

```

;           INTELLEC 8/MOD 8 MONITOR
;
;           FAST PRUM PROGRAMMING ALGORITHM ADDED, 21 JUN 1974
;
;           COPYRIGHT (C) 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>
;
; <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>
;
; <LOGICAL DEVICE> ::= CONSOLE!READER!LIST!PUNCH
;
; <PHYSICAL DEVICE> ::= CRT!TTY!PTR!PTP!BATCH!1!2
;
; <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><.>
;
0 E VER EQU 30 ; VERSION 3.0
    TITLE ' INTELLEC 8/MOD 8 MONITOR, VERSION 3.0, 14 APRIL 1975'
;
; CONDITIONAL ASSEMBLY SWITCHES
;
0000 FALSE EQU 0
FFFF TRUE EQU NOT FALSE
0000 DEBUG EQU FALSE ; DEBUG MODE -
; MODIFY CERTAIN CODE SECTIONS
; SO THAT THE 8008 VERSION 2.0
; MONITOR MAY BE ASSEMBLED BY
; MAC80 AND DEBUGGED BY THE 8080
; VERSION 1.0 MONITOR

1 IF DEBUG
1 BIAS EQU 0
    ENDIF
;
1 IF NOT DEBUG
00C 1 BIAS EQU 8 ; ASSIGN PROPER NUMERIC VALUES TO
1 ; OUTPUT PORTS FOR THE 8008
    ENDIF
;
; 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      PTRDA      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      CRTDA      IF CRTDA = 0, INPUT IS READY
;      1      1      OVERRUN ERROR
;      2      0      CRTBE      IF CRTBE = 0, OUTPUT IS READY
;      3      1      FRAMING ERROR
;      4      1      PARITY ERROR
;      5      UNASSIGNED
;      6      UNASSIGNED
;      7      UNASSIGNED
;
; I/O COMMAND CONSTANTS
;
0001      RBIT EQU      1
0002      PCMD EQU      2
0004      RCMD EQU      4
0008      DSB EQU      8
0080      PBITA EQU     80H
;
; TTY I/O CONSTANTS
;
0000      TTI EQU      0      ; TTY INPUT DATA PORT
0008      TTO EQU     BIAS+0   ; TTY OUTPUT DATA PORT
0001      TTS EQU      1      ; TTY INPUT STATUS PORT
0009      TTC EQU     BIAS+1   ; TTY OUTPUT COMMAND PORT
0009      TTYGO EQU    RBIT OR DSB ; START TTY READER
0008      TTYNO EQU    DSB     ; STOP TTY READER
0001      TTYDA EQU    1      ; DATA AVAILABLE
0004      TTYBE 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
000C CRTD EQU BIAS+4 ; CRT OUTPUT DATA PORT
0001 CRTDA EQU 1 ; DATA AVAILABLE
0004 CRTBE EQU 4 ; TRANSMIT BUFFER EMPTY
;
; PTR I/O CONSTANTS
;
0003 PTRI EQU 3 ; PTR INPUT DATA PORT (NOT INVERTED)
0001 PTRS EQU TTS ; PTR INPUT STATUS PORT
0009 PTRC EQU TTC ; PTR OUTPUT COMMAND PORT
000C PTRGO EQU RCMD OR DSB ; START PTR
0008 PTRNO EQU TTYNO ; STOP PTR
0020 PTRDA EQU 20H ; PTR DATA AVAILABLE
;
; PTP I/O CONSTANTS
;
000B PTPO EQU BIAS+3 ; PTP OUTPUT DATA PORT
0001 PTPS EQU TTS ; PTP INPUT STATUS PORT
0009 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
;
000A PAD EQU BIAS+2 ; PROM ADDRESS OUTPUT PORT
000B PDU EQU PTPO ; PROM DATA OUTPUT PORT
0002 PDI EQU 2 ; PROM DATA INPUT PORT
0009 PROMC EQU TTC ; PROGRAMMING PULSE OUTPUT PORT
0080 PRGO EQU PBITA ; START PROGRAMMING
0000 PRNO EQU 0 ; STOP PROGRAMMING
-0000 ENB EQU 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 ; 1 MS. DELAY FOR 8080 DEBUG
ENDIF
1 IF NOT DEBUG
0017 1 DLY EQU 23 ; COUNTER FOR 1 MS. DELAY
ENDIF
000D CR EQU 0DH ; ASCII VALUE OF CARRIAGE RETURN
000A LF EQU 0AH ; ASCII VALUE OF LINE FEED
;
; MACRO DEFINITIONS
;

```

```

FFFF      FIRST SET      TRUE
          MODID MACRO    TABLE, MASK
          1             LXI   D, TABLE      ; ADDRESS OF PHYSICAL UNIT TABLE
          1             MVI   C, MASK       ; C = SELECT BIT MASK
          2             IF     FIRST        ; EMIT THIS CODE ONCE,
          2                                     ; BRANCH TO IT THEREAFTER
          2
          IOMOD::
          2             FIRST SET  FALSE
          2             CALL    NOISE      ; SCAN INPUT AND ECHO UNTIL
          2                                     ; PHYSICAL DEVICE CHAR IS ENCOUNTERED
          2                                     ; SET TABLE LENGTH
          2             MVI     B, 4
          2             MOV     H, D
          2             MOV     L, E
          2             CALL    TEST       ; COMPARE PHYSICAL DEVICE AGAINST
          2             CALL    INCHL
          2             MOV     E, M       ; TABLE, RETURN HL -> BIT PATTERN
          2
          SCANOUT:
          2             CALL    TI
          2             CPI     CR
          2             JNZ    SCANOUT     ; SCAN PAST CR
          2             LXI     H, IOBYT   ; GET I/O STATUS
          2             MOV     A, M
          2             ANA     C         ; CLEAR FIELD
          2             ORA     E         ; SET NEW STATUS
          2             MOV     M, A      ; RETURN TO MEMORY
          2             JMP     START
          2
          TEST:
          2             CMP     M
          2             RZ
          2             CALL    INCHL
          2             CALL    INCHL
          2             DCR     B
          2             JNZ    TEST       ; CONTINUE LOOKUP
          2             JMP     LEK       ; ERROR RETURN
          1             ENDIF
          2             IF     NOT FIRST
          2             JMP     IOMOD
          1             ENDIF
          1             ENDM
          ;
          ; 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      CTTY EQU      00000000B      ; CONSOLE I/O = TTY
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
0000      RTTY EQU      00000000B      ; READER = TTY
0004      RPTR EQU      00000100B      ; READER = PTR
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      CILOC EQU     3700H          ; USER CONSOLE INPUT
3703      CULOC 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
;
; POINTERS TO RAM
;
1          IF          NOT DEBUG
0003 1      ORG          3              ; LOCATION OF RAM STORAGE AREA
          ENDIF
;
1          IF          DEBUG
1          ORG          100H
          ENDIF
;
0003      IOBYT:
0001          DS          1              ; I/O STATUS BYTE
0004      HPRIME:
0002          DS          2              ; TEMP FOR HL
0006      CBRCH:
0002          DS          2              ; CASE BRANCH LOCATION
0008      ADR1:
0002          DS          2              ; FIRST PARAMETER
000A      ADR2:
0002          DS          2              ; SECOND PARAMETER
000C      ADR3:
0002          DS          2              ; THIRD PARAMETER
;
1          IF          DEBUG
1          ORG          800H            ; LOCATE IN RAM FOR DEBUG

```

```

                ENDIF
;
1              IF      NOT DEBUG
3800 1         ORG      3800H          ; LOCATE IN TOP 8 ROMS IN 16K
                ENDIF
;
; BRANCH TABLE FOR I/O SYSTEM
;
3800 442938    JMP      BEGIN          ; RESET ENTRY POINT
3803 44963C    JMP      CI             ; CONSOLE INPUT
3806 44DB3E    JMP      RI             ; READER INPUT
3809 44523C    JMP      CO             ; CONSOLE OUTPUT
380C 44853E    JMP      PO             ; PUNCH OUTPUT
380F 442C3E    JMP      LO             ; LIST OUTPUT
3812 44D63C    JMP      CSTS           ; CONSOLE INPUT STATUS
3815 44F43D    JMP      IOCHK          ; I/O SYSTEM STATUS
3818 44FA3D    JMP      IOSET          ; SET I/O CONFIGURATION
381B 44433E    JMP      MEMCK          ; COMPUTE SIZE OF MEMORY
;
; INITIAL CONDITIONS FOR I/O SYSTEM
;
0000          INIT EQU      0          ; INITIALLY,
; CONSOLE = TTY,
; READER = TTY,
; PUNCH = TTY,
; LIST = TTY
;
381E 0D0A3830  VERS: DB      CR,LF,'8008 V'
3822 30382056
;
1              IF      NOT DEBUG
3826 1 332E30    DB      VER/10+'0','.',VER MOD 10+'0'
                ENDIF
;
1              IF      DEBUG
1              DB      'X.X'
                ENDIF
000B          LVER EQU      S-VERS
;
; PROGRAM ENTRY POINT
;
; LOCATE THE STACK IN THE TOP OF AVAILABLE RAM
;
3829          BEGIN:
3829 36032E00    LXI      H,IOBYT        ; POINT HL AT IOBYT
382D 3E00        MVI      M,INIT        ; INITIAL VALUE OF I/O
;
; TYPE SIGN-ON
;
382F 361E2E38    LXI      H,VERS          ; ADDRESS OF MESSAGE
3833 160B        MVI      C,LVER        ; LENGTH OF MESSAGE
3835          VERO:

```

```

3835 CF          MOV      B,M
3836 46F03D      CALL     INCHL
3839 DD          MOV      D,H
383A E6          MOV      E,L
383B 46523C      CALL     CO
383E EB          MOV      H,D
383F F4          MOV      L,E
3840 11          DCR      C
3841 483538      JNZ      VERU
;
; MAIN COMMAND LOOP.
;
; THIS LOOP IS THE STARTING POINT OF ALL COMMAND SEQUENCES.
; IN THIS CODE 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.
;
3844          START:
3844 0608          MVI      A,TTYNO          ; RESET TTY, PTR, PTP
3846 53           OUT      TTC              ; RESET READER
3847 46CC3C      CALL     CRLF             ; TYPE <CR>, <LF>
384A 0E2E          MVI      B, '.'
384C 46523C      CALL     CO              ; OUTPUT A PERIOD
384F 46493F      CALL     TI              ; GET A CHARACTER
3852 1441          SUI      'A'           ; TEST FOR A-W
3854 704438      JM       START          ; IF A, ERROR
3857 3C17          CPI      'W'-'A'+1
3859 50483C      JP       LER            ; GT W, ERROR
385C 80           ADD      A              ; *2
385D 36752E38    LXI      H,TBL          ; ADDRESS OF TABLE
3861 86           ADD      L              ; ADD BIAS
3862 F0           MOV      L,A
3863 E7           MOV      M,H          ; GET ADDRESS IN D,E
3864 30           INR      L
3865 DF          MOV      D,M
3866          GO:
3866 36062E00      LXI      H,CBRCH          ; ADDRESS OF JMP INSTRUCTION
386A 3E44          MVI      M,(JMP 0)          ; STUFF OPCODE
386C 30           INR      L
386D FC          MOV      M,E          ; STUFF 8 LSB
386E 30           INR      L
386F FB          MOV      M,D          ; STUFF 8 MSB
3870 1602          MVI      C,2          ; SET UP C FOR TWO PARAMETER COMMANDS
3872 440600      JMP      CBRCH          ; GO DO IT
;
; 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.
;

```

```

3875          TBL:
3875  A338      DW      ASSIGN      ; A - ASSIGN I/O UNITS
3877  0E39      DW      BNPFF       ; B - PUNCH BNPFF
3879  6739      DW      COMP        ; C - COMPARE PROM WITH MEMORY
387B  AC39      DW      DISP        ; D - DISPLAY RAM MEMORY
387D  D739      DW      EOF         ; E - ENDFILE A HEXADECIMAL FILE
387F  033A      DW      FILL        ; F - FILL MEMORY
3881  163A      DW      GOTO        ; G - GO TO MEMORY ADDRESS
3883  1D3A      DW      HEXN        ; H - HEXADECIMAL SUM AND DIFFERENCE
3885  483C      DW      LER         ; I -
3887  483C      DW      LER         ; J -
3889  483C      DW      LER         ; K -
388B  483A      DW      LOAD        ; L - LOAD BNPFF TAPE
388D  623A      DW      MOVE        ; M - MOVE MEMORY
388F  8E3A      DW      NULL        ; N - PUNCH NULLS FOR LEADER
3891  483C      DW      LER         ; O -
3893  943A      DW      PROG        ; P - PROGRAM A 1702A PROM
3895  483C      DW      LER         ; Q -
3897  F83A      DW      READ        ; R - READ HEXADECIMAL FILE
3899  6A3B      DW      SUBS        ; S - SUBSTITUTE MEMORY
389B  B93B      DW      TRAN        ; T - TRANSFER A PROM TO MEMORY
389D  483C      DW      LER         ; U -
389F  483C      DW      LER         ; V -
38A1  DA3B      DW      WRITE       ; W - WRITE HEX TAPE

```

```

;
; PROCESS I/O DEVICE ASSIGNMENT COMMANDS.
;

```

```

; THIS ROUTINE MAPS SYMBOLIC DEVICE IDENTIFIERS TO BITS
; IN THE I/O STATUS BYTE (IOBYT) TO ALLOW FOR CONSOLE
; MODIFICATION OF SYSTEM I/O CONFIGURATION.
;

```

```

38A3          ASSIGN:
38A3  46493F    CALL    TI           ; GET LOGICAL DEVICE CHARACTER
38A6  3C43      CPI     'C'         ; CONSOLE?
38A8  48E438    JNZ    AS0          ; TEST FOR READER
1          MODIO  ICT,CMSK        ; MODIFY CONSOLE DEVICE
38AB  1 26541E3F+ LXI    D,ICT       ; ADDRESS OF PHYSICAL UNIT TABLE
38AF  1 16FC    +     MVI    C,CMSK   ; C = SELECT BIT MASK
2          +     IF     FIRST       ; EMIT THIS CODE ONCE,
2          +     ; BRANCH TO IT THEREAFTER
38B1  2        +IOMOD:
0000  2        +FIRST SET    FALSE
38B1  2 466A3E +     CALL    NOISE   ; SCAN INPUT AND ECHO UNTIL
2          +     ; PHYSICAL DEVICE CHAR IS ENCOUNTERED
38B4  2 0E04    +     MVI    B,4     ; SET TABLE LENGTH
38B6  2 EB      +     MOV     H,D
38B7  2 F4      +     MOV     L,E
38B8  2 46D238 +     CALL    TEST     ; COMPARE PHYSICAL DEVICE AGAINST

```

```

38BB 2 46F03D + CALL INCHL
38BE 2 E7 + MOV E,M ; TABLE, RETURN HL -> BIT PATTERN
38BF 2 +SCANOUT:
38BF 2 46493F + CALL TI
38C2 2 3C0D + CPI CR
38C4 2 48BF38 + JNZ SCANOUT ; SCAN PAST CR
38C7 2 36032E00+ LXI H,IOBYT ; GET I/O STATUS
38CB 2 C7 + MOV A,M
38CC 2 A2 + ANA C ; CLEAR FIELD
38CD 2 B4 + ORA E ; SET NEW STATUS
38CE 2 F8 + MOV M,A ; RETURN TO MEMORY
38CF 2 444438 + JMP START
38D2 2 +TEST: ; INDEX THROUGH PHYSICAL UNIT TABLE
38D2 2 BF + CMP M ; COMPARE DEVICE CHAR WITH LEGAL VALUES
38D3 2 2B + RZ ; RETURN WITH HL -> DEVICE SELECT BITS
38D4 2 46F03D + CALL INCHL
38D7 2 46F03D + CALL INCHL
38DA 2 09 + DCR B
38DB 2 48D238 + JNZ TEST ; CONTINUE LOOKUP
38DE 2 44483C + JMP LER ; ERROR RETURN
1 + ENDIF
2 + IF NOT FIRST
38E1 2 448138 + JMP IOMOD
1 + ENDIF
38E4 AS0:
38E4 3C52 CPI 'R' ; READER?
38E6 48F238 JNZ AS1 ; TEST FOR PUNCH
1 + MODIO IRT,RMSK ; MODIFY READER DEVICE
38E9 1 265C1E3F+ LXI D,IRT ; ADDRESS OF PHYSICAL UNIT TABLE
38ED 1 16F3 + MVI C,RMSK ; C = SELECT BIT MASK
2 + IF FIRST ; EMIT THIS CODE ONCE,
2 + ; BRANCH TO IT THEREAFTER
2 +IOMOD::
2 +FIRST SET FALSE ; SCAN INPUT AND ECHO UNTIL
2 + CALL NOISE ; PHYSICAL DEVICE CHAR IS ENCOUNTERED
2 + MVI B,4 ; SET TABLE LENGTH
2 + MOV H,D
2 + MOV L,E
2 + CALL TEST ; COMPARE PHYSICAL DEVICE AGAINST
2 + CALL INCHL ; TABLE, RETURN HL -> BIT PATTERN
2 + MOV E,M
2 +SCANOUT:
2 + CALL TI
2 + CPI CR
2 + JNZ SCANOUT ; SCAN PAST CR
2 + LXI H,IOBYT ; GET I/O STATUS
2 + MOV A,M
2 + ANA C ; CLEAR FIELD
2 + ORA E ; SET NEW STATUS
2 + MOV M,A ; RETURN TO MEMORY

```

```

2          +      JMP      START
2          +TEST:  ; INDEX THROUGH PHYSICAL UNIT TABLE
2          +      CMP      M          ; COMPARE DEVICE CHAR WITH LEGAL VALUES
2          +      RZ          ; RETURN WITH HL -> DEVICE SELECT BITS
2          +      CALL     INCHL
2          +      CALL     INCHL
2          +      DCR      B
2          +      JNZ     TEST        ; CONTINUE LOOKUP
2          +      JMP     LER         ; ERROR RETURN
1          +      ENDIF
2          +      IF      NOT FIRST
38EF 2 44B138 +      JMP     IOMOD
1          +      ENDIF
38F2          AS1:
38F2 3C50      CPI      'P'          ; PUNCH?
38F4 480039    JNZ     AS2          ; TEST FOR LIST
1          +      MODIO   OPT,PMSK   ; MODIFY PUNCH DEVICE
38F7 1 26641E3F+ LXI     D,OPT        ; ADDRESS OF PHYSICAL UNIT TABLE
38FB 1 16CF    MVI     C,PMSK   ; C = SELECT BIT MASK
2          +      IF      FIRST     ; EMIT THIS CODE ONCE,
2          +          ; BRANCH TO IT THEREAFTER
2          +IOMOD::
2          +FIRST SET FALSE
2          +      CALL     NOISE     ; SCAN INPUT AND ECHO UNTIL
2          +          ; PHYSICAL DEVICE CHAR IS ENCOUNTERED
2          +      MVI     B,4        ; SET TABLE LENGTH
2          +      MOV     H,D
2          +      MOV     L,E
2          +      CALL     TEST      ; COMPARE PHYSICAL DEVICE AGAINST
2          +      CALL     INCHL
2          +      MOV     E,M        ; TABLE, RETURN HL -> BIT PATTERN
2          +SCANOUT:
2          +      CALL     TI
2          +      CPI     CR
2          +      JNZ     SCANOUT    ; SCAN PAST CR
2          +      LXI     H,IOBYT   ; GET I/O STATUS
2          +      MOV     A,M
2          +      ANA     C          ; CLEAR FIELD
2          +      ORA     E          ; SET NEW STATUS
2          +      MOV     M,A       ; RETURN TO MEMORY
2          +      JMP     START
2          +TEST:  ; INDEX THROUGH PHYSICAL UNIT TABLE
2          +      CMP      M          ; COMPARE DEVICE CHAR WITH LEGAL VALUES
2          +      RZ          ; RETURN WITH HL -> DEVICE SELECT BITS
2          +      CALL     INCHL
2          +      CALL     INCHL
2          +      DCR      B
2          +      JNZ     TEST        ; CONTINUE LOOKUP
2          +      JMP     LER         ; ERROR RETURN
1          +      ENDIF
2          +      IF      NOT FIRST

```

```

38FD 2 44B138 +      JMP      IOMOD
1      +      ENDIF
3900      AS2:
3900 3C4C      CPI      'L'      ; LIST?
3902 48483C    JNZ      LER      ; ERROR
1      +      MODIO    OLT,LMSK  ; MODIFY LIST DEVICE
3905 1 266C1E3F+ LXI      D,OLT    ; ADDRESS OF PHYSICAL UNIT TABLE
3909 1 163F     +      MVI      C,LMSK  ; C = SELECT BIT MASK
2      +      IF      FIRST   ; EMIT THIS CODE ONCE,
2      +      ; BRANCH TO IT THEREAFTER
2      +IOMOD::
2      +FIRST SET FALSE
2      +      CALL    NOISE    ; SCAN INPUT AND ECHO UNTIL
2      +      ; PHYSICAL DEVICE CHAR IS ENCOUNTERED
2      +      MVI      B,4     ; SET TABLE LENGTH
2      +      MOV      H,D
2      +      MOV      L,E
2      +      CALL    TEST    ; COMPARE PHYSICAL DEVICE AGAINST
2      +      CALL    INCHL
2      +      MOV      E,M     ; TABLE, RETURN HL -> BIT PATTERN
2      +SCANOUT:
2      +      CALL    TI
2      +      CPI      CR
2      +      JNZ     SCANOUT  ; SCAN PAST CR
2      +      LXI     H,I0BYT  ; GET I/O STATUS
2      +      MOV      A,M
2      +      ANA     C        ; CLEAR FIELD
2      +      ORA     E        ; SET NEW STATUS
2      +      MOV      M,A     ; RETURN TO MEMORY
2      +      JMP     START
2      +TEST:
2      +      CMP     M        ; INDEX THROUGH PHYSICAL UNIT TABLE
2      +      RZ
2      +      CALL    INCHL  ; COMPARE DEVICE CHAR WITH LEGAL VALUES
2      +      CALL    INCHL  ; RETURN WITH HL -> DEVICE SELECT BITS
2      +      DCR     B
2      +      JNZ     TEST    ; CONTINUE LOOKUP
2      +      JMP     LER     ; ERROR RETURN
1      +      ENDIF
2      +      IF     NOT FIRST
390B 2 44B138 +      JMP     IOMOD
1      +      ENDIF
;
; 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.
;
BNPF:
390E          46853D      CALL    EXPR          ; GET TWO ADDRESSES
3911          46CC3C      CALL    CRLF
3914          46203E      CALL    LEAD
3917
BN0:
3917          46953E      CALL    PEOL          ; PUNCH CR,LF
391A          46D43D      CALL    GETAD         ; GET HL AND DE
391D          1620        MVI    C,' '         ; ZERO SUPPRESSION CHARACTER
391F          26101E27    LXI    D,10000       ; PUNCH ADDRESS IN DECIMAL
3923          462C3D      CALL    DIGIT
3926          26E81E03    LXI    D,1000
392A          462C3D      CALL    DIGIT
392D          26641E00    LXI    D,100
3931          462C3D      CALL    DIGIT
3934          260A1E00    LXI    D,10
3938          462C3D      CALL    DIGIT
393B          26011E00    LXI    D,1
393F          1630        MVI    C,'0'         ; FORCE AT LEAST 1 ZERO
3941          462C3D      CALL    DIGIT
3944          0E20        MVI    B,' '
3946          46B53E      CALL    PO
3949          46D43D      CALL    GETAD         ; GET HL AND DE
394C
BN1:
394C          46553D      CALL    ENCODE        ; ENCODE A MEMORY BYTE INTO BNPF
394F          46D43D      CALL    GETAD         ; GET HL AND DE
3952          46E23D      CALL    HILO
3955          608E3A      JC     NULL          ; ALL DONE, PUNCH TRAILER AND RETURN
3958          46343F      CALL    SAVIT
395B          46D43D      CALL    GETAD         ; GET HL AND DE
395E          C6         MOV    A,L
395F          2403        ANI    03H          ; PUNCH CR,LF, ADDRESS ON MULTIPLE OF 4
3961          484C39      JNZ    BN1
3964          441739      JMP    BN0

;
; 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>
;
COMP:
3967          11         DCR    C
3968          46853D      CALL    EXPR          ; GET ONE ADDRESS
396B          46CC3C      CALL    CRLF          ; OUTPUT CR AND LF

```

```

396E 46D43D      CALL  GETAD      ; GET HL AND DE
3971 1600        MVI   C,0        ; COUNT/PROM ADDRESS
3973 0600        MVI   A,ENB     ; ENABLE PROM PROGRAMMER
3975 53          OUT   PROMC
3976           CM0:
3976 C2          MOV   A,C          ; SET PROM ADDRESS
3977 2CFF        XRI   OFFH       ; INVERT ADDRESS
3979 55          OUT   PAD
397A 46253D     CALL  DELAY      ; WAIT FOR 6-76 BOARD TO LATCH DATA
397D 45          IN    PDI        ; GET PROM DATA
397E 2CFF        XRI   OFFH
3980 BF         CMP   M          ; COMPARE WITH MEMORY
3981 68A239     JZ    CM1        ; COMPARE
3984 E2          MOV   E,C
3985 46343F     CALL  SAVIT
3988 46D43D     CALL  GETAD
398B 46003E     CALL  LADR      ; PRINT MEMORY ADDRESS
398E C7          MOV   A,M
398F 46103E     CALL  LBYTE     ; PRINT RAM DATA
3992 46503C     CALL  BLK
3995 45          IN    PDI        ; GET PROM DATA
3996 2CFF        XRI   OFFH
3998 46103E     CALL  LBYTE     ; PRINT PROM DATA
399B 46CC3C     CALL  CRLF
399E 46D43D     CALL  GETAD
39A1 D4          MOV   C,E
39A2           CM1:
39A2 46F03D     CALL  INCHL
39A5 10          INR   C          ; ADJUST PROM ADDRESS
39A6 487639     JNZ  CMO
39A9 444438     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.
;
DISP:
39AC 46853D     CALL  EXPR      ; GET TWO ADDRESSES
39AF           D10:
39AF 46CC3C     CALL  CRLF
39B2 46D43D     CALL  GETAD    ; GET HL AND DE
39B5 46003E     CALL  LADR    ; PRINT MEMORY ADDRESS
39B8           D11:
39B8 C7          MOV   A,M

```

```

39B9 46103E      CALL  LBYTE      ; PRINT DATA
39BC 46503C      CALL  BLK        ; PRINT SPACE
39BF 46D43D      CALL  GETAD      ; GET HL AND DE
39C2 46E23D      CALL  HILO       ; TEST FOR COMPLETION
39C5 604438      JC    START
39C8 46J43F      CALL  SAVIT      ; STORE HL,DE
39CB 46D43D      CALL  GETAD      ; GET HL AND DE
39CE C6             MOV   A,L
39CF 240F        ANI   0FH        ; PRINT CR,LF,ADDRESS ON MULTIPLE OF 16
39D1 48B839      JNZ  D11
39D4 44AF39      JMP  D10

```

```

;
; 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.
;

```

```

39D7      EOF:
39D7 11          DCR   C          ; GET ONE PARAMETER
39D8 46853D      CALL  EXPR
39DB 46953E      CALL  PEUL      ; PUNCH CR,LF
39DE 0E3A        MVI   B,':'
39E0 46B53E      CALL  PU
39E3 A8           XRA   A          ; CLEAR CHECKSUM
39E4 D8           MOV   D,A
39E5 467B3E      CALL  PBYTE     ; OUTPUT RECORD LENGTH
39E8 C3           MOV   A,D
39E9 46D43D      CALL  GETAD
39EC D8           MOV   D,A
39ED E6           MOV   E,L
39EE C5           MOV   A,H
39EF 467B3E      CALL  PBYTE
39F2 C4           MOV   A,E
39F3 467B3E      CALL  PBYTE
39F6 0601        MVI   A,1        ; RECORD TYPE 1
39F8 467B3E      CALL  PBYTE
39FB A8           XRA   A
39FC 93         SUB   D          ; OUTPUT CHECKSUM
39FD 467B3E      CALL  PBYTE
3A00 448E3A      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.
;
3A03          ; FILL:
3A03 10      INR    C          ; GET 3 PARAMETERS
3A04 46853D  CALL   EXPR
3A07 C4      MOV    A,E        ; GET DATA IN A
3A08 46D43D  CALL   GETAD      ; GET HL AND DE
3A0B C8      MOV    B,A
3A0C          ; F10:
3A0C F9      MOV    M,B        ; STORE CONSTANT IN MEMORY
3A0D 46E23D  CALL   HILO      ; TEST FOR COMPLETION
3A10 40UC3A  JNC    F10        ; CONTINUE LOOPING
3A13 444438  JMP    START

;
; GO TO <ADDRESS>
;
; THE G COMMAND IS USED FOR TRANSFERRING CONTROL FROM THE
; MONITOR TO A USER PROGRAM.
;
3A16          ; GOTO:
3A16 11      DCR    C
3A17 46853D  CALL   EXPR
3A1A 446638  JMP    GO

;
; 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>
;
3A1D          ; HEXN:
3A1D 46853D  CALL   EXPR          ; GET TWO NUMBERS
3A20 46CC3C  CALL   CRLF
3A23 46D43D  CALL   GETAD      ; GET HL AND DE
3A26 C6      MOV    A,L        ; COMPUTE HL + DE
3A27 84      ADD    E
3A28 E0      MOV    E,A        ; SAVE LSB IN E
3A29 C5      MOV    A,H
3A2A 8B      ADC    D
3A2B 46103E  CALL   LBYTE        ; DISPLAY MSB OF SUM
3A2E C4      MOV    A,E        ; RETRIEVE LSB
3A2F 46103E  CALL   LBYTE        ; DISPLAY IT
3A32 46503C  CALL   BLK          ; TYPE A SPACE
3A35 46D43D  CALL   GETAD      ; GET HL AND DE
3A38 C6      MOV    A,L        ; COMPUTE HL-DE
3A39 94      SUB    E
3A3A E0      MOV    E,A        ; SAVE LSB OF DIFFERENCE
3A3B C5      MOV    A,H
3A3C 9B      SBB    D

```



```

3A3D  E8          MOV     H,A
3A3E  46103E       CALL    LBYTE      ; DISPLAY MSB OF DIFFERENCE
3A41  C4          MOV     A,E      ; RETRIEVE LSB
3A42  46103E       CALL    LBYTE      ; DISPLAY LSB OF DIFFERENCE
3A45  444438       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 RIX), BUT ALL DATA READ BEFORE THE END
; OF TAPE WAS ENCOUNTERED WILL BE LOADED.
;
3A48          LOAD:
3A48  46853D       CALL    EXPR      ; GET TWO ADDRESSES
3A4B  46CC3C       CALL    CRLF
3A4E          LOO:
3A4E  46FC3C       CALL    DECODE    ; CONVERT BNPf, RETURN IN C-REGISTER
3A51  C2          MOV     A,C      ; CONVERTED DATA BYTE
3A52  46D43D       CALL    GETAD     ; GET HL AND DE
3A55  F8          MOV     M,A      ; STORE DATA
3A56  46E23D       CALL    HILO     ; TEST FOR COMPLETION
3A59  604438       JC     START
3A5C  46343F       CALL    SAVIT
3A5F  444E3A       JMP     LOO

;
; MOVE A BLOCK OF RAM MEMORY
;
3A62          MOVE:
3A62  10          INR     C      ; GET THREE ADDRESSES
3A63  46853D       CALL    EXPR
3A66          MVO:
3A66  46D43D       CALL    GETAD     ; GET DATA
3A69  C7          MOV     A,M      ; GET DATA
3A6A  360C2E00    LXI    H,ADR3    ; GET DESTINATION ADDRESS
3A6E  CF          MOV     B,M
3A6F  30          INR     L
3A70  F7          MOV     L,M
3A71  E9          MOV     H,B
3A72  F8          MOV     M,A      ; STORE DATA
3A73  46F03D       CALL    INCHL    ; INCREMENT DESTINATION ADDRESS
3A76  CD          MOV     B,H
3A77  D6          MOV     C,L
3A78  360C2E00    LXI    H,ADR3
3A7C  F9          MOV     M,B
3A7D  30          INR     L

```

3A7E FA  
 3A7F 46D43D  
 3A82 46E23D  
 3A85 604438  
 3A88 46343F  
 3A8B 44663A

MOV M,C  
 CALL GETAD  
 CALL HILO  
 JC START  
 CALL SAVIT  
 JMP MVO

;  
 ; 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.  
 ;

3A8E  
 3A8E 46203E  
 3A91 444438

NULL:  
 CALL LEAD  
 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.  
 ;

3A94  
 3A94 10  
 3A95 46853D  
 3A98 31  
 3A99 D7  
 3A9A 31  
 3A9B 31  
 3A9C E7  
 3A9D 31  
 3A9E DF  
 3A9F 31  
 3AA0 CF  
 3AA1 31  
 3AA2 EF  
 3AA3 1  
 3AA4 C4  
 3AA5 96  
 3AA6 E0  
 3AA7 20  
 3AA8 C3

PROG:  
 INR C  
 CALL EXPR ; HL = TOP AFTER RETURN  
 DCR L  
 MOV C,M ; PROM ADDRESS  
 DCR L  
 DCR L ; HL = TOP-3  
 MOV E,M ; LSB OF HIGH ADDRESS  
 DCR L  
 MOV D,M ; MSB OF HIGH ADDRESS  
 DCR L  
 MOV B,M ; LSB OF LOW ADDRESS  
 DCR L  
 MOV H,M ; MSB OF LOW ADDRESS  
 MOV L,B  
 MOV A,E  
 SUB L  
 MOV E,A ; COUNT  
 INR E ; ADJUST SO 256 = 0  
 MOV A,D

```

3AA9 9D          SBB    H
3AAA 60483C      JC     LER          ; CARRY = ERROR
3AAD          PR0:
3AAD 0600      MVI    A,ENB
3AAF 53        OUT    PROMC      ; ENABLE PROM PROGRAMMER
3AB0 C2        MOV    A,C
3AB1 2CFF      XRI    0FFH
3AB3 55        OUT    PAD          ; PROM ADDRESS
3AB4 45        IN     PDI          ; READ VALUE
3AB5 2CFF      XRI    0FFH
3AB7 BF        CMP    M          ; COMPARE WITH DESIRED
3AB8 68EA3A    JZ     PR4          ; DON'T HAVE TO PROGRAM THE LOCATION
3ABB 1EF0      MVI    D,-16       ; SET MAX TRIES = 16
3ABD          PR1:
3ABD 469F3E    CALL   PGRM        ; PULSE AND DELAY 20 MS.
3AC0 45        IN     PDI          ; READ VALUE
3AC1 2CFF      XRI    0FFH
3AC3 BF        CMP    M          ; COMPARE WITH DESIRED
3AC4 68DD3A    JZ     PR2          ; GOT IT, NOW PULSE 3*N MORE TIMES
3AC7 18        INR   D          ; INCREMENT COUNTER
3AC8 48BD3A    JNZ   PR1          ; KEEP GOING
3ACB 46CC3C    CALL   CRLF
3ACE 0E24      MVI    B,'S'       ; ERROR OUT
3AD0 46523C    CALL   CO
3AD3 46503C    CALL   BLK        ; OUTPUT A SPACE
3AD6 C2        MOV    A,C          ; DISPLAY PROM ADDRESS
3AD7 46103E    CALL   LBYTE
3ADA 44483C    JMP    LER          ; BAD PROM, ABORT
3ADD          PR2:
3ADD C3        MOV    A,D          ; MOVE COUNT RESIDUE TO A
3ADE 0411      ADI    17          ; ACTUAL COUNT OF TRIES REQUIRED
3AE0 80        ADD    A          ; COUNT = COUNT * 2
3AE1 80        ADD    A          ; COUNT = COUNT * 4
3AE2 D8        MOV    D,A
3AE3          PR3:
3AE3 469F3E    CALL   PGRM        ; OVERPROGRAM 4*N TIMES
3AE6 19        DCR    D
3AE7 48E33A    JNZ   PR3
3AEA          PR4:
3AEA 10        INR   C          ; BUMP PROM ADDRESS
3AEB 684438    JZ     START       ; PROM ADDRESS ROLLOVER, TERMINATE
3AEE 46F03D    CALL   INCHL      ; BUMP MEMORY ADDRESS
3AF1 21        DCR    E          ; DECREMENT COUNT
3AF2 48AD3A    JNZ   PR0         ; CONTINUE WITH PROGRAMMING
3AF5 444438    JMP    START
;
; 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 THOSE 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).
;

```

```

3AF8      11          DCR      C          ; GET ONE ADDRESS
3AF9      46853D     CALL     EXPR
3AFC      462B3F     CALL     RIX
3AFF      0E3A      MVI     B,':'
3B01      91        SUB     B
3B02      48FC3A     JNZ     RED0      ; SCAN TO RECORD MARK
3B05      D6        MOV     D,A      ; CLEAR CHECKSUM
3B06      467F3C     CALL     BYTE
3B09      685A3B     JZ      RED2      ; ZERO RECORD LENGTH, ALL DONE
3B0C      E0        MOV     E,A      ; E <- RECORD LENGTH
3B0D      467F3C     CALL     BYTE      ; GET MSB OF LOAD ADDRESS
3B10      360A2E00   LXI     H,ADR2
3B14      F8        MOV     M,A      ; SAVE IT
3B15      467F3C     CALL     BYTE      ; GET LSB OF LOAD ADDRESS
3B18      360B2E00   LXI     H,ADR2+1
3B1C      F8        MOV     M,A
3B1D      467F3C     CALL     BYTE      ; RECORD TYPE
3B20      360B2E00   LXI     H,ADR2+1
3B24      C7        MOV     A,M
3B25      3609      MVI     L,ADR1+1 AND OFFH
3B27      87        ADD     M
3B28      D0        MOV     C,A
3B29      31        DCR     L
3B2A      C7        MOV     A,M
3B2B      360A      MVI     L,ADR2 AND OFFH
3B2D      8F        ADC     M
3B2E      36042E00   LXI     H,HPRIME
3B32      F8        MOV     M,A
3B33      30        INR     L
3B34      FA        MOV     M,C
3B35      467F3C     CALL     BYTE      ; READ DATA
3B38      36042E00   LXI     H,HPRIME
3B3C      D7        MOV     C,M
3B3D      30        INR     L
3B3E      F7        MOV     L,M
3B3F      EA        MOV     H,C
3B40      F8        MOV     M,A      ; PUT IN MEMORY

```

```

3B41 46F03D      CALL    INCHL
3B44 CD          MOV     B,H
3B45 D6          MOV     C,L
3B46 36042E00   LXI    H,HPRIME
3B4A F9          MOV     M,B
3B4B 30          INR    L
3B4C FA          MOV     M,C
3B4D 21          DCR    E
3B4E 48353B     JNZ    RED1      ; LOOP UNTIL DONE
3B51 467F3C     CALL   BYTE      ; READ CHECKSUM
3B54 48483C     JNZ    LER       ; CHECKSUM ERROR
3B57 44FC3A     JMP    RED0      ; GET ANOTHER RECORD
3B5A          RED2:
3B5A 467F3C     CALL   BYTE      ; GET MSB OF TRANSFER ADDRESS
3B5D E0          MOV     E,A
3B5E 467F3C     CALL   BYTE
3B61 DC          MOV     D,E
3B62 E0          MOV     E,A
3B63 83          ORA    D
3B64 486638     JNZ    GU        ; TAKE THE BRANCH
3B67 444438     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.
;
3B6A          SUBS:
3B6A 11          DCR    C
3B6B 46853D     CALL   EXPR      ; GET ONE ADDRESS
3B6E 3C0D       CPI    CR
3B70 684438     JZ     START
3B73 46D43D     CALL   GETAD
3B76          SU0:
3B76 C7          MOV     A,M
3B77 46103E     CALL   LBYTE     ; DISPLAY DATA
3B7A 0E2D       MVI    B,'-'
3B7C 46523C     CALL   CO
3B7F 46493F     CALL   TI
3B82 3C20       CPI    ','
3B84 68AA3B     JZ     SU1
3B87 3C2C       CPI    ','
3B89 68AA3B     JZ     SU1
3B8C 3C0D       CPI    CR

```

```

3B8E 684438      JZ      START
3B91 1601        MVI     C,1
3B93 2600        MVI     E,0
3B95 36042E00    LXI     H,HPRIME
3B99 3E00        MVI     M,ADR2 SHR 8
3B9B 30           INR     L
3B9C 3E0A        MVI     M,ADR2 AND OFFH
3B9E 46943D      CALL    EX1
3BA1 46D43D      CALL    GETAD
3BA4 FC         MOV     M,E
3BA5 3C0D        CPI     CR          ; TEST DELIMITER
3BA7 684438      JZ      START          ; CR ENTERED AFTER LAST SUBSTITUTION
3BAA          SU1:
3BAA 46D43D      CALL    GETAD
3BAD 46F03D      CALL    INCHL
3BB0 46343F      CALL    SAVIT
3BB3 46D43D      CALL    GETAD
3BB6 44763B      JMP     SUO
;
; 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.
;
3BB9          TRAN:
3BB9 11          DCR     C
3BBA 46853D      CALL    EXPR          ; GET ONE ADDRESS
3BBD 0600        MVI     A,ENB        ; ENABLE PROM PROGRAMMER
3BBF 53          OUT     PROMC
3BC0 46D43D      CALL    GETAD        ; HL = MEM ADR
3BC3 2600        MVI     E,0          ; COUNT/PROM ADDRESS
3BC5          THO:
3BC5 C4          MOV     A,E
3BC6 2CFF        XRI     OFFH        ; INVERT ADDRESS
3BC8 55          OUT     PAD          ; SET PROM ADDRESS
3BC9 46253D      CALL    DELAY        ; WAIT FOR 6-76 BOARD TO LATCH DATA
3BCC 45          IN      PDI          ; GET PROM DATA
3BCD 2CFF        XRI     OFFH
3BCF F8          MOV     M,A          ; PUT IN MEMORY
3BD0 46F03D      CALL    INCHL        ; BUMP MEMORY POINTER
3BD3 20          INR     E          ; BUMP PROM POINTER
3BD4 48C53B      JNZ     TRU          ; GET ANOTHER BYTE
3BD7 444438      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:
3BDA      46853D      CALL      EXPR          ; GET TWO ADDRESSES
3BDD      WR10:
3BDD      46D43D      CALL      GETAD
3BE0      C6          MOV      A,L
3BE1      0410       ADI      16
3BE3      D0          MOV      C,A
3BE4      C5          MOV      A,H
3BE5      0C00       ACI      0
3BE7      C8          MOV      B,A
3BE8      C4          MOV      A,E
3BE9      92          SUB      C
3BEA      D0          MOV      C,A
3BEB      C3          MOV      A,D
3BEC      99          SBB      B
3BED      70F53B     JM       WR11          ; RECORD LENGTH = 16
3BF0      0610       MVI      A,16
3BF2      44F83B     JMP      WR12
3BF5      WR11:
3BF5      C2          MOV      A,C
3BF6      0411       ADI      17
3BF8      WR12:
3BF8      B0          ORA      A
3BF9      68443B     JZ       START
3BFc      E0          MOV      E,A          ; E <- RECORD LENGTH
3BFD      1E00       MVI      D,0          ; CLEAR CHECKSUM
3BFF      46953E     CALL    PEOL
3C02      0E20       MVI      B,' '
3C04      46B53E     CALL    PD
3C07      0E3A       MVI      B,';'
3C09      46B53E     CALL    PO
3C0C      C4          MOV      A,E
3C0D      467B3E     CALL    PBYTE          ; PUNCH LENGTH
3C10      36082E00   LXI      H,ADRI
3C14      C7          MOV      A,M
3C15      467B3E     CALL    PBYTE          ; PUNCH MSB OF ADDRESS
3C18      36092E00   LXI      H,ADRI+1
3C1C      C7          MOV      A,M
3C1D      467B3E     CALL    PBYTE          ; PUNCH LSB OF ADDRESS
3C20      A8          XRA      A
3C21      467B3E     CALL    PBYTE          ; PUNCH RECORD TYPE
3C24      WR13:
3C24      36082E00   LXI      H,ADRI
3C28      C7          MOV      A,M
3C29      30          INR      L
3C2A      F7          MOV      L,M
3C2B      E8          MOV      H,A
3C2C      C7          MOV      A,M
3C2D      46F03D     CALL    INCHL

```

```

3C30 CD          MOV      B,H
3C31 D6          MOV      C,L
3C32 36082E00   LXI      H,ADR1
3C36 F9          MOV      M,B
3C37 30          INR      L
3C38 FA          MOV      M,C
3C39 467B3E     CALL     PBYTE          ; PUNCH DATA
3C3C 21          DCR      E
3C3D 48243C     JNZ      WR13
3C40 A8          XRA      A
3C41 93          SUB      D
3C42 467B3E     CALL     PBYTE          ; PUNCH CHECKSUM
3C45 44DD38     JMP      WR10
;
; ERROR EXIT.
;
; THIS ABNORMAL EXIT IS EXECUTED FOR ALL MONITOR ERROR CONDITIONS.
;
3C48 LER:
3C48 0E2A        MVI      B,'*'
3C4A 46523C     CALL     CD
3C4D 444438     JMP      START
;
; SUBROUTINES
;
3C50 BLK:
3C50 0E20        MVI      B,' '          ; PRINT A BLANK
;
; EXTERNALLY REFERENCED ROUTINE
; CONSOLE OUTPUT CODE, VALUE EXPECTED IN B
; A,B,H,L, AND FLAGS MODIFIED
; STACK USAGE: 2 BYTES
;
3C52 CO:
3C52 36032E00   LXI      H,IOBYT          ; CONSOLE OUTPUT
3C56 C7          MOV      A,M              ; GET STATUS BYTE
3C57 2403        ANI      NOT_CMSK         ; GET CONSOLE BITS
3C59 48673C     JNZ      COO              ; TEST FOR CRT
3C5C TTYOUT:
3C5C 43          IN       TTS              ; CONSOLE = TTY
3C5D 2404        ANI      TTYBE           ;
3C5F 485C3C     JNZ      TTYOUT          ; LOOP UNTIL READY
3C62 C1          MOV      A,B
3C63 2CFF        XRI      OFFH
3C65 51          OUT     TTO              ; OUTPUT CHARACTER
3C66 07          RET                     ; RETURN
3C67 COO:
3C67 3C01        CPI      CCRT            ; CONSOLE = CRT?
3C69 48773C     JNZ      COI              ; TEST FOR BATCH
3C6C CRIOUT:
3C6C 4B          IN       CRTS           ; CONSOLE = CRT

```



```

3C6D 2404      ANI      CRTBE
3C6F 486C3C   JNZ      CRTOUT      ; LOOP UNTIL READY
3C72 C1       MOV      A,B
3C73 2CFF     XRI      OFFH
3C75 59       OUT      CKTO
3C76 07       RET
3C77
3C77 3C02     CO1:    CPI      BATCH
3C79 682C3E   JZ       LO          ; BATCH MODE, OUTPUT = LIST
3C7C 440337   JMP      COLOC       ; BRANCH TO USER CONSOLE OUTPUT
;
; READ TWO ASCII CHARACTERS, DECODE INTO 8 BITS BINARY
;
3C7F
3C7F 462B3F   BYTE:   CALL     RIX          ; READ CHAR FROM TAPE
3C82 46583E   CALL    NIBBLE       ; CONVERT ASCII TO HEX
3C85 02       RLC
3C86 02       RLC
3C87 02       RLC
3C88 02       RLC          ; SHIFT FOUR PLACES
3C89 D0       MOV      C,A
3C8A 462B3F   CALL    RIX
3C8D 46583E   CALL    NIBBLE       ; GET LOWER NIBBLE
3C90 B2       ORA      C
3C91 D0       MOV      C,A
3C92 B3       ADD      D          ; UPDATE CHECKSUM
3C93 D8       MOV      D,A
3C94 C2       MOV      A,C
3C95 07       RET          ; RETURN
;
; EXTERNALLY REFERENCED ROUTINE
; CONSOLE INPUT CODE, VALUE RETURNED IN A
; A,B,H,L, AND FLAGS MODIFIED
; STACK USAGE: 2 BYTES
;
3C96
3C96 36032E00 C1:     LXI      H,IOBYT     ; CONSOLE INPUT
3C9A C7       MOV      A,M          ; GET STATUS BYTE
3C9B 2403    ANI      NUT_CMSK    ; GET CONSOLE BITS
3C9D 48AA3C   JNZ      C11         ; TEST FOR CRT
3CA0
3CA0 43       TTYIN:  IN       TTS          ; TTY STATUS PORT
3CA1 2401    ANI      TTYDA       ; CHECK FOR DATA AVAILABLE
3CA3 48A03C   JNZ      TTYIN
3CA6 41       IN       TTI          ; READ THE CHARACTER
3CA7
3CA7 2CFF     CI0:    XRI      OFFH
3CA9 07       RET          ; RETURN
3CAA
3CAA 3C01     CI1:    CPI      CCRT       ; CONSOLE = CRT?
3CAC 48B93C   JNZ      C12         ; TEST FOR BATCH

```

```

3CAF          CRTIN:
3CAF 4B          IN      CRTS          ; CRT STATUS PORT
3CB0 2401        ANI     CRIDA         ; CHECK FOR DATA AVAILABLE
3CB2 48AF3C      JNZ     CRTIN        ; NOT READY, CONTINUE LOOPING
3CB5 49          IN      CRTI         ; READ THE CHARACTER
3CB6 44A73C      JMP     CIO
3CB9          C12:
3CB9 3C02        CPI     BATCH
3CBB 68DB3E      JZ      R1              ; BATCH MODE, INPUT = READER
3CBE 440037      JMP     CILOC          ; CONSOLE = USER DEVICE
;
; CONVERT 4 BIT HEX VALUE TO ASCII CHARACTER
;
3CC1          CONV:
3CC1 3C0A        CPI     10
3CC3 70C83C      JM      CNO              ; LESS THAN 10, (0-9)
3CC6 0407        ADI     'A'-'0'-10      ; ADJUST OF (A-F)
3CC8          CNO:
3CC8 0430        ADI     '0'              ; ADD BIAS FOR ASCII
3CCA C8          MOV     B,A
3CCB 07          RET
;
; TYPE CARRIAGE RETURN AND LINE FEED ON CONSOLE
;
3CCC          CRLF:
3CCC 0E0D        MVI     B,CR          ; <CR>
3CCE 46523C      CALL    CU
3CD1 0E0A        MVI     B,LF          ; <LF>
3CD3 44523C      JMP     CD
;
; EXTERNALLY REFERENCED ROUTINE
; CONSOLE INPUT STATUS CODE
; A, FLAGS MODIFIED
; STACK USAGE: 2 BYTES
;
3CD6          CSTS:
3CD6 36032E00    LXI     H,I0BYT          ; CONSOLE INPUT STATUS
3CDA C7          MOV     A,M              ; GET STATUS BYTE
3CDB 2403        ANI     NOT CMSK        ; CONSOLE = TTY?
3CDD 48E43C      JNZ     CS0              ; CONSOLE = CRT
3CE0 43          IN      TTS              ; GET TTY STATUS
3CE1 44EA3C      JMP     CS1
3CE4          CS0:
3CE4 3C01        CPI     CCRT
3CE6 48F23C      JNZ     CSJ
3CE9 4B          IN      CRTS          ; GET CRT STATUS
3CEA          CS1:
3CEA 2401        ANI     TTYDA
3CEC 0600        MVI     A,FALSE          ; RETURN FALSE IF NO DATA AVAILABLE
3CEE          CS2:
3CEE 0B          RNZ

```

```

3CEF 2CFF      XRI  OFFH
3CF1 07       RET           ; RETURN
3CF2          CS3:
3CF2 3C02      CPI  BATCH
3CF4 06FF      MVI  A,TRUE
3CF6 68EE3C    JZ   CS2
3CF9 441837    JMP  CSLUC
;
; READ BNPF TAPE RECORD, BUILD BYTE, STORE IN C-REGISTER
; IF ERROR, ABORT COMMAND
;
3CFC          DECODE:
3CFC 462B3F    CALL  RIX           ; READ TAPE
3CFF 3C42      CPI  'B'           ; SCAN FOR 'B'
3D01 48FC3C    JNZ  DECODE
3D04 1601      MVI  C,1           ; INITIALIZE MEMORY
3D06          DC0:
3D06 462B3F    CALL  RIX           ; GET DATA
3D09 3C4E      CPI  'N'           ; CHECK FOR 'N'
3D0B 481D3D    JNZ  DC2           ; NO, CHECK FOR 'P'
; CARRY = 0
3D0E          DC1:
3D0E C2        MOV  A,C           ; SHIFT IN DATA BIT
3D0F 12        RAL
3D10 D0        MOV  C,A
3D11 40063D    JNC  DC0           ; IF CARRY IS SET, 8 BITS READ
3D14 462B3F    CALL  RIX           ; TEST FOR REQ'D 'F'
3D17 3C46      CPI  'F'
3D19 48483C    JNZ  LER
3D1C 07        RET           ; RETURN
3D1D          DC2:
3D1D 04B0      ADI  -'P'
3D1F 48483C    JNZ  LER           ; ERROR
3D22 440E3D    JMP  DC1           ; CARRY IS SET
;
; 1.0 MS DELAY (INCLUDING CALL, DCR OR INC, AND JUMP IN CALLING LOOP)
;
3D25          DELAY:
3D25 0E17      MVI  B,DLY
3D27          DL0:
3D27 09        DCR  B
3D28 48273D    JNZ  DL0
3D2B 07        RET           ; RETURN
;
; CONVERT BINARY NUMBER TO A STRING OF ASCII DIGITS
; HL - BINARY NUMBER
; DE - DIVISOR (DESCENDING POWERS OF 10)
; C - LEADING ZERO SUPPRESSION CHARACTER
; A,B - TEMPORARIES
;
3D2C          DIGIT:

```

```

3D2C 0E30          MVI    B,'0'          ; INITIALIZE CHARACTER
3D2E              DG0:   MOV    A,L          ; SUB DENOM (DE) FROM NUMERATOR (HL)
3D2E C6           MOV    A,L
3D2F 94           SUB    E
3D30 F0           MOV    L,A
3D31 C5           MOV    A,H
3D32 9B           SBB    D
3D33 E8           MOV    H,A
3D34 603B3D       JC     DG1          ; NEGATIVE RESULT, ALL DONE
3D37 08           INR    B          ; COUNT NUMBER OF SUBTRACTS
3D38 442E3D       JMP    DG0
3D3B              DG1:   MOV    A,L
3D3B C6           MOV    A,L
3D3C 84           ADD    E
3D3D F0           MOV    L,A
3D3E C5           MOV    A,H
3D3F 8B           ADC    D
3D40 E8           MOV    H,A
3D41 C1           MOV    A,B
3D42 3C30         CPI    '0'          ; CHECK FOR LEADING ZERO SUPPRESSION
3D44 48503D       JNZ   DG3
3D47 CA           MOV    B,C
3D48              DG2:   MOV    D,H
3D48 DD           MOV    E,L
3D49 E6           MOV    PO
3D4A 46B53E       CALL  PO
3D4D EB           MOV    H,D
3D4E F4           MOV    L,E
3D4F 07           RET          ; PUNCH CHARACTER
3D50              DG3:   MVI    C,'0'
3D50 1630         MVI    C,'0'
3D52 44483D       JMP    DG2

;
; ENCODE A BPNF WORD AND PUNCH IT
;
3D55              ENCODE: MOV    D,H          ; SAVE HL
3D55 DD           MOV    E,L
3D56 E6           MOV    E,L
3D57 0E42         MVI    B,'B'        ; PUNCH A 'B'
3D59 46B53E       CALL  PO
3D5C 1608         MVI    C,8          ; 8 BIT COUNT
3D5E              EN0:   MOV    H,D
3D5E EB           MOV    H,D
3D5F F4           MOV    L,E
3D60 0609         MVI    A,9
3D62 92           SUB    C
3D63 C8           MOV    B,A
3D64 C7           MOV    A,M
3D65              EN1:   RLC
3D65 02           RLC
3D66 09           DCR    B

```

```

3D67 48653D      JNZ  EN1
3D6A 40803D      JNC  EN3
3D6D 0E50        MVI  B,'P'
3D6F          EN2:
3D6F 46B53E      CALL PO
3D72 11          DCR  C
3D73 485E3D      JNZ  EN0
3D76 0E46        MVI  B,'F'
3D78 46B53E      CALL PU
3D7B 0E20        MVI  B,' '
3D7D 44B53E      JMP  PO
3D80          EN3:
3D80 0E4E        MVI  B,'N'
3D82 446F3D      JMP  EN2
;
; EVALUATE EXPRESSION: <EXPR>,<EXPR>,<EXPR>
;
3D85          EXPR:
3D85 DD          MOV  D,H          ; SAVE HL
3D86 Eb        MOV  E,L
3D87 36042E00  LXI  H,HPRIME
3D88 Fb        MOV  M,D
3D8C 30        INR  L
3D8D FC        MOV  M,E
3D8E 1E00      MVI  D,0          ; D,E=0
3D90 E3        MOV  E,D
3D91          EX0:
3D91 46493F    CALL TI          ; GET A CHARACTER
3D94          EX1:
3D94 C8        MOV  B,A
3D95 46583E    CALL NIBBLE     ; CONVERT TO HEX
3D98 60AE3D    JC   EX2       ; NOT LEGAL CHAR, TREAT AS DELIMITER
3D9B C8        MOV  B,A
3D9C 46423F    CALL SLDE
3D9F 46423F    CALL SLDE
3DA2 46423F    CALL SLDE
3DA5 46423F    CALL SLDE
3DA8 C1        MOV  A,B
3DA9 B4        ORA  E
3DAA E0        MOV  E,A
3DAB 44913D    JMP  EX0       ; GET ANOTHER CHARACTER
3DAE          EX2:
3DAE 36042E00  LXI  H,HPRIME
3DB2 C7        MOV  A,M
3DB3 30        INR  L
3DB4 F7        MOV  L,M
3DB5 E8        MOV  H,A
3DB6 FB        MOV  M,D          ; PUT MSB IN MEMORY
3DB7 30        INR  L
3DB8 FC        MOV  M,E          ; PUT LSB IN MEMORY
3DB9 30        INR  L

```

```

3DBA C1      MOV    A,B          ; RESTORE DELIMITER
3DBB 3C2C    CPI     ' ',''
3DBD 68CF3D  JZ     EX3
3DC0 3C20    CPI     ' ',''
3DC2 68CF3D  JZ     EX3
3DC5 3C0D    CPI     CR
3DC7 48483C  JNZ    LER
3DCA 11      DCR     C
3DCB 48483C  JNZ    LER
3DCE 07      RET
3DCF        EX3:
3DCF 11      DCR     C
3DD0 48853D  JNZ    EXPR
3DD3 07      RET
;
; GET ADDRESS FROM MEMORY AND PUT IN HL AND DE
;
3DD4        GETAD:
3DD4 36082E00 LXI    H,ADRI          ; ADDRESS OF FIRST PARAMETER
3DD8 CF      MOV    B,M          ; MSB OF LOW ADDRESS
3DD9 30      INR    L
3DDA D7      MOV    C,M          ; LSB OF LOW ADDRESS
3DDB 30      INR    L
3DDC DF      MOV    D,M          ; MSB OF HIGH ADDRESS
3DDD 30      INR    L
3DDE E7      MOV    E,M          ; LSB OF HIGH ADDRESS
3DDF E9      MOV    H,B          ; RESET HL
3DE0 F2      MOV    L,C
3DE1 07      RET
;
; COMPARE HL WITH DE:
; IF HL < DE THEN CARRY = 0;
; IF HL = DE THEN CARRY = 0;
; IF HL > DE THEN CARRY = 1;
;
3DE2        HILO:
3DE2 46F03D  CALL   INCHL          ; BUMP HL
3DE5 C4      MOV    A,E          ; DE - HL, SET/RESET CARRY
3DE6 96      SUB    L
3DE7 C3      MOV    A,D
3DE8 9D      SBB    H
3DE9 07      RET          ; RETURN
;
; CONVERT NIBBLE IN A-REGISTER TO ASCII IN A-REGISTER
; AND PRINT ON TELEPRINTER
;
3DEA        HXD:
3DEA 46C13C  CALL   CONV
3DED 44523C  JMP    CD
;
; INCREMENT H AND L

```

```

;
3DF0      INCHL:
3DF0      30          INR      L
3DF1      08          RNZ
3DF2      28          INR      H
3DF3      07          RET

;
; EXTERNALLY REFERENCED ROUTINE
; I/O SYSTEM STATUS CODE
; STATUS BYTE RETURNED IN A
; STACK USAGE: 2 BYTES
;
3DF4      IOCHK:
3DF4      36032E00    LXI      H,IOBYT
3DF8      C7          MOV      A,M          ; GET STATUS BYTE
3DF9      07          RET              ; RETURN

;
; EXTERNALLY REFERENCED ROUTINE
; SET I/O CONFIGURATION
; VALUE EXPECTED IN B
; STACK USAGE: 2 BYTES
;
3DFA      IOSET:
3DFA      36032E00    LXI      H,IOBYT          ; POINT HL AT IOBYT
3DFE      F9          MOV      M,B
3DFF      07          RET              ; RETURN

;
; PRINT CONTENTS OF HL IN HEX ON CONSOLE DEVICE
;
3E00      LADR:
3E00      DD          MOV      D,H
3E01      E6          MOV      E,L
3E02      C3          MOV      A,D
3E03      46103E     CALL     LBYTE
3E06      C4          MOV      A,E
3E07      46103E     CALL     LBYTE
3E0A      46503C     CALL     BLK
3E0D      EB          MOV      H,D
3E0E      F4          MOV      L,E
3E0F      07          RET

;
; LIST A BYTE AS 2 ASCII CHARACTERS
;
3E10      LBYTE:
3E10      DD          MOV      C,A          ; SAVE A COPY OF A
3E11      0A          RRC
3E12      0A          RRC
3E13      0A          RRC
3E14      0A          RRC
3E15      240F      ANI      0FH          ; UPPER 4 BITS
3E17      46EA3D     CALL     HXD

```

```

3E1A C2          MOV    A,C          ; RETRIEVE ORIGINAL VALUE
3E1B 240F        ANI    0FH         ; LOWER 4 BITS
3E1D 44EA3D     JMP    HXD
;
; PUNCH 6 INCHES OF LEADER
;
3E20          LEAD:
3E20 163C        MVI    C,60          ; SET TO PUNCH 6 INCHES OF NULLS
3E22          LEO:
3E22 0E00        MVI    B,0
3E24 46B53E     CALL   PO
3E27 11         DCR    C
3E28 48223E     JNZ    LEO
3E2B 07         RET              ; RETURN
;
; EXTERNALLY REFERENCED ROUTINE
; LIST OUTPUT CODE
; VALUE EXPECTED IN B
; A,B,H,L, AND FLAGS MODIFIED
; STACK USAGE: 2 BYTES
;
3E2C          LO:
3E2C 36032E00    LXI    H,IOBYT          ; LIST OUTPUT
3E30 C7          MOV    A,M          ; GET STATUS BYTE
3E31 24C0        ANI    NOT LMSK        ; GET LIST BITS
3E33 685C3C     JZ     TTYOUT         ; LIST = TTY
3E36 3C40        CPI    LCRT          ; LIST = CRT
3E38 686C3C     JZ     CRTOUT        ; LIST = CRT
3E3B 3C80        CPI    LUSE1         ; TEST FOR USER DEFINED LIST DEVICE
3E3D 681237     JZ     L1LOC         ; BRANCH TO USER DEVICE
3E40 441537     JMP    L2LOC         ; ELSE BRANCH TO USER LIST 2
;
; EXTERNALLY REFERENCED ROUTINE
; RETURN ADDRESS OF END OF MEMORY TO USER
; A,B,C,H,L, AND FLAGS MODIFIED
; VALUE RETURNED IN (C,A)
; STACK USAGE: 2 BYTES
;
3E43          MEMCK:
3E43 36002E00    LXI    H,0
3E47          MO:
3E47 CF          MOV    B,M
3E48 3EAA        MVI    M,0AAH
3E4A C7          MOV    A,M
3E4B F9          MOV    M,B
3E4C 28          INR    H
3E4D 3CAA        CPI    0AAH
3E4F 68473E     JZ     M0
3E52 29         DCR    H
3E53 29         DCR    H
3E54 31         DCR    L

```



```

3E55 C6      MOV    A,L
3E56 D5      MOV    C,H
3E57 07      RET

;
; 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.
;
3E58      NIBBLE:
3E58 1430    SUI    '0'
3E5A 23      RC
3E5B 04E9    ADI    '0'-'G'
3E5D 23      RC
3E5E 0406    ADI    6
3E60 50663E  JP    N10
3E63 0407    ADI    7
3E65 23      RC
3E66      N10:
3E66 040A    ADI    10
3E68 B0      ORA    A
3E69 07      RET                ; RETURN

;
; DISREGARD NOISE CHARACTERS
;
3E6A      NOISE:
3E6A 46493F  CALL   TI
3E6D 3C3D    CPI    '='
3E6F 486A3E  JNZ    NOISE
3E72      NOO:
3E72 46493F  CALL   TI
3E75 3C20    CPI    ' '
3E77 68723E  JZ     NOO
3E7A 07      RET                ; RETURN

;
; PUNCH A BYTE AS 2 ASCII CHARACTERS
;
3E7B      PBYTE:
3E7B D0      MOV    C,A
3E7C 0A      RRC
3E7D 0A      RRC
3E7E 0A      RRC
3E7F 0A      RRC
3E80 240F    ANI    0FH
3E82 46C13C  CALL   CONV
3E85 46B53E  CALL   PO
3E88 C2      MOV    A,C
3E89 240F    ANI    0FH
3E8B 46C13C  CALL   CONV
3E8E 46B53E  CALL   PO
3E91 C2      MOV    A,C
3E92 83      ADD    D

```

```

3E93  D8          MOV     D,A
3E94  07          RET     ; RETURN
;
; PUNCH CR,LF
;
PEOL:
3E95  0E0D        MVI     B,CR
3E97  46B53E     CALL    PO
3E9A  0E0A        MVI     B,LF
3E9C  44B53E     JMP     PO
;
; PULSE A PROM LOCATION
; HL POINTS TO DATA IN MEMORY
; PROM ADDRESS IS ALREADY SET
;
PGRM:
3E9F  C7          MOV     A,M          ; GET DATA FROM MEMORY
3EA0  2CFF        XRI     OFFH        ; INVERT IT
3EA2  57          OUT     PDO          ; OUTPUT IT
3EA3  0680        MVI     A,PROGO     ; PULSE PROM PROGRAMMER
3EA5  53          OUT     PROMC
3EA6  0600        MVI     A,PRONO
3EA8  53          OUT     PROMC
3EA9  C3          MOV     A,D          ; SAVE D
3EAA  1E14        MVI     D,LDLY      ; DELAY 20 MS. FOR PROGRAMMER SETTLING
3EAC  PG0:       CALL    DELAY
3EAC  46253D     DCR     D
3EAF  19          JNZ     PG0
3EB0  49AC3E     MOV     D,A
3EB3  08          RET
3EB4  07
;
; EXTERNALLY REFERENCED ROUTINE
; PUNCH OUTPUT CODE, VALUE EXPECTED IN B
; A,B,H,L, AND FLAGS MODIFIED
; STACK USAGE: 2 BYTES
;
PO:
3EB5  36032E00    LXI     H,IOBYT    ; PUNCH OUTPUT
3EB9  C7          MOV     A,M          ; GET STATUS BYTE
3EBA  2430        ANI     NOT PMSK   ; GET PUNCH BITS
3EBC  685C3C     JZ      TTYOUT     ; NO, PUNCH = TTY
3EBF  3C10        CPI     PPTP       ; TEST FOR PTP
3EC1  48D33E     JNZ     PO1        ; TEST FOR USER DEVICE(S)
3EC4  PO0:       ; PUNCH = PTP
3EC4  43          IN      PTPS       ; GET STATUS
3EC5  2440        ANI     PRDY       ; CHECK STATUS
3EC7  68C43E     JZ      PO0        ; LOOP UNTIL READY
3ECA  C1          MOV     A,B
3ECB  57          OUT     PTPU
3ECC  060A        MVI     A,PTPGO    ; START PUNCH

```

```

3ECE 53          OUT      PTPC
3ECF 0608       MVI      A,PTPND      ; STOP PUNCH
3ED1 53          OUT      PTPC
3ED2 07          RET
3ED3
3ED3 3C20       CPI      PUSE1
3ED5 680C37     JZ       P1LOC
3ED8 440F37     JMP      P2LOC
;
; EXTERNALLY REFERENCED ROUTINE
; READER INPUT CODE
; VALUE RETURNED IN A
; A,B,H,L, AND FLAGS MODIFIED
; STACK USAGE: 4 BYTES
;
3EDB           RI:
3EDB 36032E00   LXI      H,IOBYT      ; READER INPUT
3EDF C7         MOV      A,M          ; POINT HL AT IOBYT
3EE0 240C       ANI      NOT RMSK      ; READER = PTR?
3EE2 48033F     JNZ      RI3          ; BRANCH TO PTR ROUTINE
3EE5 0609       MVI      A,TTYGO      ; READER = TTY
3EE7 53         OUT      TTC
3EE8 0608       MVI      A,TTYNO
3EEA 53         OUT      TTC
3EEB 2EFA       MVI      H,TOUT      ; SET TIMER FOR READER TIMEOUT
3EED           RI0:
3EED 43         IN       TTS
3EEE 2401       ANI      TTYDA
3EF0 68FF3E     JZ       RI2          ; DATA IS READY
3EF3 46253D     CALL    DELAY        ; DELAY 1.0 MS
3EF6 29         DCR      H
3EF7 48ED3E     JNZ      RI0
3EFA           RI1:
3EFA 60         ORA      A
3EFB 0601       MVI      A,1
3EFD 1A         RAR
3EFE 07         RET          ; SET CARRY INDICATING EOF
3EFF           RI2:
3EFF 41         IN       TTI
3F00 2CFF       XRI      OFFH        ; INVERT DATA, CLEAR CARRY
3F02 07         RET          ; RETURN
3F03           RI3:
3F03 3C04       CPI      RPTR
3F05 48233F     JNZ      RI6
3F08 060C       MVI      A,PTRGO
3F0A 53         OUT      PTRC
3F0B 0608       MVI      A,PTRNO
3F0D 53         OUT      PTRC
3F0E 2EFA       MVI      H,TOUT      ; SET TIMER FOR READER TIMEOUT
3F10           RI4:
3F10 43         IN       PTRS

```

```

3F11 2420      ANI    PTRDA
3F13 48203F    JNZ    RI5
3F16 46253D    CALL   DELAY      ; DELAY 1.0 MS.
3F19 29        DCR    H
3F1A 48103F    JNZ    RI4
3F1D 44FA3E    JMP    RI1
3F20          RIS:
3F20 47        IN     PTRI      ; GET THE DATA
3F21 B0        ORA    A          ; CLEAR CARRY
3F22 07        RET                    ; RETURN
3F23          RI6:
3F23 3C08      CPI    RUSE1
3F25 680637    JZ     RILOC
3F28 440937    JMP    R2LOC
;
; GET CHARACTER FROM READER, MASK OFF PARITY BIT
;
3F2B          RIX:
3F2B 46D83E    CALL   RI
3F2E 60483C    JC     LER
3F31 247F      ANI    7FH
3F33 07        RET                    ; RETURN
;
; SAVE HL AND DE IN MEMORY
;
3F34          SAVIT:
3F34 CD        MOV    H,H      ; SAVE HL
3F35 D6        MOV    C,L
3F36 36082E00 LXI   H,ADR1   ; POINT TO FIRST PARAMETER
3F3A F9        MOV    M,B      ; MSB OF LOW ADDRESS
3F3B 30        INR    L
3F3C FA        MOV    M,C      ; LSB OF LOW ADDRESS
3F3D 30        INR    L
3F3E FB        MOV    M,D      ; MSB OF HIGH ADDRESS
3F3F 30        INR    L
3F40 FC        MOV    M,E      ; LSB OF HIGH ADDRESS
3F41 07        RET
;
; SHIFT DE LEFT 1 PLACE
;
3F42          SLDE:
3F42 C4        MOV    A,E
3F43 80        ADD    A
3F44 E0        MOV    E,A
3F45 C3        MOV    A,D
3F46 88        ADC    A
3F47 D8        MOV    D,A
3F48 07        RET
;
; INPUT FROM CONSOLE, ECHOED AND RETURNED IN A
;

```

```

3F49          TI:
3F49 46963C   CALL    CI
3F4C 247F     ANI     7FH
3F4E C8       MOV     B,A
3F4F 46523C   CALL    CO
3F52 C1       MOV     A,B
3F53 07       RET
                ; RETURN
                ;
                ; I/O SYSTEM PHYSICAL DEVICE TABLES
                ; 2 BYTES/ENTRY
                ; BYTE 0 = IDENTIFYING CHARACTER
                ; BYTE 1 = DEVICE SELECT BIT PATTERN
                ;
3F54          ICT:
3F54 5400     DB     'T',CTTY      ; CONSOLE = TTY
3F56 4301     DB     'C',CCRT      ; CONSOLE = CRT
3F58 4202     DB     'B',BATCH   ; BATCH MODE CONSOLE = READ,LIST
3F5A 3103     DB     '1',CUSE     ; USER DEFINED CONSOLE DEVICE
3F5C          IRT:
3F5C 5400     DB     'T',RTTY     ; READER = TTY
3F5E 5004     DB     'P',RPTR     ; READER = PTR
3F60 3108     DB     '1',RUSE1    ; USER DEFINED READER DEVICE 1
3F62 320C     DB     '2',RUSE2    ; USER DEFINED READER DEVICE 2
3F64          OPT:
3F64 5400     DB     'T',PTY      ; PUNCH = TTY
3F66 5010     DB     'P',PPTP     ; PUNCH = PTP
3F68 3120     DB     '1',PUSE1    ; USER DEFINED PUNCH DEVICE 1
3F6A 3230     DB     '2',PUSE2    ; USER DEFINED PUNCH DEVICE 2
3F6C          OLT:
3F6C 5400     DB     'T',LTTY     ; LIST = TTY
3F6E 4340     DB     'C',LCRT     ; LIST = CRT
3F70 3180     DB     '1',LUSE1    ; USER DEFINED LIST DEVICE 1
3F72 32C0     DB     '2',LUSE2    ; USER DEFINED LIST DEVICE 2
                ;
                ; END OF PROGRAM
                ;
                END

```

NO PROGRAM ERRORS

## SYMBOL TABLE

\* 1

A	0000	ADR1	0008	ADR2	000A	ADR3	000C
AS0	38E4	AS1	38F2	AS2	3900	ASS0	38A3
B	0001	BATCH	0002	BEGIN	3829	BIAS	0008
B	3C50	BNO	3917	BN1	394C	BNEP	390E
B	3C7F	C	0002	CBRCH	0006	CCRT	0001
CI	3C96	CI0	3CA7	CI1	3CAA	CI2	3CB9
CILUC	3700	CM0	3976	CM1	39A2	CMSK	00FC
C	3CC8	CO	3C52	CO0	3C67	CO1	3C77
C	3703	COMP	3967	CONV	3CC1	CR	000D
C	3CCC	CRTR	0004	CRTDA	0001	CRTI	0004
CRTIN	3CAF	CRT0	000C	CRTOU	3C6C	CRTS	0005
C	3CE4	CS1	3CEA	CS2	3CEE	CS3	3CF2
C	3718	CSTS	3CD6	CTTY	0000	CUSE	0003
D	0003	DC0	3D06	DC1	3D0E	DC2	3D1D
DEBUG	0000	DECOD	3CFC	DELAY	3D25	DG0	3D2E
DG1	3D3B	DG2	3D48	DG3	3D50	DIO	39AF
D	39B8	DIGIT	3D2C	DISP	39AC	DL0	3D27
D	0017	DSB	0008	E	0004	ENO	3D5E
E	3D65	EN2	3D6F	EN3	3D80	ENB	0000
ENCOD	3D55	EOF	39D7	EX0	3D91	EX1	3D94
EX2	3DAE	EX3	3DCF	EXPR	3D85	FALSE	0000
F	3A0C	FILL	3A03	FIRST	0000	GO	3866
F	3DD4	GOTO	3A16	H	0005	HEXN	3A1D
H	3DE2	HPRIM	0004	HXD	3DEA	ICT	3F54
INCHL	3DF0	INIT	0000	IOBYT	0003	IOCHK	3DF4
MOD	38B1	IUSET	3DFA	IRT	3F5C	L	0006
L	3712	L2LOC	3715	LADR	3E00	LBYTE	3E10
L	0040	LDLY	0014	LE0	3E22	LEAD	3E20
LER	3C48	LF	000A	LMSK	003F	LO	3E2C
LO0	3A4E	LOAD	3A48	LTTY	0000	LUSE1	0080
L	00C0	LVER	000B	M	0007	MO	3E47
L	3E43	MODIO	0728	MOVE	3A62	MVO	3A66
L	3E66	NIBBL	3E58	NO0	3E72	NOISE	3E6A
NULL	3A8E	OLT	3F6C	OPT	3F64	P1LOC	370C
P2LOC	370F	PAD	000A	PBITA	0080	PBYTE	3E78
P	0002	PDI	0002	PDO	000B	PEOL	3E95
P	3EAC	PGRM	3E9F	PMSK	00CF	PO	3E85
P	3EC4	PO1	3ED3	PPTP	0010	PRO	3AAD
PRI	3A8D	PR2	3ADD	PR3	3AE3	PR4	3AEA
P	0040	PROG	3A94	PROGD	0080	PROMC	0089
P	0000	PTPC	0009	PTPGD	000A	PTPNO	0008
P	000B	PTPS	0001	PTRC	0009	PTRDA	0028
PTRGO	000C	PTRI	0003	PTRNO	0008	PTRS	0001
P	0000	PUSE1	0020	PUSE2	0030	R1LOC	3706
P	3709	RBIT	0001	RCMD	0004	READ	3AF8
P	3AFC	RED1	3B35	RED2	3B5A	RI	3ED8
P	3EED	RI1	3EFA	RI2	3EFF	RI3	3F03
RI4	3F19	RI5	3F20	RI6	3F23	RIX	3F2B
RMSK	00F3	RPTR	0004	RTTY	0000	RUSE1	0088
R	000C	SAVIT	3F34	SLDE	3F42	START	3844
R	3B76	SU1	3BAA	SUBS	3B6A	TBL	3875
RI	3F49	TOUT	00FA	TR0	3BC5	TRAN	38B9

TRUE	FFFF	TTC	0009	TTI	0000	TTO	0008
TTS	0001	TTYBE	0004	TTYDA	0001	TTYGD	0009
TTYIN	3CA0	TTYNO	0008	TTYOU	3C5C	VER	001E
VERO	3835	VERS	381E	WRIO	3BDD	WR11	3BF5
WRI2	3BF8	WRI3	3C24	WRITE	3BDA		

\* 02

SCANO	38BF	TEST	38D2
-------	------	------	------

\* 03

\* 04

\* 05