERATION CALCULATIONS
E25B 3F                 4320        CMC
E25C 38                 4325        DB     JRC
E25D F3                 4330        DB     MVE1-$-1
E25E 3D                 4335        DCR    A
E25F 20                 4340        DB     JRNZ
E260 F0                 4345        DB     MVE1-$-1
E261                    4350 * CHECK RESULT
```

```
E261 18                 4355            DB      JR
E262 CF                 4360            DB      VRFY-$-1
E263                    4364 *
E263                    4365 *
E263                    4367 * COMMAND
E263                    4368 *
E263                    4370 * G0 <CR>     EXECUTION BEGINS AT USER PC.
E263                    4375 *
E263                    4376 * COMMAND
E263                    4377 *
E263                    4380 * G0 <ADDR1>/<ADDR2> ... <ADDRN>
E263                    4385 * EXECUTION BEGINS AT ADDR1 WITH BREAKPOINTS SET
E263                    4390 * AT ADDR2,...,ADDRN.
E263                    4395 *
E263 78                 4400 G0         MOV     A,B     ;CHECK THAT THE LAST
E264 FE 6F              4401            CPI     '0'+CASE ;CMND CHAR IS '0'
E266 C2 BC E0           4402            JNZ     ERROR
E269 79                 4403            MOV     A,C     ;CMND DELIMITER
E26A 0E 00              4405            MVI     C,0     ;BP FLAG
E26C CD D1 E1           4410 G01        CALL    SKSG    ;WAIT FOR NON-SPACE
E26F FE 0D              4415            CPI     CR
E271 28                 4420            DB      JRZ
E272 3A                 4425            DB      RETN-$-1  ;RETN IF CR
E273 FE 2F              4430            CPI     '/'     ;BP?
E275 20                 4435            DB      JRNZ
E276 0E                 4440            DB      G03-$-1
E277 0E 01              4445            MVI     C,1     ;SET BRKPT FLAG
E279 21 30 00           4450            LXI     H,RSTLC  ;TRANSFER
E27C 36 C3              4455            MVI     M,0C3H  ;'JMP SVMS' TO
E27E 21 43 E0           4460            LXI     H,SVMS
E281 22 31 00           4465            SHLD    RSTLC+1 ;RST LOC
E284 AF                 4468            XRA     A
E285 CD 97 E1           4470 G03        CALL    GNHL    ;GET ADDR
E288 CB 41              4475            DW      41CBH   ;BIT 0,C; FLAG SET?
E28A EB                 4478            XCHG
E28B 28                 4480            DB      JRZ
E28C 18                 4485            DB      G05-$-1 ;JMP IF NO BP
E28D DD                 4495            DB      IX
E28E E5                 4500            PUSH    H       ;PUSH IX
E28F E1                 4505            POP     H
E290 23                 4510            INX     H
E291 6E                 4515            MOV     L,M     ;HL = BPSP
E292                    4520 *
E292 23                 4525            INX     H       ;BUMP BPSP
E293 EB                 4530            XCHG            ;DE=BPSP, HL= BP ADDR
E294 46                 4535            MOV     B,M     ;CONTENTS
E295 36 F7              4537            MVI     M,0C7H+RSTLC ;RST INSTRUCTION
E297 EB                 4540            XCHG            ;HL=BPSP
E298 70                 4545            MOV     M,B     ;TO BP STACK
E299 23                 4550            INX     H       ;BUMP BPSP
```

```
E29A 73              4555          MOV     M,E     ;BP ADDR TO STACK
E29B 23              4560          INX     H
E29C 72              4565          MOV     M,D
E29D 23              4570          INX     H
E29E 36 01           4575          MVI     M,01    ;PUNCTUATION (BP SET)
E2A0 DD              4580          DB      IX
E2A1 75              4585          MOV     M,L     ;LD (IX+1),L
E2A2 01              4590          DB      1
E2A3 18              4592          DB      JR
E2A4 C7              4593          DB      G01-$-1
E2A5                 4595 * CHANGE USER PC
E2A5 DD              4600 G05      DB      IX
E2A6 72              4605          MOV     M,D       ;LD (IX+DUPC),D
E2A7 00              4610          DB      DUPC
E2A8 DD              4615          DB      IX
E2A9 73              4620          MOV     M,E     ;LD (IX+DUPC-1),E
E2AA FF              4625          DB      DUPC-1
E2AB 18              4630          DB      JR
E2AC BF              4635          DB      G01-$-1   ;BACK FOR MORE
E2AD                 4640 *
E2AD E1              4645 RETN     POP     H       ;STRIP CMND ADDR FROM STK
E2AE E1              4650          POP     H         ;UHL2
E2AF D1              4655          POP     D         ;UDE2
E2B0 C1              4660          POP     B         ;UBC2
E2B1 F1              4665          POP     PSW       ;UAF2
E2B2 D9              4670          DB      EXX
E2B3 08              4675          DB      EXAF
E2B4 FD              4680          DB      IY
E2B5 E1              4685          POP     H         ;POP IY; UIY
E2B6 DD              4690          DB      IX
E2B7 E1              4695          POP     H         ;POP IX; UIX
E2B8                 4700 *
E2B8 F1              4705          POP     PSW       ;UIF
E2B9 ED              4710          DB      0EDH
E2BA 47              4715          DB      47H       ;LD I,A; UI
E2BB F3              4720          DI
E2BC 30              4725          DB      JRNC
E2BD 01              4730          DB      RETN1-$-1
E2BE FB              4735          EI
E2BF                 4740 *IFF NOW RESTORED
E2BF E1              4745 RETN1    POP     H         ;UHL
E2C0 D1              4750          POP     D         ;UDE
E2C1 C1              4755          POP     B         ;UBC
E2C2 F1              4760          POP     PSW       ;UAF
E2C3 E3              4765          XTHL            ;USP TO HL, UHL TO (SP)
E2C4 F5              4770          PUSH    PSW
E2C5 C5              4775          PUSH    B
E2C6 D5              4780          PUSH    D
E2C7 01 0A 00        4785          LXI     B,10
E2CA EB              4790          XCHG            ;USP TO DE
E2CB 1B              4795          DCX     D
```

```
E2CC 21 09 00          4800          LXI     H,9
E2CF 39                4805          DAD     SP
E2D0 ED                4810          DB      LDDR      ;TRANSFER UPC THRU UHL, L
E2D1 B8                4815          DB      LDDR1     ;TO USER STACK
E2D2 EB                4820          XCHG              ;IS (USER SP - 1) RAM?
E2D3 7E                4825          MOV     A,M
E2D4 34                4830          INR     M
E2D5 BE                4835          CMP     M         ;DID IT CHANGE?
E2D6 28                4840          DB      JRZ
E2D7 03                4845          DB      RETN2-$-1
E2D8                   4850 *
E2D8 35                4855          DCR     M         ;YES. RESTORE IT.
E2D9 F9                4860          SPHL              ;CHANGE TO USER STACK
E2DA 33                4865          INX     SP        ;CORRECT FOR LDDR EXTRA DCR
E2DB                   4866 *
E2DB D1                4870 RETN2    POP     D           ;OTHERWISE, CONTINUE SYS
E2DC C1                4875          POP     B
E2DD F1                4880          POP     PSW
E2DE E1                4885          POP     H
E2DF C9                4890          RET
E2E0                   4891 *
E2E0                   4900 * ENTER WITH HL POINTING TO MEMORY & B CONTAININ
E2E0                   4905 * THE 2-BYTE REG FLAG.
E2E0                   4910 * PRINTS SPACE, CONTENTS OF (HL) & ALSO (HL-1) F
E2E0                   4915 * 2-BYTE REGS, GETS SUBSTITUTION VALUE INTO DE,
E2E0                   4920 * WRITES E INTO (HL) OR (HL-1) FOR 2-BYTE REGS.
E2E0                   4925 * RETURNS WITH Z-FLAG RESET IFF A CHANGE IS INDI
E2E0                   4930 * CATED (BY A LACK OF '.') FOR A 2-BYTE REG.
E2E0                   4935 * PRESERVES BC,HL.
E2E0                   4940 *
E2E0 CD E2 E1          4945 GSUBV    CALL    PSNM      ;PRINT (HL)
E2E3 CB                4950          DB      0CBH      ;BIT 6,B
E2E4 70                4955          DB      70H       ;2-BYTE REG?
E2E5 28                4960          DB      JRZ
E2E6 04                4965          DB      GSBV1-$-1
E2E7 2B                4970          DCX     H         ;YES, PRINT
E2E8 CD E5 E1          4975          CALL    PNM       ;  LO BYTE
E2EB 3E 2E             4980 GSBV1    MVI     A,'.'
E2ED CD 0E E1          4985          CALL    PCHR
E2F0 CD 33 E1          4990          CALL    GCHR
E2F3 FE 2E             4995          CPI     '.'       ;SUBSTITUTION?
E2F5 CC 0E E1          5000          CZ      PCHR      ;IF NOT, PRINT ANOTHER
E2F8 28                5005          DB      JRZ
E2F9 08                5007          DB      GSBV2-$-1
E2FA EB                5010          XCHG
E2FB CD 97 E1          5015          CALL    GNHL      ;NEW VALUE
E2FE EB                5020          XCHG              ;TO DE
E2FF 73                5025          MOV     M,E       ;LOAD MEM
E300                   5035 * THE FOLLOWING TEST IS FOR SBSR
E300 CB                5040          DB      0CBH      ;BIT 6,B
```

```
E301 70                 5045            DB      70H     ;2-BYTE REG?
E302 23                 5050 GSBV2      INX     H
E303 C9                 5052            RET
E304                    5055 *
E304                    5060 *
E304                    5061 * COMMAND
E304                    5062 *
E304                    5065 * SM <ADDR>     SUBSTITUTE MEMORY LOCATION.
E304                    5070 *
E304                    5072 * COMMAND
E304                    5073 *
E304                    5075 * SR <REGISTER NAME>     SUBSTITUTE USER REGISTER
E304                    5080 *
E304                    5085 * REGISTER NAMES: P (PC), S (SP),
E304                    5090 *                  A, F, B, C, D, E, H, L,
E304                    5095 *                  I, T (IFF), X (IX), Y (IY),
E304                    5100 *                  A',F',B',C',D',E',H',L'.
E304                    5105 *
E304 78                 5110 SUBST      MOV     A,B     ;LAST CMND CHAR
E305 FE 72              5115            CPI     'R'+CASE  ;SR?
E307 79                 5117            MOV     A,C     ;DELIMITER
E308 28                 5120            DB      JRZ
E309 0F                 5125            DB      SBSR-$-1
E30A                    5130 *
E30A CD 97 E1           5135 SBSM       CALL    GNHL    ;HL GETS ADDR
E30D 06 00              5140 SBSM1      MVI     B,0     ;REG FLAGS
E30F                    5145 * PRINT CURRENT VALUE, REQUEST NEW VALUE &
E30F                    5150 * PRINT IT IF GIVEN
E30F CD E0 E2           5155            CALL    GSUBV
E312 3E 07              5160            MVI     A,7     ;8 ENTRIES PER LINE
E314 CD 58 E1           5165            CALL    CKBND
E317 18                 5170            DB      JR
E318 F4                 5175            DB      SBSM1-$-1
E319                    5180 *
E319 CD 1A E2           5185 SBSR       CALL    GCMND   ;DE GETS LETTER - 'A'
E31C 21 E7 E3           5190            LXI     H,RGTBL
E31F 19                 5195            DAD     D       ;PNTS TO REG DISPLACEMENT
E320 42                 5200            MOV     B,D     ;D = 0
E321 CB                 5205            DB      0CBH    ;BIT 7,(HL)
E322 7E                 5210            DB      7EH     ;A THRU L?
E323 28                 5215            DB      JRZ
E324 0C                 5220            DB      SBSR1-$-1
E325 79                 5225            MOV     A,C     ;LAST CMND DELIMITER
E326 FE 20              5230            CPI     20H     ;SPACE?
E328 28                 5235            DB      JRZ
E329 07                 5240            DB      SBSR1-$-1
E32A FE 27              5245            CPI     ''''    ;PRIMED?
E32C C2 BC E0           5250            JNZ     ERROR
E32F 06 0E              5265            MVI     B,DUAF-DUAF2 ;YES
E331                    5270 *
E331 7E                 5275 SBSR1      MOV     A,M     ;DISPLACEMENT & FLAGS
```

```
E332 B7             5277          ORA     A       ;IF 0, ILLEGAL CMND
E333 CA BC E0       5278          JZ      ERROR
E336 E6 1F          5280          ANI     1FH       ;STRIP FLAGS OFF
E338 80             5285          ADD     B       ;ADJUST FOR PRIMES
E339 5F             5290          MOV     E,A       ;DE GETS DISPL (D=0)
E33A 46             5295          MOV     B,M       ;SAVE ORIG ENTRY
E33B DD             5300          DB      IX
E33C E5             5305          PUSH    H         ;PUSH IX
E33D E1             5310          POP     H         ;STACK FRAME
E33E ED             5315          DB      0EDH    ;SBC HL,DE
E33F 52             5317          DB      52H     ;PNTS TO USER REG
E340                5320 * PRINT CURRENT VALUE, DE GETS SUBSTITUTION
E340                5325 * VALUE, IF ANY, & (HL) OR (HL-1) GETS E.
E340                5330 * Z-FLAG RESET IFF CHANGE FOR A 2-BYTE REG.)
E340 CD E0 E2       5335          CALL    GSUBV
E343 28             5340          DB      JRZ
E344 01             5345          DB      SBSR3-$-1
E345 72             5350          MOV     M,D       ;NO. HI BYTE
E346 CD 52 E1       5355 SBSR3    CALL    SPACE
E349 18             5360          DB      JR
E34A CE             5365          DB      SBSR-$-1
E34B                5370 *
E34B                5371 *
E34B 78             5372 DISPL    MOV     A,B     ;LAST CMND CHAR
E34C FE 72          5373          CPI     'R'+CASE  ;DR?
E34E 79             5374          MOV     A,C     ;CMND DELIMITER
E34F CA F5 E0       5375          JZ      DSPR
E352                5377 *
E352                5379 *
E352                5380 * COMMAND
E352                5381 *
E352                5390 * DISPLAY MEMORY.
E352                5395 *
E352 CD 64 E1       5400 DSPM     CALL    L2NCR   ;N1TO DE, INCRM TO BC,
E355                5405 *                        ;DELIMITER TO A
E355 EB             5410          XCHG              ;N1 TO HL
E356 CD 5D E1       5415 DSPM1    CALL    PADR1   ;PRINT ADDR, ':'
E359 CD E2 E1       5420 DSPM2    CALL    PSNM      ;PRINT CONTENTS OF MEM
E35C 23             5425          INX     H
E35D 0B             5430          DCX     B
E35E 78             5435          MOV     A,B
E35F B1             5440          ORA     C       ;DONE?
E360 CA 0C E1       5445          JZ      CRLF
E363 CD 56 E1       5450          CALL    CK16B     ;CHECK FOR 16 COUNT
E366 18             5455          DB      JR
E367 F1             5460          DB      DSPM2-$-1
E368                5465 *
E368                5470 *
E368                5475 *...SUBDM 00 7E  5 585 BY 5    100 DBE++
E368                5500 *
```

```
E368                    5505 *
E368                    5507 * COMMAND
E368                    5510 * READ BINARY INPUT FROM DATA PORT
E368                    5512 *
E368 CD 64 E1           5514 READB  CALL   L2NCR  ;GET MEM ADDRS
E36B CD 41 E1           5516 RDB1   CALL   GBYTE  ;GET INPUT
E36E 12                 5518        STAX   D      ;TO MEM
E36F 13                 5520        INX    D
E370 0B                 5522        DCX    B      ;COUNT
E371 78                 5524        MOV    A,B
E372 B1                 5526        ORA    C      ;BC = 0?
E373 20                 5528        DB     JRNZ
E374 F6                 5530        DB     RDB1-$-1
E375 C9                 5532        RET
E376                    5534 *
E376                    5536 *
E376                    5537 * COMMAND
E376                    5538 * WRITE BINARY OUTPUT TO DATA PORT
E376                    5540 *
E376 CD 64 E1           5542 WRITB  CALL   L2NCR  ;GET MEM ADDRS
E379 DB 00              5544 WRTB1  IN     STAT
E37B E6 80              5546        ANI    TBE
E37D 28                 5548        DB     JRZ
E37E FA                 5550        DB     WRTB1-$-1
E37F 1A                 5552        LDAX   D
E380 D3 01              5554        OUT    DATA
E382 13                 5556        INX    D
E383 0B                 5558        DCX    B
E384 78                 5560        MOV    A,B
E385 B1                 5562        ORA    C
E386 20                 5564        DB     JRNZ
E387 F1                 5566        DB     WRTB1-$-1
E388 C9                 5568        RET
E389                    5570 *
E389                    5572 *
E389                    5574 * COMMAND
E389                    5576 * OUT <DATA-BYTE> <PORT NNUMBER>
E389                    5578 *
E389 CD 97 E1           5580 OUTP   CALL   GNHL
E38C EB                 5582        XCHG           ;E GETS DATA
E38D CD 97 E1           5584        CALL   GNHL    ;GET PORT NUMBER
E390                    5585 *
E390 4D                 5586        MOV    C,L     ;  TO C
E391 ED 59              5588        DW     59EDH   ;OUT (C),E
E393 C9                 5590        RET
E394                    5591 *
E394                    5592 *
E394 0D                 5593 HEAD   DB     CR
E395 0D                 5595        DB     CR
```

```
E396 43 52 4F 4D      5600          ASC     "CROMEMCO MON1.0 C.1976"
     45 4D 43 4F
     20 4D 4F 4E
     31 2E 30 20
     43 2E 31 39
     37 36
E3AC 00               5605          DB      0
E3AD                  5610 *
E3AD                  5630 *
E3AD 20 3F            5635 SQS      ASC     " ?"
E3AF 07               5640          DB      BELL
E3B0                  5645 *
E3B0 0A               5650 LFNN     DB      LF
E3B1 7F               5655          DB      7FH       ;NULL
E3B2 00               5660          DB      0
E3B3                  5665 *
E3B3                  5690 *
E3B3 3A               5695 PRMPT    DB      ':'
E3B4 00               5700          DB      0
E3B5                  5705 * THE COMMAND TBL MUST IMMEDIATELY FOLLOW
E3B5                  5706 * THE PROMPT MESSAGE
E3B5 BC E0            5710          DW      ERROR   ;A
E3B7 BC E0            5715          DW      ERROR   ;BANK
E3B9 BC E0            5720          DW      ERROR   ;C
E3BB 4B E3            5725          DW      DISPL   ;DISPLAY
E3BD BC E0            5730          DW      ERROR   ;ENTER
E3BF BC E0            5735          DW      ERROR   ;FILE
E3C1 63 E2            5740          DW      GO
E3C3 BC E0            5745          DW      ERROR   ;H
E3C5 BC E0            5750          DW      ERROR   ;INPUT
E3C7 BC E0            5755          DW      ERROR   ;J
E3C9 BC E0            5760          DW      ERROR   ;K
E3CB BC E0            5765          DW      ERROR   ;LIST
E3CD 4B E2            5770          DW      MOVE
E3CF BC E0            5775          DW      ERROR   ;NUMBER
E3D1 89 E3            5780          DW      OUTP    ;OUTPUT
E3D3 C9 E0            5785          DW      PROG    ;PROGRAM
E3D5 BC E0            5790          DW      ERROR   ;Q
E3D7 68 E3            5795          DW      READB   ;READ BINARY OR ASCII
E3D9 04 E3            5800          DW      SUBST   ;SUBSTITUTE
E3DB BC E0            5805          DW      ERROR   ;TRAP
E3DD BC E0            5810          DW      ERROR   ;UNEQUAL
E3DF 2F E2            5815          DW      VERIF   ;VERIFY
E3E1 76 E3            5820          DW      WRITB   ;WRITE BINARY OR ASCII
E3E3 BC E0            5825          DW      ERROR   ;X
E3E5 BC E0            5830          DW      ERROR   ;Y
E3E7                  5840 *
E3E7                  5850 *
E3E7                  5851 *
E3E7 84               5852 RGTBL    DB      -DUAF+PF ;A
```

```
E3E8 86          5853        DB      -DUBC+PF ;B
E3E9 87          5854        DB      -DUBC+1+PF ;C
E3EA 88          5855        DB      -DUDE+PF ;D
E3EB 89          5856        DB      -DUDE+1+PF ;E
E3EC 85          5857        DB      -DUAF+1+PF ;F
E3ED 00          5858        DB      0
E3EE 8A          5859        DB      -DUHL+PF ;H
E3EF 0C          5860        DB      -DUIT  ;I
E3F0 00          5861        DB      0
E3F1 00          5862        DB      0
E3F2 8B          5863        DB      -DUHL+1+PF ;L
E3F3 00          5864        DB      0
E3F4 00          5865        DB      0
E3F5 00          5866        DB      0
E3F6 40          5867        DB      -DUPC+R2F ;PC
E3F7 00          5868        DB      0
E3F8 00          5869        DB      0
E3F9 42          5870        DB      -DUSP+R2F ;SP
E3FA 0D          5871        DB      -DUIT+1 ;T (INTERRUPT ENABLE)
E3FB 00          5872        DB      0
E3FC 00          5873        DB      0
E3FD 00          5874        DB      0
E3FE 4E          5875        DB      -DUIX+R2F ;X (IX)
E3FF 50          5876        DB      -DUIY+R2F ;Y (IY)
```