

```

;
; INTELLEC8/MOD 80 MONITOR
;
; COPYRIGHT (C) 1973, 1974, 1975
; INTEL CORPORATION
; 3065 BOWERS AVENUE
; SANTA CLARA, CALIFORNIA 95051
;
; <LEGAL COMMAND> ::= <ASSIGN I/O COMMAND>
; <BNPF PUNCH COMMAND>
; <COMPARE COMMAND>
; <DISPLAY MEMORY COMMAND>
; <ENDFILE COMMAND>
; <FILL MEMORY COMMAND>
; <PROGRAM EXECUTE COMMAND>
; <HEXADECIMAL ARITHMETIC COMMAND>
; <LOAD BNPF COMMAND>
; <MOVE MEMORY COMMAND>
; <LEADER COMMAND>
; <PROGRAM COMMAND>
; <READ HEXADECIMAL FILE COMMAND>
; <SUBSTITUTE MEMORY COMMAND>
; <TRANSFER COMMAND>
; <WRITE HEXADECIMAL RECORD COMMAND>
; <REGISTER MODIFY COMMAND>
;
; <ASSIGN I/O COMMAND> ::= A<LOGICAL DEVICE>=<PHYSICAL DEVICE>
;
; <BNPF PUNCH COMMAND> ::= B<NUMBER>,<NUMBER>
;
; <COMPARE COMMAND> ::= C<NUMBER>
;
; <DISPLAY MEMORY COMMAND> ::= D<NUMBER>,<NUMBER>
;
; <ENDFILE COMMAND> ::= E<NUMBER>
;
; <FILL MEMORY COMMAND> ::= F<NUMBER>,<NUMBER>,<NUMBER>
;
; <PROGRAM EXECUTE COMMAND> ::= G<NUMBER>,<NUMBER>,<NUMBER>
;
; <HEXADECIMAL ARITHMETIC COMMAND> ::= H<NUMBER>,<NUMBER>
;
; <LOAD BNPF COMMAND> ::= L<NUMBER>,<NUMBER>
;
; <MOVE MEMORY COMMAND> ::= M<NUMBER>,<NUMBER>,<NUMBER>
;
; <LEADER COMMAND> ::= N
;
; <PROGRAM COMMAND> ::= P<NUMBER>,<NUMBER>,<NUMBER>
;
; <READ HEXADECIMAL FILE COMMAND> ::= R<NUMBER>
;

```

```

; <SUBSTITUTE MEMORY COMMAND> ::= S<NUMBER>
;
; <TRANSFER COMMAND> ::= T<NUMBER>
;
; <WRITE HEXADECIMAL RECORD COMMAND> ::= W<NUMBER>,<NUMBER>
;
; <REGISTER MODIFY COMMAND> ::= X<REGISTER IDENTIFIER>
;
; <LOGICAL DEVICE> ::= CONSOLE!READER!LIST!PUNCH
;
; <PHYSICAL DEVICE> ::= CRT!TTY!PTR!PTP!BATCH!1!2
;
; <REGISTER IDENTIFIER> ::= A!B!C!D!E!F!H!L!M!P!S
;
; <NUMBER> ::= <HEX DIGIT>
; <NUMBER><HEX DIGIT>
;
; <HEX DIGIT> ::= 0!1!2!3!4!5!6!7!8!9!A!B!C!D!E!F
;
; SYSTEM SIGNS ON WITH <CR><LF><.>
;
001E VER EQU 30 ; VERSION 3.0
; TITLE ' INTELLEC8/MOD 80 MONITOR, VERSION 3.0, 14 APRIL 1975 '
;
; I/O DEVICE OUTPUT COMMAND PORT 1 (TTC) BIT VALUES
;
; BIT REST MNEMONIC DESCRIPTION
;
; 0 0 RBIT TTY READER GO/NO GO
; 1 0 PCMD PTP GO/NO GO
; 2 0 RCMD PTR GO/NO GO
; 3 1 DSB PROM ENABLE/DISABLE, DSB=1
; 4 0 DATA IN T/C
; 5 0 DATA OUT T/C
; 6 0 PBIT 1702 PROM PROG. GO/NO GO
; 7 0 PBITA 1702A PROM PROG. GO/NO GO
;
; I/O DEVICE INPUT STATUS PORT 1 (TTS) BIT VALUES
;
; BIT REST MNEMONIC DESCRIPTION
;
; 0 1 TTYDA IF TTYDA = 0, INPUT IS READY
; 1 1 OVERRUN ERROR
; 2 0 TTYBE IF TTYBE = 0, OUTPUT IS READY
; 3 1 FRAMING ERROR
; 4 1 PARITY ERROR
; 5 0 PIRDA IF PTRDA = 1, PTR HAS CHAR
; 6 1 PRDY IF PRDY = 1, PTP IS READY
; 7 UNASSIGNED
;
; I/O DEVICE INPUT STATUS PORT 5 (CRTS) BIT VALUES

```

```

;
; BIT      REST  MNEMONIC      DESCRIPTION
;
; 0        1      CRIDA      IF CRIDA = 0, INPUT IS READY
; 1        1      OVERRUN    OVERRUN ERROR
; 2        0      CRIBE      IF CRIBE = 0, OUTPUT IS READY
; 3        1      FRAMING    FRAMING ERROR
; 4        1      PARITY     PARITY ERROR
; 5        1      UNASSIGNED UNASSIGNED
; 6        1      UNASSIGNED UNASSIGNED
; 7        1      UNASSIGNED UNASSIGNED
;
; I/O COMMAND CONSTANTS
;
0001      RB11 EQU      1
0002      PCMD EQU      2
0004      RCMD EQU      4
0008      DSB  EQU      8
0080      PB11A EQU     80H
;
; TTY I/O CONSTANTS
;
0000      TTI  EQU      0      ; TTY INPUT DATA PORT
0000      TTO  EQU      0      ; TTY OUTPUT DATA PORT
0001      TTS  EQU      1      ; TTY INPUT STATUS PORT
0001      TTC  EQU      1      ; TTY OUTPUT COMMAND PORT
0009      TTYG EQU      8BIT OR DSB ; START TTY READER
0008      TTYN EQU      DSB     ; STOP TTY READER
0001      TTYD EQU      1      ; DATA AVAILABLE
0004      TTYE EQU      4      ; TRANSMIT BUFFER EMPTY
;
; CRT I/O CONSTANTS
;
0004      CRTI EQU      4      ; CRT INPUT DATA PORT
0005      CRTS EQU      5      ; CRT INPUT STATUS PORT
0004      CRTO EQU      4      ; CRT OUTPUT DATA PORT
0001      CRIDA EQU      1      ; DATA AVAILABLE
0004      CRIBE EQU      4      ; TRANSMIT BUFFER EMPTY
;
; PIR I/O CONSTANTS
;
0003      PIRI EQU      3      ; PIR INPUT DATA PORT (NOT INVERTED)
0001      PIRS EQU      TTS     ; PIR INPUT STATUS PORT
0001      PTRC EQU      TTC     ; PTR OUTPUT COMMAND PORT
000C      PTRG EQU      RCMD OR DSB ; START PTR
0008      PTRN EQU      TTYN    ; STOP PTR
0020      PTRD EQU      20H     ; PTR DATA AVAILABLE
;
; PIP I/O CONSTANTS
;
0003      PIPU EQU      3      ; PIP OUTPUT DATA PORT

```

```

0001      PTPS EQU      ITS          ; PTP INPUT STATUS PORT
0001      PTPC EQU      TTC          ; PTP OUTPUT COMMAND PORT
0040      PRDY EQU      40H         ; PUNCH READY STATUS
000A      PTPGO EQU     PCMD OR DSB  ; START PUNCH
0008      PTPNO EQU     TTYNO       ; STOP PUNCH
;
; PROM PROGRAMMER I/O CONSTANTS
;
0002      PAD EQU      2            ; PROM ADDRESS OUTPUT PORT
0003      PDD EQU      PTPU        ; PROM DATA OUTPUT PORT
0002      PDI EQU      2            ; PROM DATA INPUT PORT
0001      PROMC EQU     TTC          ; PROGRAMMING PULSE OUTPUT PORT
0080      PRUGO EQU     PBITA       ; START PROGRAMMING
0000      PRONG EQU     0           ; STOP PROGRAMMING
0000      ENB EQU      0            ; ENABLE PROGRAMMER
;
; CONDITIONAL ASSEMBLY SWITCHES
;
0000      FALSE EQU    0            ;
FFFF      TRUE EQU     NOT FALSE    ;
0000      DEBUG EQU    FALSE        ; DEBUG MODE -
;                                     ; MODIFY CERTAIN CODE SECTIONS
;                                     ; SO THAT VERSION 2.0 MAY BE DEBUGGED
;                                     ; BY VERSION 1.0
;
; GLOBAL CONSTANTS
;
00FA      TOUT EQU     250          ; 250 MS. COUNTER FOR READER TIMEOUT
0014      LDLY EQU     20           ; COUNTER FOR 20 MS DELAY
;
;
1         IF          DEBUG
1         DLY EQU     111          ; COUNTER FOR 1.0 MS DELAY IN RAM
;
1         ENDIF
1         IF          NOT DEBUG
0054 1     DLY EQU     84           ; COUNTER FOR 1.0 MS DELAY IN PROM
;
1         ENDIF
000D      CR EQU      0DH          ; ASCII VALUE OF CARRIAGE RETURN
000A      LF EQU      0AH          ; ASCII VALUE OF LINE FEED
;
; MACRO DEFINITIONS
;
1         FETCH MACRO VALUE        ; FETCH THE ADDRESS OF A VALUE
1         LXI H,VALUE              ; IN THE STACK
1         DAD SP
;
;
;
; I/O STATUS BYTE MASKS AND VALUES
;
00FC      CMSK EQU     11111100B   ; MASK FOR CONSOLE I/O
00F3      RMSK EQU     11110011B   ; MASK FOR READER INPUT
00CF      PMSK EQU     11001111B   ; MASK FOR PUNCH OUTPUT
003F      LMSK EQU     00111111B   ; MASK FOR LIST OUTPUT

```

```

;
0000 CITY EQU 0000000B ; CONSOLE I/O = ITY
0001 CCRT EQU 00000001B ; CONSOLE I/O = CRT
0002 BATCH EQU 00000010B ; BATCH MODE,
; INPUT = READER, OUTPUT = LIST
0003 CUSE EQU 00000011B ; USER DEFINED CONSOLE I/O
0004 RTTY EQU 00000000B ; READER = TTY
0004 RPTR EQU 00000100B ; READER = PIR
0008 RUSE1 EQU 00001000B ; USER DEFINED READER (1)
000C RUSE2 EQU 00001100B ; USER DEFINED READER (2)
0000 PTTY EQU 00000000B ; PUNCH = TTY
0010 PPTP EQU 00010000B ; PUNCH = PTP
0020 PUSE1 EQU 00100000B ; USER DEFINED PUNCH (1)
0030 PUSE2 EQU 00110000B ; USER DEFINED PUNCH (2)
0000 LTTY EQU 00000000B ; LIST = TTY
0040 LCRT EQU 01000000B ; LIST = CRT
0080 LUSE1 EQU 10000000B ; LIST = LPT
00C0 LUSE2 EQU 11000000B ; USER DEFINED LIST
;
; USER DEFINED DEVICE ENTRY POINTS
;
3700 C1LOC EQU 3700H ; USER CONSOLE INPUT
3703 C0LOC EQU 3703H ; USER CONSOLE OUTPUT
3706 R1LOC EQU 3706H ; USER READER 1
3709 R2LOC EQU 3709H ; USER READER 2
370C P1LOC EQU 370CH ; USER PUNCH 1
370F P2LOC EQU 370FH ; USER PUNCH 2
3712 L1LOC EQU 3712H ; USER LIST (1)
3715 L2LOC EQU 3715H ; USER LIST (2)
3718 CSLUC EQU 3718H ; USER CONSOLE STATUS
;
1 IF DEBUG
1 ORG 800H ; LOCATE IN RAM FOR DEBUG
ENDIF
;
1 IF NOT DEBUG
3800 1 ORG 3800H ; LOCATE IN TOP 8 KOMS IN 16K
ENDIF
;
; BRANCH TABLE FOR I/O SYSTEM
;
3800 C32938 JMP BEGIN ; RESET ENTRY POINT
3803 C3993C JMP CI ; CONSOLE INPUT
3806 C3A53E JMP R1 ; READER INPUT
3809 C3553C JMP C0 ; CONSOLE OUTPUT
380C C37D3E JMP PU ; PUNCH OUTPUT
380F C3E33D JMP LU ; LIST OUTPUT
3812 C3D73C JMP CSTS ; CONSOLE INPUT STATUS
3815 C3B43D JMP IOCHK ; I/O SYSTEM STATUS
3818 C3B83D JMP IOSET ; SET I/O CONFIGURATION
381B C3F83D JMP MEMCK ; COMPUTE SIZE OF MEMORY

```

```

;
; POINTERS TO RAM
;
0008      RS1   EQU    8           ; RESTART 1 LOGIC
;
; STATUS BYTE FOR I/O SYSTEM
;
1         IF     NOT DEBUG
0003 1    IOBYT EQU    3           ; USE LOCATION 0003H
        ENDF
1         IF     DEBUG
1         IOBYT EQU    4           ; USE LOCATION 0004H
        ENDF
0000      INIT  EQU    0           ; INITIALLY,
;          ; CONSOLE = TTY,
;          ; READER = TTY,
;          ; PUNCH = TTY,
;          ; LIST = TTY
;
381E     0D0A3830  VERS: DB      CR,LF,'8080 V'
3822     38302056
1         IF     NOT DEBUG
3826 1    332E30      DB      VER/10+'0','.',VER MOD 10+'0'
        ENDF
;
1         IF     DEBUG
1         DB      'X.X'
        ENDF
000B     LVER  EQU    $-VERS       ; LENGTH OF SIGN-ON MESSAGE
;
; PROGRAM ENTRY POINT.
;
; COMPUTE THE TOP OF RAM MEMORY AVAILABLE IN THIS SYSTEM
; AND LOCATE THE USER REGISTER SAVE AREA, EXIT TEMPLATE, AND
; MONITOR STACK AT THE TOP OF RAM.
;
3829     BEGIN:
3829     210300      LXI    H,IOBYT   ; POINT HL AT IOBYT
382C     3600      MVI    M,INIT     ; INITIAL VALUE OF I/O
382E     2EFF      MVI    L,OFFH    ; FIND END OF MEMORY
3830
3830     BGO:
3830     25        DCR    H
3831     7E        MOV    A,M        ; FETCH CONTENTS OF MEMORY
3832     2F        CMA                ; INVERT IT
3833     77        MOV    M,A        ; ATTEMPT TO WRITE INTO MEMORY
3834     BE        CMP    M          ; IS LOCATION READ/WRITE?
3835     2F        CMA                ; INVERT AGAIN
3836     77        MOV    M,A        ; WRITE DATA BACK
3837     C23038   JNZ    BGO        ; YES, CONTINUE
383A     23        INX    H          ; POINT TO FIRST NON-RAM LOCATION
1         IF     DEBUG

```

```

1          MVI    H,2          ; SET STACK AT 200H FOR DEBUG
          ENDIF
3838 0612    MVI    B,ENDX-EXIT ; MOVE EXIT TEMPLATE TO RAM
383D 11C23F  LXI    D,ENDX
3840          BG1:
3840 1B      DCX    D
3841 1A      LDAX   D
3842 2B      DCX    H
3843 77      MOV    M,A
3844 05      DCR    B
3845 C24038  JNZ    BG1
3848 F9      SPHL                   ; SET STACK
3849 210001  LXI    H,100H
384C E5      PUSH   H
384D 2600    MVI    H,0
384F E5      PUSH   H
3850 E5      PUSH   H
3851 E5      PUSH   H          ; PUSH REGISTERS ON STACK
1          IF    NOT DEBUG    ; IF IN DEBUG MODE, DON'T SET TRAPS
3852 1 JEC3   MVI    A,(JMP RESTART)
3854 1 J20800 STA   RSI
3857 1 21083F LXI    H,RESTART          ; SET UP RESTART 1 FOR BREAKPOINT
385A 1 220900 SHLD  RSI+1          ; LOGIC
          ENDIF
;
; TYPE SIGN-ON
;
385D 211E38  LXI    H,VERS          ; ADDRESS OF MESSAGE
3860 160B    MVI    D,LVER          ; LENGTH OF MESSAGE
3862          VERO:
3862 4E      MOV    C,M
3863 23      INX    H
3864 CD553C  CALL   CU
3867 15      DCR    D
3868 C26238  JNZ    VERO
;
; MAIN COMMAND LOOP.
;
; THIS LOOP IS THE STARTING POINT OF ALL COMMAND SEQUENCES.
; IN THIS CODE INTERRUPTS ARE ENABLED, ALL I/O DEVICES
; ARE INITIALIZED, A CARRIAGE RETURN AND LINE FEED ARE TYPED,
; ALONG WITH THE PROMPT CHARACTER, '.'. WHEN A CHARACTER IS
; ENTERED FROM THE CONSOLE KEYBOARD, IT IS CHECKED FOR VALIDITY,
; THEN A BRANCH TO THE PROPER PROCESSING ROUTINE IS COMPUTED.
;
386B          START:
386B FB      EI          ; ENABLE INTERRUPTS
386C 3E08    MVI    A,11100          ; RESEI PTY, PIR, PTP,
386E 0301    OUT   TIC          ; AND PROM PROGRAMMER
3870 CD3C3C  CALL   CRLF          ; TYPE <CR>,<LF>
3873 0E2E    MVI    C, '.'

```

```

3875 CD553C      CALL    CO          ; OUTPUT A PERIOD
3878 CD733F      CALL    TI          ; GET A CHARACTER
387B D641        SUI     'A'         ; TEST FOR A-X
387D FA6B38      JM      START      ; LT A, ERROR
3880 FE18        CPI     'X'-'A'+1
3882 F2433C      JP      LER        ; GT X, ERROR
3885 87          ADD     A          ; *2
3886 219238      LXI    H,TBL     ; ADDRESS OF TABLE
3889 85          ADD     L
388A 6F          MOV     L,A         ; COMPUTE TABLE ADDRESS, PUT IN HL
388B 7E          MOV     A,M         ; GET LSB OF ADDRESS
388C 23          INX    H          ; POINT TO NEXT ADDRESS
388D 66          MOV     H,M         ; GET MSB OF ADDRESS
388E 6F          MOV     L,A         ; LSB TO L
388F 0E02        MVI    C,2         ; C IS SET UP FOR 2 PARAMETER COMMANDS
3891 E9          PCHL   ; BRANCH TO ROUTINE

```

```

;
; COMMAND BRANCH TABLE.
;
; THIS TABLE CONTAINS THE ADDRESSES OF THE ENTRY POINTS OF
; ALL THE COMMAND PROCESSING ROUTINES. NOTE THAT AN ENTRY TO 'LER'
; IS AN ERROR CONDITION, I.E., NO COMMAND CORRESPONDING TO THAT
; CHARACTER EXISTS.

```

```

3892          TBL:
3892 C238        DW     ASSIGN    ; A - ASSIGN I/O UNITS
3894 1139        DW     BNPFF      ; B - PUNCH BNPFF
3896 5A39        DW     COMP      ; C - COMPARE PROM WITH MEMORY
3898 9039        DW     DISP      ; D - DISPLAY RAM MEMORY
389A B139        DW     EOF       ; E - ENDFILE A HEXADECIMAL FILE
389C D339        DW     FILL     ; F - FILL MEMORY
389E E439        DW     GOTO     ; G - GO TO MEMORY ADDRESS
38A0 2E3A        DW     HEXN     ; H - HEXADECIMAL SUM AND DIFFERENCE
38A2 433C        DW     LER      ; I -
38A4 433C        DW     LER      ; J -
38A6 433C        DW     LER      ; K -
38A8 4B3A        DW     LOAD     ; L - LOAD BNPFF TAPE
38AA 5F3A        DW     MOVE     ; M - MOVE MEMORY
38AC 723A        DW     NULL     ; N - PUNCH NULLS FOR LEADER
38AE 433C        DW     LER      ; O -
38B0 783A        DW     PRUG     ; P - PROGRAM A 1702A PROM
38B2 433C        DW     LER      ; Q -
38B4 CB3A        DW     READ     ; R - READ HEXADECIMAL FILE
38B6 123B        DW     SUBS     ; S - SUBSTITUTE MEMORY
38B8 403B        DW     TRAN     ; T - TRANSFER A PROM TO MEMORY
38BA 433C        DW     LER      ; U -
38BC 433C        DW     LER      ; V -
38BE 5E3B        DW     WRITE    ; W - WRITE HEX TAPE
38C0 AB3B        DW     X        ; X - EXAMINE AND MODIFY REGISTERS

```

```

;
; PROCESS I/O DEVICE ASSIGNMENT COMMANDS.

```



```

;
; THIS ROUTINE MAPS SYMBOLIC DEVICE IDENTIFIERS TO BITS
; IN THE I/O STATUS BYTE (IOBIT) TO ALLOW FOR CONSOLE
; MODIFICATION OF SYSTEM I/O CONFIGURATION.
;
38C2          ASSIGN:
38C2          CD733F      CALL    T1          ; GET LOGICAL DEVICE CHARACTER
38C5          21803F      LXI     H,LTBL        ; ADDRESS OF MASTER TABLE
38C8          0E04        MVI     C,4          ; MAXIMUM OF 4 ENTRIES
38CA          AS0:
38CA          8E          CMP     H          ; TEST FOR IDENTIFYING CHARACTER
38CB          23          INX     H
38CC          CAD938      JZ      AS1          ; FOUND IT
38CF          23          INX     H          ; POINT TO NEXT ENTRY
38D0          23          INX     H
38D1          23          INX     H
38D2          0D          DCR     C          ; DECREMENT LOOP COUNT
38D3          C2CA38      JNZ     AS0          ; TRY NEXT ENTRY
38D6          C3433C      JMP     LER          ; NO MATCH, ERROR
38D9          AS1:
38D9          46          MOV     D,M        ; GET SELECT BIT MASK
38DA          23          INX     H
38DB          5E          MOV     E,M        ; GET PHYSICAL DEVICE TABLE
38DC          23          INX     H
38DD          56          MOV     D,M
38DE          EB          XCHG
38DF          AS2:
38DF          CD733F      CALL    T1          ; SCAN UNTIL '=' IS FOUND
38E2          FE3D      CPI     '='
38E4          C2DF38      JNZ     AS2
38E7          AS3:
38E7          CD733F      CALL    T1          ; DEBLANK
38EA          FE20      CPI     ' '
38EC          CAE738      JZ      AS3
38EF          0E04        MVI     C,4          ; SET TABLE LENGTH
38F1          AS4:
38F1          8E          CMP     M          ; INDEX THROUGH PHYSICAL UNIT TABLE
38F2          23          INX     H          ; COMPARE DEVICE CHAR WITH LEGAL VALUES
38F3          CAFE38      JZ      AS5          ; RETURN WITH HL -> DEVICE SELECT BITS
38F6          23          INX     H
38F7          0D          DCR     C
38F8          C2F138      JNZ     AS4          ; CONTINUE LOOKUP
38FB          C3433C      JMP     LER          ; ERROR RETURN
38FE          AS5:
38FE          CD733F      CALL    T1
3901          FE0D      CPI     CR
3903          C2FE38      JNZ     AS5          ; LOOP UNTIL CR ENCOUNTERED
3906          4E          MOV     C,M        ; GET DEVICE SELECT BITS
3907          210300      LDI     H,IOBIT    ; GET I/O STATUS
390A          7E          MOV     A,M
390B          A0          ANA     B          ; CLEAR FIELD

```

```

390C B1          ORA      C          ; SET NEW STATUS
390D 77          MOV      M,A       ; RETURN TO MEMORY
390E C36B38     JMP      START

;
; PUNCH A BNPf TAPE.
;
; THIS ROUTINE EXPECTS TWO HEXADECIMAL PARAMETERS TO BE
; ENTERED FROM THE KEYBOARD AND INTERPRETS THEM AS
; THE BOUNDS OF A MEMORY AREA TO BE PUNCHED ON THE
; ASSIGNED PUNCH DEVICE IN BNPf FORMAT. THE TAPE
; PRODUCED IS FORMATTED WITH 4 BNPf 8-BIT WORDS PER
; LINE, WITH A REFERENCE ADDRESS IN DECIMAL PRECEDING
; EACH LINE.
;
3911          BNPf:
3911 CD703D     CALL     EXPR          ; GET TWO ADDRESSES
3914 CDCD3C     CALL     CRLF
3917 CDD73D     CALL     LEAD
391A D1        POP      D          ; GET HIGH ADDRESS
391B E1        POP      H          ; GET LOW ADDRESS
391C          BNO:
391C E5        PUSH     H
391D D5        PUSH     D
391E CD5B3E     CALL     PEOL          ; PUNCH CR,LF
3921 0620     MVI      B,' '        ; ZERO SUPPRESSION CHARACTER
3923 111027     LXI      D,10000     ; PUNCH ADDRESS IN DECIMAL
3926 CD2E3D     CALL     DIGIT
3929 11E803     LXI      D,1000
392C CD2E3D     CALL     DIGIT
392F 116400     LXI      D,100
3932 CD2E3D     CALL     DIGIT
3935 1E0A     MVI      E,10
3937 CD2E3D     CALL     DIGIT
393A 1E01     MVI      E,1
393C 0630     MVI      B,'0'        ; FORCE AT LEAST 1 ZERO
393E CD2E3D     CALL     DIGIT
3941 0E20     MVI      C,' '
3943 CD7D3E     CALL     PU
3946 D1        POP      D
3947 E1        POP      H
3948          BN1:
3948 CD4D3D     CALL     ENCODE        ; ENCODE A MEMORY BYTE INTO BNPf
394B CDA43D     CALL     HILU
394E DA723A     JC      NULL          ; ALL DONE, PUNCH TRAILER AND RETURN
3951 7D        MOV      A,L
3952 E603     ANI      03H          ; PUNCH CR,LF, ADDRESS ON MULTIPLE OF 4
3954 C24839     JNZ     BN1
3957 C31C39     JMP      BNO

;
; COMPARE PROM WITH MEMORY.
;

```

```

; THIS ROUTINE EXPECTS ONE HEXADECIMAL PARAMETER WHICH
; IT INTERPRETS AS A MEMORY ADDRESS. THE ROUTINE
; COMPARES THE PROM IN THE FRONT PANEL SOCKET WITH A
; 256 BYTE AREA OF MEMORY POINTED TO BY THE INPUT PARAMETER.
; ALL DIFFERENCES BETWEEN THE PROM AND THE MEMORY AREA
; ARE DISPLAYED IN THE FOLLOWING FORMAT:
;
; <MEM ADDRESS> <MEM CONTENTS> <CORRESPONDING PROM CONTENTS>
;
395A      COMP:
395A      00          DCR      C
395B      C0703D     CALL     EXPR      ; GET ONE ADDRESS
395E      21          POP      H          ; LOAD HL
395F      1E00       MVI      E,0       ; COUNT/PROM ADDRESS
3961      CM0:
3961      3E00       MVI      A,ENB     ; ENABLE PROM PROGRAMMER
3963      D301       OUT      PROMC     ; SET PROM ADDRESS
3965      7B          MOV      A,E       ; INVERT ADDRESS
3966      2F          CMA
3967      D302       OUT      PAD
3969      C0253D     CALL     DELAY    ; WAIT FOR 6-76 BOARD IO LATCH DATA
396C      D302       IN        PD1       ; GET PROM DATA
396E      2F          CMA
396F      4E          CMP      M          ; COMPARE WITH MEMORY
3970      CA8839     JZ        CM1       ; COMPARE
3973      F5          PUSH     PSW
3974      C0CD3C     CALL     CRLF
3977      CDBF3D     CALL     LADR      ; PRINT MEMORY ADDRESS
397A      C0533C     CALL     BLK
397D      7E          MOV      A,M
397E      C0C73D     CALL     LBYTE    ; PRINT RAM DATA
3981      C0533C     CALL     BLK
3984      F1          POP      PSW       ; RETRIEVE DATA
3985      C0C73D     CALL     LBYTE    ; PRINT PROM DATA
3988      CM1:
3988      23          INA      H
3989      1C          INR      E          ; ADJUST PROM ADDRESS
398A      C26139     JNZ      CM0
398D      C36B38     JMP      START

;
; DISPLAY MEMORY IN HEX ON CONSOLE DEVICE.
;
; THIS ROUTINE EXPECTS TWO HEXADECIMAL PARAMETERS SPECIFYING
; THE BOUNDS OF A MEMORY AREA TO BE DISPLAYED ON THE
; CONSOLE DEVICE. THE MEMORY AREA IS DISPLAYED 16 BYTES
; PER LINE, WITH THE MEMORY ADDRESS OF THE FIRST BYTE
; PRINTED FOR REFERENCE. ALL LINES ARE BLOCKED INTO INTEGRAL
; MULTIPLES OF 16 FOR CLARITY, SO THAT THE FIRST AND LAST
; LINES MAY BE LESS THAN 16 BYTES IN ORDER TO SYNCHRONIZE THE
; DISPLAY.
;

```

```

3990          DISP:
3990 CD703D      CALL   EXPR      ; GET TWO ADDRESSES
3993 D1         POP    D          ; GET HIGH ADDRESS
3994 E1         POP    H          ; GET LOW ADDRESS
3995
3995 CD0D3C      CALL   CRLF      ;
3998 CDBF3D      CALL   LADR      ; PRINT MEMORY ADDRESS
399B
399B CD533C      CALL   BLK       ; PRINT SPACE
399E 7E         MOV    A,M
399F CDC73D      CALL   LBYTE     ; PRINT DATA
39A2 CDA43D      CALL   HILU     ; TEST FOR COMPLETION
39A5 DA6B38      JC     START
39A8 7D         MOV    A,L
39A9 E60F      ANI   0FH        ; PRINT CR,LF,ADDRESS ON MULTIPLE OF 16
39AB C29B39      JNZ   D11
39AE C39539      JMP    DIU

```

```

;
; END OF FILE COMMAND.
;

```

```

; THIS ROUTINE PRODUCES A TERMINATION RECORD WHICH PROPERLY
; COMPLETES A HEXADECIMAL FILE CREATED BY 'W' COMMANDS. IT
; EXPECTS ONE HEXADECIMAL PARAMETER WHICH IS ENCODED IN THE
; TERMINATION RECORD IN THE LOAD ADDRESS FIELD AND SPECIFIES
; THE ENTRY POINT OF THE FILE CREATED. A SUBSEQUENT 'R' COMMAND
; WILL LOAD THE FILE CREATED AND TRANSFER CONTROL TO THE
; ENTRY POINT SPECIFIED IF IT IS NON-ZERO.
;

```

```

3981          EOF:
3981 0D         DCR    C          ; GET ONE PARAMETER
3982 CD703D      CALL   EXPR
3985 CD5B3E      CALL   PEOL     ; PUNCH CR,LF
3988 0E3A      MVI    C,
398A CD7D3E      CALL   PO
398D AF        XRA    A          ; CLEAR CHECKSUM
398E 57         MOV    D,A
398F CD303E      CALL   PBYTE     ; OUTPUT RECORD LENGTH
39C2 E1         POP    H
39C3 CD283E      CALL   PADR     ; PUNCH EXECUTION ADDRESS
39C6 3E01      MVI    A,1        ; RECORD TYPE 1
39C8 CD303E      CALL   PBYTE
39CB AF        XRA    A
39CC 92         SUB    D          ; OUTPUT CHECKSUM
39CD CD303E      CALL   PBYTE
39D0 C3723A      JMP    NULL      ; PUNCH TRAILER AND RETURN

```

```

;
; FILL RAM MEMORY BLOCK WITH CONSTANT.
;

```

```

; THIS ROUTINE EXPECTS THREE HEXADECIMAL PARAMETERS, THE
; FIRST AND SECOND (16 BITS) ARE INTERPRETED AS THE BOUNDS
; OF A MEMORY AREA TO BE INITIALIZED TO A CONSTANT VALUE,

```

```

; THE THIRD PARAMETER (8 BITS) IS THAT VALUE.
;
39D3          ; FILL:
39D3 UC      INR   C           ; GET 3 PARAMETERS
39D4 CD703D  CALL  EXPR
39D7 C1      POP   B           ; GET DATA IN C
39D8 D1      POP   D           ; GET HIGH ADDRESS
39D9 E1      POP   H           ; GET LOW ADDRESS
39DA          ; F10:
39DA 71      MOV   M,C         ; STORE CONSTANT IN MEMORY
39DB CDA43D  CALL  H10        ; TEST FOR COMPLETION
39DE D2DA39  JNC   F10        ; CONTINUE LOOPING
39E1 C36B38  JMP   START

;
; GO TO <ADDRESS>, OPTIONALLY SET BREAKPOINTS.
;
; THE G COMMAND IS USED FOR TRANSFERRING CONTROL FROM THE
; MONITOR TO A USER PROGRAM. IT HAS SEVERAL MODES OF
; OPERATION.
;
; IF ONE HEXADECIMAL PARAMETER IS ENTERED, IT IS INTERPRETED
; AS THE ENTRY POINT OF THE USER PROGRAM AND A TRANSFER TO
; THAT LOCATION IS EXECUTED.
;
; IF ADDITIONAL (UP TO 2) PARAMETERS ARE ENTERED, THESE ARE
; CONSIDERED "BREAKPOINTS", I.E., LOCATIONS WHERE
; CONTROL IS TO BE RETURNED TO THE MONITOR, IF THEY ARE
; ENCOUNTERED.
;
; IF THE FIRST PARAMETER IS NOT ENTERED, THE STORED VALUE
; OF THE USER'S PROGRAM COUNTER (REGISTER P) IS USED AS
; THE USER PROGRAM ENTRY POINT.
;
39E4          ; GOTO:
39E4 CD4B3E  CALL  PCHK        ; GET A CHARACTER
39E7 DA263A  JC    GO3         ; CR ENTERED, EXIT
39EA CAFE39  JZ    GO0         ; DON'T MODIFY PC
39ED CD9C3D  CALL  EAF         ; GET NEW PC VALUE
39F0 D1      POP   D
1          +   FETCH  PL0C
39F1 1 211300 +   LXI   H,PL0C        ; IN THE STACK
39F4 1 39    +   DAD   SP
39F5 74      MOV   M,D         ; STORE MODIFIED PC IN RAM
39F6 2B      DCX   H
39F7 73      MOV   M,E
39F8 78      MOV   A,B         ; RETRIEVE DELIMITER CHARACTER
39F9 FE0D    CPI   CR
39FB CA263A  JZ    GO3         ; NO TRAPS TO BE SET
39FE          ; G00:
39FE 1602    MVI   D,2         ; SET MAXIMUM OF TWO TRAPS
1          +   FETCH  TL0C

```

```

3A00 1 211400 + LXI H,TLOC ; IN THE STACK
3A03 1 39 + DAD SP
3A04 G01:
3A04 E5 PUSH H ; SAVE ADDRESS OF TRAP AREA
3A05 0E01 MVI C,1
3A07 CD703D CALL EXPR ; GET A TRAP ADDRESS
3A0A 58 MOV E,B ; SAVE DELIMITER CHARACTER
3A0B C1 POP B ; GET ADDRESS IN BC
3A0C E1 POP H
3A0D 78 MOV A,B
3A0E 81 DRA C
3A0F CA1C3A JZ G02 ; DON'T ALLOW A TRAP AT 0
3A12 71 MOV M,C ; PUT TRAP ADDRESS AWAY
3A13 23 INX H
3A14 70 MOV M,B
3A15 23 INX H
3A16 0A LDAX B ; FETCH OPCODE
3A17 77 MOV M,A ; PUT IN TRAP AREA
3A18 23 INX H
3A19 JECF MVI A,(HST 1) ; RESTART 1
3A1B 02 STAX B ; SET TRAP IN MEMORY
3A1C G02:
3A1C 7B MOV A,E ; TEST DELIMITER CHARACTER
3A1D FE0D CPI CR
3A1F CA263A JZ G03 ; ALL DONE
3A22 15 DCH D
3A23 C2043A JNZ G01 ; GO GET NEXT TRAP
3A26 G03:
3A26 CDCD3C CALL CRLF
1 + FETCH B
3A29 1 210800 + LXI H,00008H ; IN THE STACK
3A2C 1 39 + DAD SP
3A2D E9 PCHL ; TAKE THE BRANCH
;
; COMPUTE HEXADECIMAL SUM AND DIFFERENCE.
;
; THIS ROUTINE EXPECTS TWO HEXADECIMAL PARAMETERS.
; IT COMPUTES THE SUM AND DIFFERENCE OF THE TWO VALUES
; AND DISPLAYS THEM ON THE CONSOLE DEVICE AS FOLLOWS:
;
; <P1+P2> <P1-P2>
;
HEXN:
3A2E CD703D CALL EXPR ; GET TWO NUMBERS
3A31 D1 POP D
3A32 E1 POP H
3A33 CDCD3C CALL CRLF
3A36 E5 PUSH H
3A37 19 DAD D ; COMPUTE HL+DE
3A38 CDBF3D CALL LADR ; DISPLAY SUM
3A3B CD533C CALL BLA ; TYPE A SPACE

```

```

3A3E E1      PUP      H
3A3F 7D      MOV      A,L      ; COMPUTE HL-DE
3A40 93      SUB      E
3A41 6F      MOV      L,A
3A42 7C      MOV      A,H
3A43 9A      SUB      D
3A44 67      MOV      H,A
3A45 C0BF3D   CALL    LADR      ; DISPLAY DIFFERENCE
3A46 C36B38   JMP     START

;
; LOAD A BNPF TAPE INTO RAM MEMORY.
;
; THIS ROUTINE EXPECTS TWO HEXADECIMAL PARAMETERS AND
; INTERPRETS THEM AS BOUNDS OF A MEMORY AREA TO BE
; LOADED BY BNPF DATA TO BE READ FROM THE READER.
; IT IS ASSUMED THAT ENOUGH DATA IS AVAILABLE IN THE
; TAPE TO BE READ TO SATISFY THE MEMORY BOUNDS ENTERED.
; IF END OF TAPE IS ENCOUNTERED BEFORE THE MEMORY BOUNDS
; ARE SATISFIED, THIS ROUTINE WILL TERMINATE ON AN ERROR
; CONDITION (SEE R1X), BUT ALL DATA READ BEFORE THE END
; OF TAPE WAS ENCOUNTERED WILL BE LOADED.
;
3A4B      LOAD:
3A4B C0703D   CALL    EXPR      ; GET TWO ADDRESSES
3A4E C0CD3C   CALL    CRDF
3A51 D1      PUP      D      ; GET HIGH ADDRESS
3A52 E1      PUP      H      ; GET LOW ADDRESS
3A53      LOU:
3A53 C0FC3C   CALL    DECODE    ; CONVERT BNPF, STORE IN MEMORY
3A56 C0A43D   CALL    HILU      ; TEST FOR COMPLETION
3A59 D2533A   JNC     LOU       ; KEEP GOING
3A5C C36B38   JMP     START

;
; MOVE A BLOCK OF MEMORY.
;
; THIS ROUTINE EXPECTS THREE HEXADECIMAL PARAMETERS FROM THE
; CONSOLE. THE FIRST AND SECOND PARAMETERS ARE THE BOUNDS OF
; THE MEMORY AREA TO BE MOVED, THE THIRD PARAMETER IS THE
; STARTING ADDRESS OF THE DESTINATION AREA.
;
3A5F      MOVE:
3A5F 0C      INR      C      ; GET THREE ADDRESSES
3A60 CD703D   CALL    EXPR
3A63 C1      PUP      B      ; DESTINATION
3A64 D1      PUP      D      ; SOURCE END
3A65 E1      PUP      H      ; SOURCE BEGIN
3A66      MOV:
3A66 7E      MOV      A,M      ; GET A DATA BYTE
3A67 02      STA    B      ; STORE AT DESTINATION
3A68 03      INR      B      ; MOVE DESTINATION POINTER
3A69 C0A43D   CALL    HILU      ; TEST FOR COMPLETION

```

```

3A6C D2663A      JNC  MVO
3A6F C36B38      JMP  START
;
; PUNCH LEADER OR TRAILER.
;
; THIS ROUTINE PUNCHES 60 NULL CHARACTERS ON THE DEVICE ASSIGNED
; AS THE PUNCH. IT IS BRANCHED TO BY THE 'B' AND 'E' COMMANDS
; AS WELL AS BEING INVOKED BY THE 'N' COMMAND.
;
3A72          NULL:
3A72 CDD73D      CALL  LEAD
3A75 C36B38      JMP  START
;
; PROGRAM A 1702A PROM WITH FAST ALGORITHM
; (20.48 TO 409.6 SECONDS)
;
; THIS ROUTINE EXPECTS THREE HEXADECIMAL PARAMETERS FROM THE CONSOLE.
; THE FIRST AND SECOND ARE THE BOUNDS OF A MEMORY AREA TO BE
; REPRODUCED IN THE 1702A PROM IN THE FRONT PANEL SOCKET. THE THIRD
; PARAMETER IS THE ADDRESS IN THE PROM (8 BITS) WHERE THE DUPLICATION
; IS TO COMMENCE. THE ALGORITHM USED IN THIS ROUTINE TAKES ADVANTAGE
; OF THE FACT THAT MOST PROMS MAY BE PROGRAMMED IN A SMALL FRACTION
; OF THE TIME IT WOULD TAKE UNDER WORST CASE CONDITIONS, THEREFORE
; GREATLY REDUCING PROGRAMMING TIME FOR MOST PROMS. THE WIDE VARIATION
; IN TIMES QUOTED IS DUE TO THE ALLOWABLE RANGE BETWEEN BEST AND WORST
; CASE PROGRAMMING TIMES.
;
3A78          PRUG:
3A78 0C          INR  C
3A79 CD703D      CALL  EXPR      ; HL = TOP AFTER RETURN
3A7C CDCD3C      CALL  CRLF
3A7F C1          POP  B      ; C ← PROM ADDRESS
3A80 D1          POP  D      ; HIGH ADDRESS
3A81 E1          POP  H      ; LOW ADDRESS
3A82          PRU:
3A82 3E00        MVI  A,ENB
3A84 D301        OUT  PROMC      ; ENABLE PROM PROGRAMMER
3A86 79          MOV  A,C
3A87 D3FF        OUT  OFFH      ; DISPLAY ADDRESS
3A89 2F          CMA
3A8A D302        OUT  PAD      ; PROM ADDRESS
3A8C DB02        IN   PDI
3A8E 2F          CMA
3A8F BE          CMP  M
3A90 CAC13A      JZ   PR4      ; DON'T HAVE TO PROGRAM THE LOC
3A93 06F0        MVI  B,-16     ; SET MAX TRIES = 16
3A95          PR1:
3A95 CD653E      CALL  PGRM      ; PULSE AND DELAY 16 MS.
3A98 DB02        IN   PDI      ; READ VALUE
3A9A 2F          CMA
3A9B BE          CMP  M      ; COMPARE WITH DESIRED

```



```

3A9C CAB43A      JZ      PR2      ; GOT IT, NOW PULSE 4*N MORE TIMES
3A9F 04         INR      B      ; INCREMENT COUNT
3AA0 C2953A     JNZ     PR1      ; KEEP GOING
3AA3 C5         PUSH    B      ; ERROR OUT
3AA4 0E24      MVI     C,'S'
3AA6 CD553C     CALL    CD      ; PRINT ERROR FLAG
3AA9 CD533C     CALL    BLK     ; RESTORE PROM ADDRESS IN C
3AAC C1         POP     B
3AAD 79        MOV     A,C
3AAE CDC73D     CALL    LBYTE   ; DISPLAY IT
3AB1 C3433C     JMP     LER     ; ERROR EXIT
3AB4          PR2:
3AB4 78        MOV     A,B      ; MOVE COUNT RESIDUE TO A
3AB5 C611      ADI     17      ; ACTUAL COUNT OF TRIES REQUIRED
3AB7 87        ADD     A      ; COUNT = COUNT * 2
3AB8 87        ADD     A      ; COUNT = COUNT * 4
3AB9 47        MOV     B,A
3ABA          PR3:
3ABA CD653E     CALL    PGRM    ; OVERPROGRAM 4*N TIMES
3ABD 05        DCR     B
3ABE C2BA3A     JNZ     PR3
3AC1          PR4:
3AC1 0C        INR     C      ; INCREMENT PROM ADDRESS
3AC2 CDA43D     CALL    HILU
3AC5 D2823A     JNC     PRO     ; CONTINUE
3AC8 C36B38     JMP     START   ; ALL DONE
;
; READ ROUTINE.
;
; THIS ROUTINE READS A HEXADECIMAL FILE FROM THE ASSIGNED
; READER DEVICE AND LOADS IT INTO MEMORY. ONE HEXADECIMAL
; PARAMETER IS EXPECTED. THIS PARAMETER IS A BIAS ADDRESS
; TO BE ADDED TO THE MEMORY ADDRESS OF EACH DATA BYTE ENCOUNTERED.
; IN THIS WAY, HEXADECIMAL FILES MAY BE LOADED INTO MEMORY
; IN AREAS OTHER THAN THAT FOR WHICH THEY WERE ASSEMBLED OR COMPILED.
; ALL RECORDS READ ARE CHECKSUMMED AND COMPARED AGAINST THE
; CHECKSUM IN THE RECORD. IF A CHECKSUM ERROR (OR TAPE READ ERROR)
; OCCURS, THE ROUTINE TAKES AN ERROR EXIT. NORMAL LOADING IS
; TERMINATED WHEN A RECORD OF LENGTH 0 IS ENCOUNTERED. THIS IS
; INTERPRETED AS AN END OF FILE RECORD AND THE LOAD ADDRESS
; FIELD OF THAT RECORD IS TAKEN TO BE THE ENTRY POINT OF THE
; PROGRAM (IF IT IS NON-ZERO).
;
3ACB          READ:
3ACB 0D        DCR     C      ; GET ONE ADDRESS
3ACC CD703D     CALL    EXPR
3ACF          READ:
3ACF E1        POP     H      ; GET BIAS ADDRESS
3AD0 E5        PUSH    H
3AD1 CDF3E     CALL    HIX
3AD4 063A      MVI     B,':'

```

```

3AD6 90          SUB      B
3AD7 C2CF3A     JNZ     RED0          ; SCAN TO RECORD MARK
3ADA 57          MOV     D,A          ; CLEAR CHECKSUM
3ADB CD823C     CALL    BYTE
3ADE CA013B     JZ      RED2          ; ZERO RECORD LENGTH, ALL DONE
3AE1 5F          MOV     E,A          ; E <- RECORD LENGTH
3AE2 CD823C     CALL    BYTE          ; GET MSB OF LOAD ADDRESS
3AE5 F5          PUSH   PSW          ; SAVE IT
3AE6 CD823C     CALL    BYTE          ; GET LSB OF LOAD ADDRESS
3AE9 C1          POP     B          ; RETRIEVE MSB, PUT IN B
3AEA 4F          MOV     C,A
3AEB 09          DAD     B          ; BIAS ADDRESS + LOAD ADDRESS -> HL
3AEC CD823C     CALL    BYTE          ; RECORD TYPE
3AEF          RED1:
3AEF CD823C     CALL    BYTE          ; READ DATA
3AF2 77          MOV     M,A          ; PUT IN MEMORY
3AF3 23          INX     H
3AF4 1D          DCR     E
3AF5 C2EF3A     JNZ     RED1          ; LOOP UNTIL DONE
3AF8 CD823C     CALL    BYTE          ; READ CHECKSUM
3AFB C2433C     JNZ     LER          ; CHECKSUM ERRJR
3AFE C3CF3A     JMP     RED0          ; GET ANOTHER RECORD
3B01          RED2:
3B01 CD823C     CALL    BYTE          ; GET MSB OF TRANSFER ADDRESS
3B04 67          MOV     H,A
3B05 CD823C     CALL    BYTE
3B08 6F          MOV     L,A
3B09 B4          ORA     H
3B0A CA0E3B     JZ      RED3          ; IF TRANSFER ADDRESS = 0, RETURN TO KB
3B0D E9          PCHL
3B0E          RED3:
3B0E E1          POP     H
3B0F C36B3B     JMP     START
;
; SUBSTITUTE MEMORY CONTENTS ROUTINE.
;
; THIS ROUTINE EXPECTS ONE PARAMETER FROM THE CONSOLE, FOLLOWED
; BY A SPACE. THE PARAMETER IS INTERPRETED AS A MEMORY LOCATION
; AND THE ROUTINE WILL DISPLAY THE CONTENTS OF THAT LOCATION,
; FOLLOWED BY A DASH (-). TO MODIFY MEMORY, TYPE IN THE NEW DATA
; FOLLOWED BY A SPACE OR A CARRIAGE RETURN. IF NO MODIFICATION
; OF THE LOCATION IS REQUIRED, TYPE ONLY A SPACE OR CARRIAGE RETURN.
; IF A SPACE WAS LAST TYPED, THE NEXT MEMORY LOCATION WILL BE DISPLAYED
; AND MODIFICATION OF IT IS ALLOWED. IF A CARRIAGE RETURN WAS ENTERED,
; THE COMMAND IS TERMINATED.
;
SUBS:
3B12 0D          DCR     C
3B13 CD703D     CALL    EXPR          ; GET ONE ADDRESS
3B16 CD4E3E     CALL    P2C
3B19 DA433C     JC      LER

```

```

3B1C E1 POP H
3B1D SUU:
3B1D 7E MOV A,M
3B1E CDC73D CALL LBYTE ; DISPLAY DATA
3B21 0E2D MVI C,'-'
3B23 CD553C CALL CO
3B26 CD4B3E CALL PCHK
3B29 DA6B38 JC START ; CR ENTERED, RETURN TO COMMAND MODE
3B2C CA3C3B JZ SU1 ; SPACE ENTERED, SPACE BY
3B2F E5 PUSH H ; SAVE MEMORY ADDRESS
3B30 CD9C3D CALL EXF ; GET NEW VALUE
3B33 D1 POP D ; E = VALUE
3B34 E1 POP H ; RESTORE MEMORY ADDRESS
3B35 73 MOV M,E ; STORE NEW VALUE
3B36 78 MOV A,B ; TEST DELIMITER
3B37 FE0D CPI CR
3B39 CA6B38 JZ START ; CR ENTERED AFTER LAST SUBSTITUTION
3B3C SU1:
3B3C 23 INX H
3B3D C31D3B JMP SUU
;
; TRANSFER CONTENTS OF A PROM TO MEMORY.
;
; THIS ROUTINE EXPECTS ONE HEXADECIMAL PARAMETER WHICH
; IT INTERPRETS AS THE LOCATION IN MEMORY WHERE A COPY OF THE
; PROM IN THE FRONT PANEL IS TO BE STORED. THIS COPY IS ALWAYS
; 256 BYTES IN LENGTH.
;
3B40 TRAN:
3B40 0D DCR C
3B41 CD703D CALL EXPR ; GET ONE ADDRESS
3B44 E1 POP H ; HL = MEM ADR
3B45 1E00 MVI E,0 ; COUNT/PROM ADDRESS
3B47 TRU:
3B47 3E00 MVI A,ENB
3B49 D301 OUT PROMC ; ENABLE PROM PROGRAMMER
3B4B 7B MOV A,E
3B4C 2F CMA ; INVERT ADDRESS
3B4D D302 OUT PAD ; SET PROM ADDRESS
3B4F CD253D CALL DELAY ; WAIT FOR 6-76 BOARD TO LATCH DATA
3B52 DB02 IN PDI ; GET PROM DATA
3B54 2F CMA
3B55 77 MOV M,A ; PUT IN MEMORY
3B56 23 INX H ; BUMP MEMORY POINTER
3B57 1C INR E ; BUMP PROM POINTER
3B58 C2473B JNZ TRU ; GET ANOTHER BYTE
3B5B C36B38 JMP START
;
; WRITE ROUTINE.
;
; THIS ROUTINE EXPECTS TWO HEXADECIMAL PARAMETERS WHICH ARE

```

```

; INTERPRETED AS THE BOUNDS OF A MEMORY AREA TO BE ENCODED
; INTO HEXADECIMAL FORMAT AND PUNCHED ON THE ASSIGNED PUNCH
; DEVICE.
;
WRITE:
3B5E          CD703D      CALL   EXPR           ; GET TWO ADDRESSES
3B61          CDCD3C      CALL   CRLF
3B64          D1          POP    D           ; GET HIGH ADDRESS
3B65          E1          POP    H           ; GET LOW ADDRESS
3B66          WR10:
3B66          7D          MOV    A,L
3B67          C610        ADI    16
3B69          4F          MOV    C,A
3B6A          7C          MOV    A,H
3B6B          CE00        ACI    0
3B6D          47          MOV    B,A
3B6E          7B          MOV    A,E
3B6F          91          SUB    C
3B70          4F          MOV    C,A
3B71          7A          MOV    A,D
3B72          98          SBB    B
3B73          DA7B3B      JC     WR11           ; RECORD LENGTH = 16
3B76          3E10        MVI    A,16
3B78          C37E3B      JMP    WR12
3B7B          WR11:
3B7B          79          MOV    A,C           ; LAST RECORD
3B7C          C611        ADI    17
3B7E          WR12:
3B7E          B7          ORA    A
3B7F          CA6B38      JZ     START
3B82          D5          PUSH   D           ; SAVE HIGH ADDRESS
3B83          5F          MOV    E,A           ; E = LENGTH
3B84          1600        MVI    D,0           ; CLEAR CHECKSUM
3B86          CD5B3E      CALL   PEUL           ; PUNCH CR,LF
3B89          0E3A        MVI    C,':'
3B8B          CD7D3E      CALL   PO
3B8E          7B          MOV    A,E
3B8F          CD303E      CALL   PBYTE           ; PUNCH LENGTH
3B92          CD283E      CALL   PADR           ; PUNCH ADDRESS
3B95          AF          XRA    A
3B96          CD303E      CALL   PBYTE           ; PUNCH RECORD TYPE
3B99          WR13:
3B99          7E          MOV    A,M
3B9A          23          INX    H
3B9B          CD303E      CALL   PBYTE           ; PUNCH DATA
3B9E          1D          DCR    E           ; DECREMENT LENGTH
3B9F          C2993B      JNZ   WR13           ; CONTINUE LOOPING
3BA2          AF          XRA    A
3BA3          92          SUB    D
3BA4          CD303E      CALL   PBYTE           ; PUNCH CHECKSUM
3BA7          D1          POP    D           ; RESTORE HIGH ADDRESS

```

```

3BA8 C3663B JMP WR10
;
; EXAMINE AND MODIFY CPU REGISTERS.
;
; THIS ROUTINE ALLOWS THE OPERATOR TO EXAMINE AND/OR MODIFY
; THE CONTENTS OF THE USER PROGRAM'S REGISTERS. THE REGISTER
; VALUES WERE STORED AS A RESULT OF A PREVIOUS BREAKPOINT AND
; WILL BE RESTORED TO THE USER PROGRAM DURING A SUBSEQUENT 'G'
; COMMAND.
;
X1:
3BA8 CU733F CALL II ; GET REGISTER IDENTIFIER
3BAE 21C23F LXI H,ACTBL ; POINT TO ACCESS TABLE
3BB1 FE0D CPI CR
3BB3 CA143C JZ X0 ; FULL REGISTER DISPLAY
3BB6 X0:
3BB6 BE CMP M
3BB7 CAC73B JZ X1
3BBA F5 PUSH PSW ; SAVE THE CHARACTER
3BBB 7E MOV A,M
3BBC 07 ORA A ; TEST FOR END OF TABLE
3BBD FA433C JM LER
3BC0 23 INX H
3BC1 23 INX H
3BC2 23 INX H
3BC3 F1 POP PSW
3BC4 C3B63B JMP XU
3BC7 X1:
3BC7 CU533C CALL BLK
3BCA X2:
3BCA 23 INX H
3BCB 7E MOV A,M ; DISPLACEMENT
3BCC EB XCHG ; SAVE HL IN DE (HL = POINTER TO ACTBL)
3BCD 0F MOV L,A
3BCE 2000 MVI H,0
3BD0 39 DAD SP
3BD1 EB XCHG ; RESTORE HL
3BD2 23 INX H
3BD3 96 MOV B,M ; PRECISION;
3BD4 23 INX H ; POINT AT NEXT REGISTER IDENTIFIER
3BD5 1A LDAX D ; 8/16 BIT DISPLAY AND MODIFICATION
3BD6 CDC73D CALL LBYTE ; MSB OF 16 BIT REG, ALL OF 8 BIT REG
3BD7 05 DCR B
3BD8 CAE23B JZ X3 ; 8 BIT DISPLAY
3BD9 18 DCA D
3BDE 1A LDAX D
3BDF CDC73D CALL LBYTE ; LSB OF 16 BIT REG
3BE2 X3:
3BE2 04 INR B
3BE3 0E2D MVI C, '-'
3BE5 CD553C CALL CD

```

```

3BE8 CD4B3E CALL PCHK ; SKIP IF NULL ENTRY
3BE9 DA6B38 JC STAKT ; CR ENTERED, RETURN TO COMMAND MODE
3BEE CA063C JZ X5
3BF1 E5 PUSH H ; SAVE POINTER TO ACTBL
3BF2 C5 PUSH B ; SAVE PRECISION
3BF3 CD9C3D CALL EXF ; GET NEW REG VALUE
3BF6 E1 POP H
3BF7 F1 POP PSW ; A = PRECISION
3BF8 C5 PUSH B ; B = DELIMITER CHAR
3BF9 F5 PUSH PSW ; A = PRECISION
3BFA 7D MOV A,L
3BFB 12 STAX D ; STORE LSB IN REGISTER AREA
3BFC C1 POP B ; RETRIEVE PRECISION
3BFD 05 DCR B
3BFE CA043C JZ X4 ; 8 BITS ONLY
3C01 13 INX D
3C02 7C MOV A,H
3C03 12 STAX D ; STORE MSB IN REGISTER AREA
3C04 X4:
3C04 C1 POP B ; RETRIEVE DELIMITER CHAR
3C05 E1 POP H
3C06 X5:
3C06 7E MOV A,M ; TEST FOR END OF TABLE
3C07 B7 ORA A
3C08 FA6B38 JM START
3C0B 78 MOV A,B ; TEST DELIMITER
3C0C FE00 CPI CR
3C0E CA6B38 JZ START
3C11 C3CA3B JMP X2
3C14 X6: ; FULL REGISTER DISPLAY
3C14 CD0D3C CALL CRLF
3C17 X7:
3C17 CD533C CALL BLK ; OUTPUT A SPACE
3C1A 7E MOV A,M ; GET THE CHARACTER
3C1B 23 INX H ; POINT AT DISPLACEMENT
3C1C B7 ORA A ; SET CONDITION CODES
3C1D FA6B38 JM START ; ALL DONE
3C20 4F MOV C,A
3C21 CD553C CALL CO ; PRINT CHARACTER
3C24 0E3D MVI C,'='
3C26 CD553C CALL CO ; PRINT EQUAL SIGN
3C29 7E MOV A,M ; FETCH DISPLACEMENT
3C2A 23 INX H ; POINT AT PRECISION
3C2B EB XCHG ; SAVE HL IN DE
3C2C 6F MOV L,A
3C2D 2600 MVI H,0
3C2F 39 DAD SP ; COMPUTE ADDRESS OF VALUE
3C30 EB XCHG ; HL = CURRENT TABLE ADDRESS,
; DE = MEMORY ADDRESS
3C31 46 MOV B,M ; FETCH PRECISION
3C32 23 INX H ; POINT TO NEXT DISPLAY

```

```

3C33 1A          LDAX  D          ; GET LSB OF DATA
3C34 CDC73D     CALL  LBYTE     ; DISPLAY IT
3C37 05          DCR   B
3C38 CA173C     JZ    X7          ; 8 BIT PRECISION, ALL DONE
3C3B 1B          DCX   D          ; POINT TO MSB OF 16 BIT VALUE
3C3C 1A          LDAX  D          ; FETCH IT
3C3D CDC73D     CALL  LBYTE     ; DISPLAY IT
3C40 C3173C     JMP   X7          ; CONTINUE

;
; ERROR EXIT.
;
; THIS ABNORMAL EXIT IS EXECUTED FOR ALL MONITOR ERROR CONDITIONS.
; THE STACK POINTER VALUE IS RECOMPUTED BECAUSE OF THE UNKNOWN STATE
; OF THE PROCESSOR AS THE RESULT OF A COMMAND OR DATA ERROR.
;
3C43          LER:
3C43 CD013E     CALL  MEMSIZ     ; COMPUTE TOP OF MEMORY
3C46 11F8FF     LXI  D,-8
3C49 19          DAD   D
3C4A F9          SPHL
3C4B 0E2A       MVI  C,'*'
3C4D CD553C     CALL  CO
3C50 C36B38     JMP   START

;
; SUBROUTINES
;
3C53          BLK:
3C53 0E20       MVI  C,' '          ; PRINT A BLANK

;
; EXTERNALLY REFERENCED ROUTINE
; CONSOLE OUTPUT CODE, VALUE EXPECTED IN C
; A,FLAGS,C MODIFIED
; STACK USAGE: 2 BYTES
;
3C55          CO:
3C55 3A0300     LDA   10BY1     ; GET STATUS BYTE
3C58 E603       ANI  NOT CMSK  ; GET CONSOLE BITS
3C5A C2693C     JNZ  CO0       ; TEST FOR CRT
3C5D          TTYOUT:
3C5D DB01       IN    TTS          ; CONSOLE = ITY
3C5F E604       ANI  ITYBE
3C61 C25D3C     JNZ  TTYOUT    ; LOOP UNTIL READY
3C64 79          MOV  A,C
3C65 2F          CMA
3C66 D300       OUT  TIO          ; OUTPUT CHARACTER
3C68 C9          RET          ; RETURN
3C69          COO:
3C69 FE01       CPI  CCR1     ; CONSOLE = CRT?
3C6B C27A3C     JNZ  CU1       ; TEST FOR BATCH
3C6E          CRTOUT:
3C6E DB05       IN    CRTS     ; CONSOLE = CRT

```

```

3C70 E604      ANI      CRTBE
3C72 C26E3C   JNZ      CRTOUT      ; LOOP UNTIL READY
3C75 79       MOV      A,C
3C76 2F       CMA
3C77 D304     OUT      CRTU
3C79 C9       RET
3C7A          CUI:
3C7A FE02     CPI      BATCH
3C7C CAE33D   JZ       LU           ; BATCH MODE, OUTPUT = LIST
3C7F C30337   JMP      COLUC       ; BRANCH TO USER CONSOLE OUTPUT
;
; READ TWO ASCII CHARACTERS, DECODE INTO 8 BITS BINARY
;
3C82          BYTE:
3C82 C0FF3E   CALL    RIX          ; READ CHAR FROM TAPE
3C85 CD163E   CALL    NIBBLE       ; CONVERT ASCII TO HEX
3C88 07       RLC
3C89 07       RLC
3C8A 07       RLC
3C8B 07       RLC          ; SHIFT FOUR PLACES
3C8C 4F       MOV      C,A
3C8D C0FF3E   CALL    RIX
3C90 CD163E   CALL    NIBBLE       ; GET LOWER NIBBLE
3C93 B1       ORA      C
3C94 4F       MOV      C,A
3C95 82       ADD      D          ; UPDATE CHECKSUM
3C96 57       MOV      D,A
3C97 79       MOV      A,C
3C98 C9       RET          ; RETURN
;
; EXTERNALLY REFERENCED ROUTINE
; CONSOLE INPUT CODE, VALUE RETURNED IN A
; A, FLAGS MODIFIED
; STACK USAGE: 2 BYTES
;
3C99          C1:
3C99 3A0300   LDA      IOBYT       ; CONSOLE INPUT
3C9C E603      ANI      NOT CMSK   ; GET STATUS BYTE
3C9E C2AC3C   JNZ      C11        ; GET CONSOLE BITS
3CA1          TTYIN:
3CA1 DB01     IN       T1S      ; TTY STATUS PORT
3CA3 E601     ANI      TTYDA     ; CHECK FOR DATA AVAILABLE
3CA5 C2A13C   JNZ      TTYIN
3CA8 DB00     IN       T1I      ; READ THE CHARACTER
3CAA          C10:
3CAA 2F       CMA
3CAB C9       RET          ; RETURN
3CAC          C11:
3CAC FE01     CPI      CCRT    ; CONSOLE = CRT?
3CAE C2BD3C   JNZ      C12        ; TEST FOR BATCH
3CB1          CRTIN:

```



```

3CB1 0B05      IN      CRIS      ; CRT STATUS PORT
3CB3 E601      ANI      CRIDA     ; CHECK FOR DATA AVAILABLE
3CB5 C2B13C    JNZ      CR1IN     ; NOT READY, CONTINUE LOOPING
3CB8 0B04      IN      CRTI      ; READ THE CHARACTER
3CBA C3AA3C    JMP      C10
3CBD          C12:
3CB0 FE02      CPI      BATCH
3CBF CA853E    JZ       RI        ; BATCH MODE, INPUT = HEADER
3CC2 C30037    JMP      CILUC     ; CONSOLE = USER DEVICE
;
; CONVERT + BIT HEX VALUE TO ASCII CHARACTER
; INPUT      = 0...0FH
; OUTPUT     = 30H...39H, 41H...46H
;
3CC5          CONV:
3CC5 C690      ADI      90H      ; SET UP A SU THAT A-F CAUSE CARRY
3CC7 27       DAA
3CC8 CE40      ACI      40H      ; ADD IN CARRY AND ADJUST UPPER NIBBLE
3CCA 27       DAA
3CCB 4F       MOV      C,A
3CCC C9       RET        ; RETURN
;
; TYPE CARRIAGE RETURN AND LINE FEED ON CONSOLE
;
3CCD          CRLF:
3CCD 0E0D     MVI      C,CR      ; <CR>
3CCF CD553C   CALL     CU
3CD2 0E0A     MVI      C,LF      ; <LF>
3CD4 C3553C   JMP      CU
;
; EXTERNALLY REFERENCED ROUTINE
; CONSOLE INPUT STATUS CODE
; A, FLAGS MODIFIED
; STACK USAGE: 2 BYTES
;
3CD7          CS1S:
3CD7 3A0300   LDA      10BIT     ; CONSOLE INPUT STATUS
3CDA E603     ANI      001 CMSK  ; GET STATUS BYTE
3CDC C2E43C   JNZ      CS0      ; CONSOLE = TTY?
3CDF 0B01     IN      IIS      ; GET TTY STATUS
3CE1 C3EB3C   JMP      CS1
3CE4          CS0:
3CE4 FE01     CPI      CCR1
3CE6 C2F23C   JNZ      CS3
3CE9 0B05     IN      CR1S     ; GET CRI STATUS
3CEB          CS1:
3CEB E601     ANI      TTYDA
3CED 3E00     MVI      A,FALSE  ; RETURN FALSE IF NO DATA AVAILABLE
3CEF          CS2:
3CEF C0       RNZ
3CF0 2F       CMA

```

```

3CF1 C9 RET ; RETURN
3CF2 CSJ:
3CF2 FE02 CPI BATCH
3CF4 3EFF MVI A,TRUE
3CF6 CAEF3C JZ CS2
3CF9 C31837 JMP CSLUC
;
; HEAD BNPF TAPE RECORD, BUILD BYTE, STORE IN MEMORY
; IF ERROR, ABOUT CUMMAND
;
3CFC DECODE:
3CFC C0FF3E CALL RIX ; READ TAPE
3CFE FE42 CPI 'B' ; SCAN FOR 'B'
3D01 C2FC3C JNZ DECODE
3D04 3601 MVI H,1 ; INITIALIZE MEMORY
3D06 DCU:
3D06 C0FF3E CALL RIX ; GET DATA
3D09 FE4E CPI 'N' ; CHECK FOR 'N'
3D0B C21D3D JNZ DC2 ; NO, CHECK FOR 'P'
; CARRY = 0
3D0E DC1:
3D0E 7E MOV A,M ; SHIFT IN DATA BIT
3D0F 17 RAL
3D10 77 MOV M,A
3D11 D2063D JNC DCU ; IF CARRY IS SET, 8 BITS READ
3D14 C0FF3E CALL RIX ; TEST FOR REQ'D 'F'
3D17 FE46 CPI 'F'
3D19 C2433C JNZ LER
3D1C C9 RET ; RETURN
3D1D DC2:
3D1D C680 ADI -'P'
3D1F C2433C JNZ LER ; ERROR
3D22 C30E3D JMP DC1 ; CARRY IS SET
;
; 1.0 MS DELAY
;
3D25 DELAY:
3D25 C5 PUSH B
3D26 0654 MVI B,DLY
3D28 DLO:
3D28 05 DCR B
3D29 C2283D JNZ DLO
3D2C C1 POP B
3D2D C9 RET ; RETURN
;
; CONVERT BINARY NUMBER TO A STRING OF ASCII DIGITS
; HL - BINARY NUMBER
; DE - DIVISOR (DESCENDING POWERS OF 10)
; B - LEADING ZERO SUPPRESSION CHARACTER
; A,C - TEMPORARIES
;

```

```

3D2E          DIGIT:
3D2E 0E30      MVI    C,'0'          ; INITIALIZE CHARACTER
3D30          DGO:
3D30 7D        MOV    A,L            ; SUB DENUM (DE) FROM NUMERATOR (HL)
3D31 93        SUB    E
3D32 6F        MOV    L,A
3D33 7C        MOV    A,H
3D34 9A        SBB    D
3D35 67        MOV    H,A
3D36 DA3D3D   JC     DG1            ; NEGATIVE RESULT, ALL DONE
3D39 UC       INR    C              ; COUNT NUMBER OF SUBTRACTS
3D3A C3303D   JMP    DGO
3D3D          DGI:
3D3D 19        DAD    D              ; ADJUST HL
3D3E 79        MOV    A,C
3D3F FE30     CPI    '0'           ; CHECK FOR LEADING ZERO SUPPRESSION
3D41 C2483D   JNZ    DG3
3D44 48        MOV    C,B
3D45          DG2:
3D45 C37D3E   JMP    PU              ; PUNCH CHARACTER
3D48          DG3:
3D48 0630     MVI    B,'0'
3D4A C3453D   JMP    DG2
;
; ENCODE A BNPF WORD AND PUNCH IT
;
3D4D          ENCODE:
3D4D 0E42     MVI    C,'B'          ; PUNCH A 'B'
3D4F C07D3E   CALL   PU
3D52 0608     MVI    B,B              ; 8 BIT COUNT
3D54 7E        MOV    A,M          ; GET DATA
3D55          ENO:
3D55 07        RLC                ; ROTATE TO SET CARRY
3D56 F5        PUSH   PSH         ; SAVE INTERMEDIATE RESULT
3D57 3E00     MVI    A,0            ; COMPUTE EITHER 'P' OR 'N'
3D59 17        RAL                ; BASED ON FOLLOWING ALGORITHM:
3D5A 17        RAL                ; CHAR = 'N' + 2*CARRY
3D5B C64E     ADI    'N'          ; CHAR = 'N' IF CARRY = 0
3D5D 4F        MOV    C,A          ; CHAR = 'P' IF CARRY = 1
3D5E C07D3E   CALL   PU
3D61 F1        POP    PSH
3D62 05        DCR    B
3D63 C2553D   JNZ    ENO
3D66 0E46     MVI    C,'F'
3D68 C07D3E   CALL   PU
3D6B 0E20     MVI    C,' '
3D6D C37D3E   JMP    PU
;
; EVALUATE EXPRESSION: <EXPR>,<EXPR>,<EXPR>
;
; THE C REGISTER CONTAINS THE NUMBER OF PARAMETERS REQUIRED

```

```

; (1,2, OR 3). PARAMETERS ARE RETURNED ON THE STACK.
;
3D70      EXPR:
3D70      210000      LXI      H,0          ; INITIAL VALUE OF PARAMETER
3D73      EXO:
3D73      CD733F      CALL     TI          ; GET A CHARACTER
3D76      EX1:
3D76      47          MOV      B,A        ; SAVE DELIMITER CHARACTER
3D77      CD163E      CALL     NIBBLE     ; CONVERT TO HEX
3D7A      DA863D      JC       EX2       ; NOT LEGAL CHAN, TREAT AS DELIMITER
3D7D      29          DAD      H          ; *2
3D7E      29          DAD      H          ; *4
3D7F      29          DAD      H          ; *8
3D80      29          DAD      H          ; *16
3D81      B5          ORA      L
3D82      6F          MOV      L,A
3D83      C3733D      JMP      EXO       ; GET ANOTHER CHARACTER
3D86      EX2:
3D86      E3          XTHL
; GET RETURN ADDRESS OFF STACK
; PUT HL ON
; REPLACE RETURN ADDRESS
3D87      E5          PUSH     H
3D88      78          MOV      A,B
3D89      CD4E3E      CALL     P2C       ; TEST DELIMITER CHARACTER
3D8C      D2943D      JNC      EX3
3D8F      0D          DCR      C          ; CR ENTERED
3D90      C2433C      JNZ      LER       ; TOO FEW PARAMS
3D93      C9          RET
3D94      EX3:
3D94      C2433C      JNZ      LER       ; ILLEGAL DELIMITER
3D97      0D          DCR      C
3D98      C2703D      JNZ      EXPR
3D9B      C9          RET
3D9C      EXF:
; ENTRY POINT FOR CONDITIONAL PARAMETERS
3D9C      0E01      MVI      C,1
3D9E      210000      LXI      H,0
3DA1      C3763D      JMP      EX1
;
; COMPARE HL WITH DE:
; IF HL < DE THEN CARRY = 0;
; IF HL = DE THEN CARRY = 0;
; IF HL > DE THEN CARRY = 1;
;
3DA4      HILO:
3DA4      23          INX      H          ; BUMP HL
3DA5      7C          MOV      A,H          ; TEST FOR HL = 0
3DA6      B5          ORA      L
3DA7      37          STC
3DA8      C8          RZ
3DA9      7B          MOV      A,B          ; DE - HL, SET/RESET CARRY
3DAA      95          SUB      L
3DAB      7A          MOV      A,D

```

```

3D8C 9C          S6B  H
3D8D C9          REI                      ; RETURN
;
; CONVERT NIBBLE IN A-REGISTER TO ASCII IN A-REGISTER
; AND PRINT ON CONSOLE DEVICE
;
3D8E          HXD:
3D8E CDC53C      CALL  CONV
3D8F C3553C      JMP   CO
;
; EXTERNALLY REFERENCED ROUTINE
; I/O SYSTEM STATUS CODE
; STATUS BYTE RETURNED IN A
; STACK USAGE: 2 BYTES
;
3D94          IOCHK:
3D94 3A0300      LDA   IOBIT          ; GET STATUS BYTE
3D95 C9          RET                      ; RETURN
;
; EXTERNALLY REFERENCED ROUTINE
; SET I/O CONFIGURATION
; VALUE EXPECTED IN C
; STACK USAGE: 2 BYTES
;
3D98          IOSET:
3D98 E5          PUSH  H                ; SAVE HL
3D99 210300      LXI  H,IOBIT          ; POINT HL AT IOBIT
3D9A 71          MOV  M,C
3D9B E1          POP  H                ; RESTORE HL
3D9C C9          RET                      ; RETURN
;
; PRINT CONTENTS OF HL IN HEX ON CONSOLE DEVICE
;
3D9F          LADR:
3D9F 7C          MOV  A,H                ; PRINT MSB
3DA0 CDC73D      CALL  LBYTE
3DA1 7D          MOV  A,L                ; PRINT LSB
3DA2 C3C73D      JMP  LBYTE
;
; LIST A BYTE AS 2 ASCII CHARACTERS
;
3DC7          LBYTE:
3DC7 F5          PUSH  PSW              ; SAVE A COPY OF A
3DC8 0F          RRC
3DC9 0F          RRC
3DCA 0F          RRC
3DCB 0F          RRC
3DCC E60F        ANI  0FH              ; UPPER 4 BITS
3DCE CDAE3D      CALL  HXD
3DD1 F1          POP  PSW              ; RETRIEVE ORIGINAL VALUE
3DD2 E60F        ANI  0FH              ; LOWER 4 BITS

```

```

3DD4 C3AE3D      JMP      HXD
;
; PUNCH 6 INCHES OF LEADER
;
3DD7          LEAD:
3DD7 063C      MVI      B,60      ; SET TO PUNCH 6 INCHES OF NULLS
3DD9          LEQ:
3DD9 0E00      MVI      C,0
3DDB CD7D3E      CALL     PO
3DDE 05        DCR      B
3DDF C2D93D      JNZ     LEQ
3DE2 C9         RET              ; RETURN
;
; EXTERNALLY REFERENCED ROUTINE
; LIST OUTPUT CODE
; VALUE EXPECTED IN C, A AND FLAGS MODIFIED
; STACK USAGE: 2 BYTES
;
3DE3          LO:
3DE3 3A0300      LDA      IOBYT      ; LIST OUTPUT
3DE6 E6C0      ANI      NOT LMSK   ; GET STATUS BYTE
3DE8 CA5D3C      JZ       TTYOUT     ; GET LIST BITS
3DEB FE40      CPI      LCKT      ; LIST = TTY
3DED CAbE3C      JZ       CRTOUT     ; LIST = CRT
3DF0 FE80      CPI      LUSE1     ; TEST FOR USER DEFINED LIST DEVICES
3DF2 CA1237      JZ       L1LOC     ; BRANCH TO USER DEVICES
3DF5 C31537      JMP      L2LOC
;
; EXTERNALLY REFERENCED ROUTINE
; RETURN ADDRESS OF END OF MEMORY TO USER
; VALUE RETURNED IN (B,A)
; STACK USAGE: 8 BYTES
;
3DF8          MEMCK:
3DF8 E5         PUSH     H
3DF9 CD013E      CALL     MEMSIZ
3DFC 44         MOV      B,H
3DFD 3EC0      MVI      A,0CUH
3DFF E1         POP      H
3E00 C9         RET
;
; FIND END OF MEMORY, SET STACK
;
3E01          MEMSIZ:
3E01 C5         PUSH     B      ; SAVE BC
3E02 21FF00      LXI     H,0FFH      ; FIND END OF MEMORY
3E05          MEMO:
3E05 25         DCR      H
3E06 7E         MOV      A,H      ; FETCH CONTENTS OF MEMORY
3E07 2F         CMA
3E08 77         MOV      M,A      ; ATTEMPT TO WRITE INTO MEMORY

```

```

3E09 BE      CMP      M      ; IS LOCATION READ/WRITE?
3E0A 2F      CMA      ; INVERT AGAIN
3E0B 77      MOV      M,A     ; WRITE DATA BACK
3E0C C2053E  JNZ      MEMO     ; YES, CONTINUE
3E0F 23      INX      H      ; POINT TO FIRST NON-RAM LOCATION
      1      IF      DEBUG
      1      MVI      H,2    ; SET STACK AT 200H FOR DEBUG
      ENDIF
3E10 01EEFF  LXI      B,EXIT-ENDA ; COMPUTE TOP OF NEW STACK
3E13 09      DAD      B
3E14 C1      POP      B      ; RESTORE BC
3E15 C9      RET      ; RETURN
;
; DECODE ASCII CHAR IN A-REGISTER INTO HEX DIGIT IN A-REGISTER
; FILTER OUT ALL CHARACTERS NOT IN THE SEQUENCE (0...9,A...F).
; RETURN CARRY = 1 FOR ILLEGAL CHARACTERS.
;
3E16 NIBBLE:
3E16 D630      SUB      '0'
3E18 DB      KC      ; FILTER OUT 0-2FH
3E19 C6E9      ADI      '0'-'9'
3E1B DB      KC      ; FILTER OUT 47H-0FFH
3E1C C606      ADI      0
3E1E F2243E    JP      N10     ; TAKE BRANCH FOR A-F
3E21 C607      ADI      7
3E23 DB      KC      ; FILTER OUT 3AH-40H
3E24 N10:
3E24 C60A      ADI      10
3E26 B7      ORA      A      ; CLEAR ERROR FLAG
3E27 C9      RET      ; RETURN
;
; PUNCH CONTENTS OF HL IN HEX ON PUNCH DEVICE
;
3E28 PAOR:
3E28 7C      MOV      A,H
3E29 C0303E  CALL     PBYTE
3E2C 7D      MOV      A,L
3E2D C3303E  JMP      PBYTE
;
; PUNCH A BYTE AS 2 ASCII CHARACTERS
;
3E30 PBYTE:
3E30 FD      PUSH     PSW
3E31 0F      RRC
3E32 0F      RRC
3E33 0F      RRC
3E34 0F      RRC
3E35 E60F    ANI      0FH
3E37 C0C53C  CALL     CONV
3E3A CD7D3E  CALL     PU
3E3D F1      POP      PSW

```

```

3E3E F5          PUSH   PSW
3E3F E60F       ANI    OFH
3E41 CDC53C    CALL   CUNV
3E44 CD7D3E    CALL   PO
3E47 F1         POP    PSW
3E48 82        ADD    D
3E49 57        MOV    D,A
3E4A C9        RET          ; RETURN
;
; TEST FOR NULL INPUT PARAMETER
;
3E4B          PCHK:
3E4B CD733F    CALL   FI          ; GET A CHARACTER
3E4E          PZC:
3E4E FE20      CPI    ','
3E50 C8       RZ
3E51 FE2C      CPI    ','
3E53 C8       RZ
3E54 FE0D      CPI    CR
3E56 37       STC
3E57 3F       CMC
3E58 C0       RNZ
3E59 37       STC
3E5A C9       RET
;
; PUNCH CR,LF
;
3E5B          PEO:
3E5B 0E0D      MVI    C,CR
3E5D CD7D3E    CALL   PU
3E60 0E0A      MVI    C,LF
3E62 C37D3E    JMP    PO
;
; PULSE A PRUM LOCATION
; HL POINTS TO DATA IN MEMORY
; PRUM ADDRESS IS ALREADY SET
;
3E65          PGRM:
3E65 7E        MOV    A,M          ; GET DATA FROM MEMORY
3E66 2F        CMA          ; INVERT IT
3E67 D303      OUT    PDU          ; OUTPUT IT
3E69 3E80      MVI    A,PRUGD    ; PULSE PRUM PROGRAMMER
3E6B D301      OUT    PRUMC
3E6D 3E00      MVI    A,PRUND
3E6F D301      OUT    PRUMC
3E71 C5        PUSH   B          ; DELAY 20 MS. FOR PROGRAMMER SETTLING
3E72 0614      MVI    B,LULY
3E74          PGO:
3E74 CD253D    CALL   DELAY
3E77 05        DCR    B
3E78 C2743E    JNZ   PGO

```



```

3E7B C1          POP      8
3E7C C9          RET

;
; EXTERNALLY REFERENCED ROUTINE
; PUNCH OUTPUT CODE, VALUE EXPECTED IN C
; A, FLAGS, AND C MODIFIED
; STACK USAGE: 2 BYTES
;
3E7D          PU:          ; PUNCH OUTPUT
3E7D 3A0300     LDA      IOBYT ; GET STATUS BYTE
3E80 E630      ANI      NOT PMSK ; GET PUNCH BITS
3E82 CA5D3C    JZ       TTYOUT ; NO, PUNCH = TTY
3E85 FE10      CPI      PPIP ; TEST FOR PTP
3E87 C29D3E    JNZ      POI ; TEST FOR USER DEVICE(S)
3E8A          PU0:        ; PUNCH = PTP
3E8A D601      IN       PTPS ; GET STATUS
3E8C E640      ANI      PRDY ; CHECK STATUS
3E8E CA8A3E    JZ       PJO ; LOOP UNTIL READY
3E91 79        MOV      A,C
3E92 D303      OUT     PTPO
3E94 3E0A      MVI     A,PTPGU ; START PUNCH
3E96 D301      OUT     PIPC
3E98 3E08      MVI     A,PTPNU ; STOP PUNCH
3E9A D301      OUT     PIPC
3E9C C9        RET
3E9D          PU1:
3E9D FE20      CPI      PUSE1
3E9F CA9C37    JZ       P1LOC
3EA2 C30F37    JMP      P2LOC

;
; EXTERNALLY REFERENCED ROUTINE
; READER INPUT CODE
; VALUE RETURNED IN A, FLAGS MODIFIED
; STACK USAGE: 8 BYTES
;
3EA5          RI:          ; READER INPUT
3EA5 E5        PUSH     H ; SAVE HL
3EA6 210300     LXI     H,IOBYT ; POINT HL AT IOBYT
3EA9 7E        MOV      A,M
3EAA E60C      ANI     NOT RMSK ; READER = PTR?
3EAC C2D13E    JNZ     RIJ ; BRANCH TO PTR ROUTINE
3EAF 3E09      MVI     A,TTYGU ; READER = TTY
3EB1 D301      OUT     TIC
3EB3 3E08      MVI     A,TTYNU
3EB5 D301      OUT     TIC
3EB7 26FA      MVI     H,1001 ; SET READER TIMEOUT TIMER
3EB9          RI0:
3EB9 D601      IN       TTS
3EBB E601      ANI     ITYDA
3EBD CACB3E    JZ       RIZ ; DATA IS READY
3EC0 C02530     CALL    DELAY ; DELAY 1.0 MS

```

```

3EC3 25          DCR      H
3EC4 C2B93E     JNZ      R10
3EC7          R11:     XRA      A
3EC7 AF         SRC      A
3EC8 37                ; SET CARRY INDICATING EOF
3EC9 E1         POP      H
3ECA C9         RET      ; RETURN
3ECB          R12:     IN       T11
3ECB DB06      CMA
3ECD 2F         ORA      A
3ECE B7                ; CLEAR CARRY
3ECF E1         POP      H
3ED0 C9         RET      ; RETURN
3ED1          R13:     RET      ; PTR ROUTINE
3ED1 FE04      CPI      RPTR
3ED3 C2F63E     JNZ      R16
3ED6 3E0C      MVI      A,PTRGO ; START PTR
3ED8 D301      OUT      P1RC
3EDA 3E08      MVI      A,PIRNU ; STOP PTR
3EDC D301      OUT      P1RC
3EDE 26FA      MVI      H,TOUT  ; SET READER TIMEOUT TIMER
3EE0          R14:     IN       PTRS
3EE0 DB01      ANI      PIRDA
3EE2 E620      JNZ      R15
3EE4 C2F13E     CALL     DELAY
3EE7 CD253D     CALL     DCR      H
3EEA 25         JNZ      R14
3EEB C2E03E     JMP      R11
3EEE C3C73E
3EF1          R15:     IN       PTR1  ; GET THE DATA
3EF1 DB03
3EF3 B7         ORA      A
3EF4 E1         POP      H
3EF5 C9         RET      ; RETURN
3EF6          R16:     POP      H
3EF6 E1         CPI      RUSE1
3EF7 FE08      JZ       R1LUC
3EF9 CA0637     JZ       R1LUC
3EFC C30937     JMP      R2LUC
;
; GET CHARACTER FROM READER, MASK OFF PARITY B11
;
R1X:
3EFF CDA53E     CALL     R1
3F02 DA433C     JC       LER      ; HEADER TIMEOUT ERROR
3F05 E67F      ANI      /FH
3F07 C9         RET      ; RETURN
;
; RESTART 1 CODE, (PROGRAMMED BREAKPOINT).
;
; THIS ROUTINE IS ENTERED VIA A RESTART 1 (RST 1) INSTRUCTION.

```

```

; THE INSTRUCTION IS ENCOUNTERED EITHER IN THE USER PROGRAM (AS
; A BREAKPOINT) OR IS INPUT VIA A CONSOLE INTERRUPT. THIS
; ROUTINE SAVES THE STATE OF THE CALLING PROCESS AND TURNS CONTROL
; OVER TO THE MONITOR.

```

```

;
; RESTART:
3F08      E5          PUSH    H          ; SAVE MACHINE STATE
3F09      D5          PUSH    D
3F0A      C5          PUSH    B
3F0B      F5          PUSH    PSW
3F0C      CD013E     CALL    MEMSIZ   ; HL = NEW STACK POINTER
3F0F      EB          XCHG
          1          +      FETCH    10   ; COMPUTE ORIGINAL STACK POINTER
3F10 1 210A00 +      LXI    H,0000AH   ; IN THE STACK
3F13 1 39          +      DAD     SP
3F14      0604      MVI     B,4      ; COUNT FOR TRANSFER OF MACHINE STATE
          ; TO STORAGE (MOVE THE STACK)

3F16      EB          XCHG
3F17      RST0:
3F17      2B          DCX     H
3F18      72          MOV     M,D
3F19      2B          DCX     H
3F1A      73          MOV     M,E
3F1B      D1          POP     D
3F1C      05          DCR     B
3F1D      C2173F     JNZ    RST0
3F20      C1          POP     B          ; GET OLD PC = B,C; OLD HL = D,E
3F21      0B          DCX     B          ; DECREMENT TO POINT AT TRAPPED CODE
3F22      F9          SPHL
          1          +      FETCH    TLUC
3F23 1 211400 +      LXI    H,TLUC   ; IN THE STACK
3F26 1 39          +      JAD     SP
3F27      7E          MOV     A,M
          ; TEST IF THIS IS A PROGRAMMED RESTART
3F28      91          SUB     C          ; OR A CONSOLE RESTART
3F29      23          INX     H
3F2A      C2323F     JNZ    RST1
3F2D      7E          MOV     A,M
3F2E      90          SUB     B
3F2F      CA403F     JZ     RST3
3F32      RST1:
3F32      23          INX     H
3F33      23          INX     H
3F34      7E          MOV     A,M
3F35      91          SUB     C
3F36      C23F3F     JNZ    RST2
3F39      23          INX     H
3F3A      7E          MOV     A,M
3F3B      90          SUB     B
3F3C      CA403F     JZ     RST3
3F3F      RST2:

```

```

3F3F 03          INX    B
3F40          RST13:
1            +      FETCH  LLOC
3F40 1 210F00  +      LXI    H,LLUC      ; IN THE STACK
3F43 1 39      +      DAD    SP
3F44 73       MOV    M,E
3F45 23       INX    H
3F46 72       MOV    M,D      ; SAVE OLD HL
3F47 23       INX    H
3F48 23       INX    H
3F49 71       MOV    M,C      ; SAVE OLD PC
3F4A 23       INX    H
3F4B 70       MOV    M,B
3F4C C5       PUSH  B
3F4D 0E2A     MVI    C,'*'
3F4F CD553C   CALL  CU
3F52 E1       POP   H      ; RETRIEVE OLD PC FOR DISPLAY
3F53 C0BF3D   CALL  LADR      ; DISPLAY PC
1            +      FETCH  TLUC      ; CLEAN TRAPS
3F56 1 211400 +      LXI    H,TLUC      ; IN THE STACK
3F59 1 39      +      DAD    SP
3F5A 1602     MVI    D,2      ; SET COUNT FOR TWO TRAPS
3F5C          RST14:
3F5C 4E       MOV    C,M      ; GET LSB OF ADDRESS
3F5D 3600     MVI    M,0      ; CLEAR MEMORY
3F5F 23       INX    H
3F60 46       MOV    B,M      ; GET MSB OF ADDRESS
3F61 3600     MVI    M,0
3F63 23       INX    H
3F64 79       MOV    A,C
3F65 60       ORA    B      ; TEST FOR VALID TRAP
3F66 CA6B3F   JZ     RS15      ; ADDRESS = 0, NO TRAP TO RESTORE
3F69 7E       MOV    A,M      ; GET UPCODE BYTE
3F6A 02       STAX  B      ; REPLACE IT
3F6B          RST15:
3F6B 23       INX    H      ; POINT TO NEXT TRAP ADDRESS
3F6C 15       OCR    D
3F6D C25C3F   JNZ   RS14      ; REPEAT FOR TRAP 2
3F70 C36B38   JMP   START

;
; INPUT FROM CONSOLE, ECHOED AND RETURNED IN A
;
3F73          FI:
3F73 CD993C   CALL  C1
3F76 E67F     ANI   /FH
3F78 C5       PUSH  B
3F79 4F       MOV    C,A
3F7A CD553C   CALL  CU
3F7D 79       MOV    A,C
3F7E C1       POP   B
3F7F C9       RET      ; RETURN

```

```

;
; MASTER I/O DEVICE TABLE
; 4 BYTES/ENTRY
;
; BYTE 0 = IDENTIFYING CHARACTER
; BYTE 1 = LOGICAL DEVICE MASK
; BYTES 2,3 = SUBORDINATE PHYSICAL DEVICE TABLE
;
3F80          LTbL:
3F80 43FC      DB      'C',CMSK
3F82 903F      DW      ACT
3F84 52F3      DB      'R',RMSK
3F86 983F      DW      ART
3F88 50CF      DB      'P',PMSK
3F8A A03F      DW      APT
3F8C 4C3F      DB      'L',LMSK
3F8E A83F      DW      ALT
;
; I/O SYSTEM PHYSICAL DEVICE TABLES
; 2 BYTES/ENTRY
;
; BYTE 0 = IDENTIFYING CHARACTER
; BYTE 1 = DEVICE SELECT BIT PATTERN
;
3F90          ACT:
3F90 5400      DB      'T',CTTY      ; CONSOLE = TTY
3F92 4301      DB      'C',CCRT      ; CONSOLE = CRT
3F94 4202      DB      'B',BATCH     ; BATCH MODE CONSOLE = READ,LIST
3F96 3103      DB      '1',CUSE      ; USER DEFINED CONSOLE DEVICE
3F98          ART:
3F98 5400      DB      'T',RTTY      ; READER = TTY
3F9A 5004      DB      'P',RPTR      ; READER = PTR
3F9C 3108      DB      '1',RUSE1     ; USER DEFINED READER DEVICE 1
3F9E 320C      DB      '2',RUSE2     ; USER DEFINED READER DEVICE 2
3FA0          APT:
3FA0 5400      DB      'T',PTTY      ; PUNCH = TTY
3FA2 5010      DB      'P',PPTP      ; PUNCH = PTP
3FA4 3120      DB      '1',PUSE1     ; USER DEFINED PUNCH DEVICE 1
3FA6 3230      DB      '2',PUSE2     ; USER DEFINED PUNCH DEVICE 2
3FA8          ALT:
3FA8 5400      DB      'T',LTTY      ; LIST = TTY
3FAA 4340      DB      'C',LCRT      ; LIST = CRT
3FAC 3180      DB      '1',LUSE1     ; USER DEFINED LIST DEVICE 1
3FAE 32C0      DB      '2',LUSE2     ; USER DEFINED LIST DEVICE 2
;
; EXIT CODE TEMPLATE, TO BE EXECUTED IN RAM
;
;      DB      E      POP D      ; MONITOR WORK STACK ORIGIN
;      DB      D
;      DB      C      POP B
;      DB      B

```

```

; DB FLAGS POP PSW
; DB A
; DB SPL POP H
; DB SPU SPHL
;
3FB0 EXI1: ; MONITOR STACK ORIGIN
3FB0 D1 POP D ; RESTORE D,E
3FB1 C1 POP B ; RESTORE B,C
3FB2 F1 POP PSW ; RESTORE A AND FLAGS
3FB3 E1 POP H ; RESTORE ORIGINAL STACK VALUE
3FB4 F9 SPHL
3FB5 FB EI ; ENABLE INTERRUPTS
3FB6 210000 LXI H,S-S ; RESTORE H,L
3FB7 HLX EQU S-2
3FB9 C30000 JMP S-S ; RETURN TO INTERRUPTED CODE
3FBA PCX EQU S-2
3FBC 0000 TIA: DW 0 ; TRAP 1 ADDRESS
3FBE 00 DW 0 ; TRAP 1 VALUE
3FBF 0000 DW 0 ; TRAP 2 ADDRESS
3FC1 00 DW 0 ; TRAP 2 VALUE
3FC2 ENDX:
;
; DISPLACEMENT OF REGISTER LOCATION FROM SP (LEVEL 0)
;
0005 ALUC EQU 5
0003 BLUC EQU 3
0002 CLUC EQU 2
0001 DLUC EQU 1
0000 ELUC EQU 0
0004 FLUC EQU 4
0010 HLUC EQU HLX-EXIT+9
000F LLUC EQU HLX-EXIT+8
0013 PLUC EQU PCX-EXIT+9
0007 SLUC EQU 7
0014 TLUC EQU TIA-EXIT+8
;
; TABLE FOR ACCESSING REGISTERS
; TABLE CONTAINS:
; (1) REGISTER IDENTIFIER
; (2) STACK POINTER DISPLACEMENT
; (3) PRECISION
;
3FC2 ACIBL:
3FC2 410501 DB 'A', ALUC, 1
3FC5 420301 DB 'B', BLUC, 1
3FC8 430201 DB 'C', CLUC, 1
3FCH 440101 DB 'D', DLUC, 1
3FCE 450001 DB 'E', ELUC, 1
3FD1 460401 DB 'F', FLUC, 1
3FD4 481001 DB 'H', HLUC, 1
3FD7 4C0F01 DB 'L', LLUC, 1

```

```
3FDA 401002      DB      'M',    HLUC,  2
3FDD 501102      DB      'P',    PLUC,  2
3FE0 530702      DB      'S',    SLUC,  2
3FE3 FF          DB      -1
```

```
      ;
      ; END OF PROGRAM
      ;
      END
```

NO PROGRAM ERRORS

## SYMBOL TABLE

\* 01

A	0007	ACI	3F90	ACTBL	3FC2	ALUC	0005
ALT	3FA8	APT	3FA0	ART	3F98	ASU	38CA
AS1	38D9	AS2	38DF	AS3	38E7	AS4	38F1
ASS	38FE	ASSIG	38C2	B	0000	BATCH	0002
BEGIN	3829	BG0	3830	BG1	3840	BLK	3C53
BLOC	0003	BNU	391C	BN1	3948	BNPF	3911
BYTE	3C82	C	0001	CCRT	0001	CI	3C99
C10	3CAA	C11	3CAC	C12	3CB0	CILUC	3700
CLOC	0002	CM0	3961	CM1	3988	CMSK	00FC
CO	3C55	CO0	3C69	CU1	3C7A	COLUC	3703
COMP	395A	CONV	3CC5	CR	0000	CRLF	3CC0
CRTBE	0004	CRIDA	0001	CRT1	0004	CRTIN	3CB1
CRTU	0004	CKTUU	3C6E	CRIS	0005	CSU	3CE4
CS1	3CEB	CS2	3CEF	CS3	3CF2	CSLOC	3718
CSTS	3CD7	CTTY	0000	CUSE	0003	D	0002
DC0	3D06	DC1	3D0E	DC2	3D10	DEBUG	0000
DECUD	3CFC	DELAY	3D25	DG0	3D30	DG1	3D30
DG2	3D45	DG3	3D48	D10	3995	D11	399B
DIGIT	3D2E	DISP	3990	DL0	3D28	DLUC	0001
DLY	0054	DSB	0008	E	0003	ELUC	0000
ENO	3D55	ENB	0000	ENCUD	3D4D	ENDX	3FC2
EOF	39B1	EXO	3D73	EX1	3D76	EX2	3D86
EX3	3D94	EXF	3D9C	EXIT	3FB0	EXPR	3D70
FALSE	0000	FETCH	0F99	F10	39DA	FILL	39D3
FLOC	0004	GO0	39FE	GO1	3A04	GO2	3A1C
GO3	3A26	GOTO	39E4	H	0004	HEXN	3A2E
H1L0	3DA4	HLOC	0010	HLX	3FB7	HXD	3DAE
INIT	0000	I0BYT	0003	I0CHK	3DB4	I0SET	3DB8
L	0005	L1LUC	3712	L2LUC	3715	LADR	3DBF
L8YTE	3DC7	LCRT	0040	LDLY	0014	LEU	3DD9
LEAD	3DD7	LER	3C43	LF	000A	LLUC	000F
LMSK	003F	LU	3DE3	LU0	3A53	LOAD	3A4B
LTBL	3FB0	LTTY	0000	LUSE1	0080	LUSE2	00C0
LVER	000B	M	0006	MEM0	3E05	MEMCK	3DF8
MEMS1	3E01	MUVE	3A5F	MV0	3A66	NIU	3E24
NIBBL	3E16	NULL	3A72	P1LUC	370C	P2C	3E4E
P2LDC	370F	PAD	0002	PADR	3E28	PBITA	0080
P8YTE	3E30	PCHK	3E4B	PCMD	0002	PCX	3FBA
PDI	0002	PDU	0003	PEUL	3E5B	PGU	3E74
PGRM	3E65	PLUC	0013	PMSK	00CF	PU	3E7D
PO0	3E8A	PU1	3E9D	PPIP	0010	PRU	3A82
PR1	3A95	PR2	3AB4	PR3	3ABA	PR4	3AC1
PRDY	0040	PROG	3A78	PRUG0	0080	PROMC	0001
PRONO	0000	PSW	0006	PTPC	0001	PTPG0	000A
P1PNO	0008	PTPU	0003	PTPS	0001	PTNC	0001
PTRDA	0020	PTRGU	000C	PTR1	0003	PTRNO	0008
PTRS	0001	PTTY	0000	PUSE1	0020	PUSE2	0030
R1LUC	3706	R2LUC	3709	RBIT	0001	RCMD	0004
READ	3ACB	RE00	3ACF	RED1	3AEF	RED2	3B01
RED3	3B0E	RESTA	3F08	R1	3EA5	R10	3EB9
R11	3EC7	R12	3ECB	R13	3ED1	R14	3EE0
R15	3EF1	R16	3EF6	R1X	3EFF	RMSK	00F3



RPTR	0004	RS1	0008	RST0	3F17	RST1	3F32
RST2	3F3F	RST3	3F40	RST4	3F5C	RST5	3F6B
RITY	0000	RUSE1	0008	RUSE2	000C	SLOC	0007
SP	0006	START	386B	SUU	3B1D	SU1	3B3C
SUBS	3B12	T1A	3F8C	TBL	3892	TI	3F73
TLOC	0014	TOUT	00FA	TR0	3B47	TRAN	3B40
TRUE	FFFF	TIC	0001	TTI	0000	TTU	0000
TTS	0001	TTYBE	0004	TTYDA	0001	TTYGU	0009
TYIN	3CA1	TYND	0008	TYOU	3C5D	VER	001E
VER0	3862	VERS	381E	WR10	3B66	WRI1	3B7B
WR12	3B7E	WR13	3B99	WRITE	3B5E	X	3B8B
X0	3B86	X1	3BC7	X2	3BCA	X3	3BE2
X4	3C04	X5	3C06	X6	3C14	X7	3C17

\* 02

\* 03

\* 04

\* 05

\* 06

\* 07

\* 08