MDS MONITOR LISTING VERSION 1.1

To be used with MDS
Monitor Bootstrap
Version 1.2

```
                        :SELECTION CODES FOR USER I/O ENTRY POINTS
                        ;
0000                    UCI     EQU     (CILOC-XTBL)/3
0001                    UCO     EQU     (COLOC-XTBL)/3
0002                    UR1     EQU     (R1LOC-XTBL)/3
0003                    UR2     EQU     (R2LOC-XTBL)/3
0004                    UP1     EQU     (P1LOC-XTBL)/3
0005                    UP2     EQU     (P2LOC-XTBL)/3
0006                    UL1     EQU     (L1LOC-XTBL)/3
0007                    UCS     EQU     (CSLOC-XTBL)/3
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ; COMPUTE BASE ADDRESS OF MONITOR PROPER
                        ;
F800                    BASE    SET     0F800H          ; BASE ADDRESS OF MONITOR
                                                        ; TOP 2K OF ADDRESS SPACE
     1                          IF      DEBUG
     1                  BASE    SET     800H            ; DEBUG IN RAM
                                ENDIF
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ; START OF MONITOR PROPER
                        ;
F800                            ORG     BASE
                        ;
                        ; BRANCH TABLE FOR I/O SYSTEM
                        ;
F800                    BEGIN:
F800    C330F8                  JMP     START           ; RESET ENTRY POINT
F803    C3A2FC                  JMP     CI              ; CONSOLE INPUT
F806    C3D5FC                  JMP     RI              ; READER INPUT
F809    C33AFD                  JMP     CO              ; CONSOLE OUTPUT
F80C    C36FFD                  JMP     PO              ; PUNCH OUTPUT
F80F    C3A1FD                  JMP     LO              ; LIST OUTPUT
F812    C3C1FD                  JMP     CSTS            ; CONSOLE INPUT STATUS
F815    C3E8FD                  JMP     IOCHK           ; I/O SYSTEM STATUS
F818    C3ECFD                  JMP     IOSET           ; SET I/O CONFIGURATION
F81B    C3F1FD                  JMP     MEMCK           ; COMPUTE SIZE OF MEMORY
F81E    C3F9FD                  JMP     IODEF           ; DEFINE USER I/O ENTRY POINTS
F821    C316FE                  JMP     SPCL            ; OPEN, CLOSE, REWIND ENTRY POINT
F824    0826                    DW      DATE            ; DATE STAMP FOR MONITOR ROM
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ; ERROR EXIT.
                        ;
                        ; THIS ABNORMAL EXIT IS EXECUTED FOR ALL MONITOR ERROR CONDITIONS.
                        ; THE STACK POINTER VALUE IS RECOMPUTED BECAUSE OF ITS UNKNOWN STATE
```

```
                        ; AS THE RESULT OF A COMMAND OR DATA ERROR.
                        ;
                        ; REGISTER USAGE
                        ;
                        ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                        ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                        ; U = USED AS INPUT.
                        ;
                        ;       A -
                        ;       B -             C - S
                        ;       D -             E -
                        ;       H - X           L - X
                        ;       CARRY - X       ZERO - X
                        ;       SIGN - X        PARITY - X
                        ;       SP - S          PC -
                        ;       STACK USAGE: 2 BYTES
                        ;
F826                    ERROR:
    1                   +       GET     TOS
F826 1 2A0400           +       LHLD    MEMTOP
F829 1 2EC8             +       MVI     L,TOS AND 0FFH
F82B   F9                       SPHL
F82C   CD64FD                   CALL    COMC
F82F   2A                       DB      '*'
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ; MAIN COMMAND LOOP.
                        ;
                        ; THIS LOOP IS THE STARTING POINT OF ALL COMMAND SEQUENCES.
                        ; IN THIS CODE INTERRUPTS ARE ENABLED AND 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.
                        ;
                        ; REGISTER USAGE
                        ;
                        ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                        ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                        ; U = USED AS INPUT.
                        ;
                        ;       A - X
                        ;       B - X           C - S
                        ;       D - S           E - S
                        ;       H - X           L - X
                        ;       CARRY - X       ZERO - X
                        ;       SIGN - X        PARITY - X
                        ;       SP - X          PC - X
                        ;       STACK USAGE: 4 BYTES
                        ;
```

```
F830                START:
F830   DBFF                 IN      RTC             ; LOOP UNTIL BOOTSTRAP MODE IS DISABLED
F832   E602                 ANI     BOOT
F834   C230F8               JNZ     START
F837   FB                   EI                      ; ENABLE INTERRUPTS
F838   CD40FE               CALL    CRLF            ; TYPE <CR>,<LF>
F83B   CD64FD               CALL    COMC            ; OUTPUT A PERIOD
F83E   2E                   DB      '.'
F83F   CDE8FF               CALL    TI              ; GET A CHARACTER, ECHO IT.
F842   D641                 SUI     'A'             ; TEST FOR A-X
F844   FA30F8               JM      START           ; LT A, IGNORE
F847   0E02                 MVI     C,2             ; C IS SET UP FOR 2 PARAMETER COMMANDS
F849   1130F8               LXI     D,START         ; SET UP PSEUDO RETURN ADDRESS
F84C   D5                   PUSH    D               ; TO SIMULATE EFFECT OF CALL
     1                +     CASE    CTBL,LCT        ; BRANCH TO PROCESSING ROUTINE
F84D 1 215FF8         +     LXI     H,CTBL
F850 1 FE18           +     CPI     LCT             ; TEST FOR OVERRUN
F852 1 F226F8         +     JP      ERROR
F855 1 5F             +     MOV     E,A             ; MOVE INDEX TO DE
F856 1 1600           +     MVI     D,0
F858 1 19             +     DAD     D               ; ADD BASE + 2 * INDEX -> HL
F859 1 19             +     DAD     D
F85A 1 7E             +     MOV     A,M             ; GET LSB OF BRANCH LOCATION
F85B 1 23             +     INX     H
F85C 1 66             +     MOV     H,M             ; GET MSB OF BRANCH LOCATION
F85D 1 6F             +     MOV     L,A
F85E 1 E9             +     PCHL                    ; TAKE THE BRANCH
                     ;
                     ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                     ;
                     ; COMMAND BRANCH TABLE.
                     ;
                     ; THIS TABLE CONTAINS THE ADDRESSES OF THE ENTRY POINTS OF
                     ; ALL THE COMMAND PROCESSING ROUTINES.  NOTE THAT AN ENTRY TO 'ERROR'
                     ; IS AN ERROR CONDITION, I.E., NO COMMAND CORRESPONDING TO THAT
                     ; CHARACTER EXISTS.
                     ;
F85F                 CTBL:
F85F   8FF8                 DW      ASSIGN          ; A - ASSIGN I/O UNITS
F861   0CF9                 DW      BNPF            ; B - PUNCH BNPF
F863   49F9                 DW      COMP            ; C - COMPARE PROM WITH MEMORY
F865   94F9                 DW      DISP            ; D - DISPLAY RAM MEMORY
F867   C8F9                 DW      EOF             ; E - ENDFILE A HEXADECIMAL FILE
F869   E6F9                 DW      FILL            ; F - FILL MEMORY
F86B   F5F9                 DW      GOTO            ; G - GO TO MEMORY ADDRESS
F86D   3EFA                 DW      HEXM            ; H - HEXADECIMAL SUM AND DIFFERENCE
F86F   26F8                 DW      ERROR           ; I -
F871   26F8                 DW      ERROR           ; J -
F873   26F8                 DW      ERROR           ; K -
F875   59FA                 DW      LOAD            ; L - LOAD BNPF TAPE
F877   8EFA                 DW      MOVE            ; M - MOVE MEMORY
```

```
F879  9FFA            DW      NULL            ; N - PUNCH NULLS FOR LEADER
F87B  26F8            DW      ERROR           ; O -
F87D  B7FA            DW      PROG            ; P - PROGRAM A PROM
F87F  F5FA            DW      QUERY           ; Q - QUERY I/O SYSTEM STATUS
F881  33FB            DW      READ            ; R - READ HEXADECIMAL FILE
F883  93FB            DW      SUBS            ; S - SUBSTITUTE MEMORY
F885  B1FB            DW      TRAN            ; T - TRANSFER A PROM TO MEMORY
F887  26F8            DW      ERROR           ; U -
F889  26F8            DW      ERROR           ; V -
F88B  DDFB            DW      WRITE           ; W - WRITE HEX TAPE
F88D  22FC            DW      X               ; X - EXAMINE AND MODIFY REGISTERS
0018            LCT   EQU     ($-CTBL)/2
                ;
                ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                ;
                ; 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.
                ;
F88F            ASSIGN:
F88F  CDE8FF          CALL    TI              ; GET LOGICAL DEVICE CHARACTER
F892  21DCF8          LXI     H,LTBL          ; ADDRESS OF MASTER TABLE
F895  0E04            MVI     C,4             ; MAXIMUM OF 4 ENTRIES
F897            AS0:
F897  BE              CMP     M               ; TEST FOR IDENTIFYING CHARACTER
F898  23              INX     H
F899  CAA6F8          JZ      AS1             ; FOUND IT
F89C  23              INX     H               ; POINT TO NEXT ENTRY
F89D  23              INX     H
F89E  23              INX     H
F89F  0D              DCR     C               ; DECREMENT LOOP COUNT
F8A0  C297F8          JNZ     AS0             ; TRY NEXT ENTRY
F8A3  C326F8          JMP     ERROR           ; NO MATCH, ERROR
F8A6            AS1:
F8A6  46              MOV     B,M             ; GET SELECT BIT MASK
F8A7  23              INX     H
F8A8  5E              MOV     E,M             ; GET PHYSICAL DEVICE TABLE
F8A9  23              INX     H
F8AA  56              MOV     D,M
F8AB  EB              XCHG
    1           +     UNTIL   ',,,='
F8AC 1          +LOOP:
F8AC 1 CDE8FF +       CALL    TI
F8AF 1 FE3D   +       CPI     '='
F8B1 1 C2ACF8 +       JNZ     LOOP
    1           +     WHILE   ',,,'
F8B4 1          +LOOP:
F8B4 1 CDE8FF +       CALL    TI
F8B7 1 FE20   +       CPI     ' '
```

```
F8B9 1 CAB4F8  +       JZ      LOOP
F8BC   0E04            MVI     C,4             ; SET TABLE LENGTH
F8BE            AS2:                           ; INDEX THROUGH PHYSICAL UNIT TABLE
F8BE   BE              CMP     M               ; COMPARE DEVICE CHAR WITH LEGAL VALUES
F8BF   23              INX     H
F8C0   CACBF8          JZ      AS3             ; RETURN WITH HL -> DEVICE SELECT BITS
F8C3   23              INX     H
F8C4   0D              DCR     C
F8C5   C2BEF8          JNZ     AS2             ; CONTINUE LOOKUP
F8C8   C326F8          JMP     ERROR           ; ERROR RETURN
F8CB            AS3:
     1         +       UNTIL   CR
F8CB 1         +LOOP:
F8CB 1 CDE8FF  +       CALL    TI
F8CE 1 FE0D    +       CPI     CR
F8D0 1 C2CBF8  +       JNZ     LOOP
F8D3   3A0300          LDA     IOBYT           ; GET I/O STATUS
F8D6   A0              ANA     B               ; CLEAR FIELD
F8D7   B6              ORA     M               ; SET NEW STATUS
F8D8   320300          STA     IOBYT           ; RETURN TO MEMORY
F8DB   C9              RET
                ;
                ; MASTER I/O DEVICE TABLE
                ; 4 BYTES/ENTRY
                ;
                ;    BYTE 0 = IDENTIFYING CHARACTER
                ;    BYTE 1 = LOGICAL DEVICE MASK
                ;    BYTES 2,3 = SUBORDINATE PHYSICAL DEVICE TABLE
                ;
F8DC            LTBL:
F8DC   43FC            DB      'C',CMSK
F8DE   ECF8            DW      ACT
F8E0   52F3            DB      'R',RMSK
F8E2   F4F8            DW      ART
F8E4   50CF            DB      'P',PMSK
F8E6   FCF8            DW      APT
F8E8   4C3F            DB      'L',LMSK
F8EA   04F9            DW      ALT
                ;
                ; I/O SYSTEM PHYSICAL DEVICE TABLES
                ; 2 BYTES/ENTRY
                ;
                ;   BYTE 0 = IDENTIFYING CHARACTER
                ;   BYTE 1 = DEVICE SELECT BIT PATTERN
                ;
F8EC            ACT:
F8EC   5400            DB      'T',CTTY        ; CONSOLE = TTY
F8EE   4301            DB      'C',CCRT        ; CONSOLE = CRT
F8F0   4202            DB      'B',BATCH       ; BATCH MODE CONSOLE = READ,LIST
F8F2   3103            DB      '1',CUSE        ; USER DEFINED CONSOLE DEVICE
F8F4            ART:
```

```
F8F4  5400            DB      'T',RTTY        ; READER = TTY
F8F6  5004            DB      'P',RPTR        ; READER = PTR
F8F8  3108            DB      '1',RUSE1       ; USER DEFINED READER DEVICE 1
F8FA  320C            DB      '2',RUSE2       ; USER DEFINED READER DEVICE 2
F8FC            APT:
F8FC  5400            DB      'T',PTTY        ; PUNCH = TTY
F8FE  5010            DB      'P',PPTP        ; PUNCH = PTP
F900  3120            DB      '1',PUSE1       ; USER DEFINED PUNCH DEVICE 1
F902  3230            DB      '2',PUSE2       ; USER DEFINED PUNCH DEVICE 2
F904            ALT:
F904  5400            DB      'T',LTTY        ; LIST = TTY
F906  4340            DB      'C',LCRT        ; LIST = CRT
F908  4C80            DB      'L',LLPT        ; LIST = LPT
F90A  31C0            DB      '1',LUSE        ; USER DEFINED LIST DEVICE
                      ;
                      ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                      ;
                      ; PUNCH A BNPF TAPE.
                      ;
                      ; THIS ROUTINE EXPECTS TWO HEXADECIMAL PARAMETERS TO BE
                      ; ENTERED FROM THE CONSOLE 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.
                      ;
F90C            BNPF:
F90C  CD7FFE          CALL    EXPR            ; GET TWO ADDRESSES
F90F  CD40FE          CALL    CRLF
F912  CD12FE          CALL    IOCON           ; OPEN PUNCH FOR OUTPUT
F915  0B              DB      OPFLG
F916  CDC0FE          CALL    LEAD
F919  D1              POP     D               ; GET HIGH ADDRESS
F91A  E1              POP     H               ; GET LOW ADDRESS
F91B            BN0:
F91B  CD6BFD          CALL    POC
F91E  42              DB      'B'             ; PUNCH A 'B'
F91F  0608            MVI     B,8             ; 8 BIT COUNT
F921  7E              MOV     A,M             ; GET DATA
F922            BN1:
F922  07              RLC                     ; ROTATE TO SET CARRY
F923  F5              PUSH    PSW             ; SAVE INTERMEDIATE RESULT
F924  3E27            MVI     A,'N'/2         ; COMPUTE EITHER 'P' OR 'N'
                                              ; BASED ON THE FOLLOWING ALGORITHM:
F926  CE00            ACI     0               ; CHAR = 'N'/2 + CARRY
F928  87              ADD     A               ; CHAR = CHAR * 2
                                              ; CHAR = 'N' IF CARRY = 0
F929  4F              MOV     C,A             ; CHAR = 'P' IF CARRY = 1
F92A  CD6FFD          CALL    PO
F92D  F1              POP     PSW
F92E  05              DCR     B
```

```
F92F  C222F9          JNZ     BN1
F932  CD6BFD          CALL    POC
F935  46              DB      'F'
F936  CD6BFD          CALL    POC
F939  20              DB      ' '
F93A  CD92FE          CALL    HILO
F93D  DAABFA          JC      NUO             ; ALL DONE, PUNCH TRAILER AND RETURN
F940  7D              MOV     A,L
F941  E603            ANI     03H             ; PUNCH CR,LF ON MULTIPLE OF 4
F943  CC2EFF          CZ      PEOL
F946  C318F9          JMP     BNO
                      ;
                      ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                      ;
                      ; COMPARE PROM WITH MEMORY.
                      ;
                      ; THIS ROUTINE EXPECTS (1) AN OPTION CHARACTER (X,Y,Z)
                      ; WHICH SPECIFIES WHICH SOCKET OPTION IS SELECTED, AND
                      ; (2) TWO HEXADECIMAL PARAMETERS WHICH IT INTERPRETS
                      ; AS THE RAM MEMORY AREA WHICH IS TO BE COMPARED TO THE
                      ; CONTENTS OF THE PROM IN THE SELECTED PROGRAMMING SOCKET.
                      ; ALL DIFFERENCES BETWEEN THE PROM AND THE MEMORY AREA
                      ; WILL BE DISPLAYED ON THE CONSOLE IN THE FOLLOWING
                      ; FORMAT:
                      ;
                      ; <MEM ADDRESS> <MEM CONTENTS> <CORRESPONDING PROM CONTENTS>
                      ;
F949                  COMP:
F949  CDB6FF              CALL    SELECT      ; SELECT SOCKET
F94C  C5                  PUSH    B           ; SAVE PROM ADDRESS AND COMMAND
F94D  0E02                MVI     C,2
F94F  CD7FFE              CALL    EXPR        ; GET MEMORY AREA
F952  D1                  POP     D           ; HIGH ADDRESS
F953  E1                  POP     H           ; LOW ADDRESS
F954  C1                  POP     B           ; POP PROM ADDRESS
F955  C5                  PUSH    B           ; SAVE BIT MASK
F956  0E00                MVI     C,0         ; ZERO 8 LSB OF PROM ADDRESS
F958                  CM0:
F958  78                  MOV     A,B
F959  D3F1                OUT     PHI         ; OUTPUT COMMAND AND 4 MSB
F95B  79                  MOV     A,C
F95C  D3F2                OUT     PLO         ; OUTPUT 8 LSB
F95E  DBF0                IN      PDATA       ; READ PROM
F960  E5                  PUSH    H
F961  210400              LXI     H,MSK       ; FETCH T/F FLAG
F964  AE                  XRA     M
F965  E1                  POP     H
F966  AE                  XRA     M           ; COMPARE PORT WITH MEMORY
F967  E3                  XTHL                ; GET BIT MASK
F968  A5                  ANA     L
F969  E3                  XTHL
```

```
F96A  CA8BF9          JZ      CM1             ; OK
F96D  C5              PUSH    B               ; SAVE ADDRESS
F96E  CD40FE          CALL    CRLF            ; NEW LINE FOR ERROR DISPLAY
F971  CDA2FE          CALL    LADR            ; DISPLAY MEMORY ADDRESS
F974  CD2EFD          CALL    BLK
F977  7E              MOV     A,M             ; DISPLAY MEMORY CONTENTS
F978  CDAAFE          CALL    LBYTE
F97B  CD2EFD          CALL    BLK
F97E  DBF0            IN      PDATA
F980  CDAAFE          CALL    LBYTE           ; DISPLAY PROM CONTENTS
F983  DBF1            IN      PSTAT
F985  E602            ANI     PCOMP
F987  CA26F8          JZ      ERROR
F98A  C1              POP     B               ; RESTORE PROM ADDRESS
F98B            CM1:
F98B  03              INX     B               ; INCREMENT PROM ADDRESS
F98C  CD92FE          CALL    HILO            ; TEST FOR COMPLETION
F98F  D258F9          JNC     CM0
F992  C1              POP     B               ; FLUSH MASK OFF STACK
F993  C9              RET
                ;
                ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                ;
                ; DISPLAY MEMORY IN HEX ON LIST DEVICE.
                ;
                ; THIS ROUTINE EXPECTS TWO HEXADECIMAL PARAMETERS SPECIFYING
                ; THE BOUNDS OF A MEMORY AREA TO BE DISPLAYED ON THE
                ; LIST 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.
                ;
F994            DISP:
F994  CD7FFE          CALL    EXPR            ; GET TWO ADDRESSES
F997  D1              POP     D               ; GET HIGH ADDRESS
F998  E1              POP     H               ; GET LOW ADDRESS
F999  CD12FE          CALL    IOCON           ; OPEN LIST FOR OUTPUT
F99C  0C              DB      OLFLG
F99D            DI0:
F99D  CDB6FE          CALL    LCRLF           ; PRINT CR,LF
F9A0  CD49FE          CALL    DADR            ; PRINT MEMORY ADDRESS
F9A3            DI1:
F9A3  0E20            MVI     C,' '
F9A5  CD97FD          CALL    LOM             ; PRINT SPACE
F9A8  7E              MOV     A,M
F9A9  CD51FE          CALL    DBYTE           ; PRINT DATA
F9AC  CD92FE          CALL    HILO            ; TEST FOR COMPLETION
F9AF  DABBF9          JC      DI2             ; RETURN TO MAIN LOOP
F9B2  7D              MOV     A,L
```

```
F9B3   E60F          ANI     0FH             ; PRINT CR,LF,ADDRESS ON MULTIPLE OF 16
F9B5   C2A3F9        JNZ     DI1
F9B8   C39DF9        JMP     DI0
F9BB             DI2:
F9BB   CDB6FE        CALL    LCRLF           ; WRITE CR,LF
F9BE   CD12FE        CALL    IOCON           ; CLOSE LIST OUTPUT
F9C1   0F            DB      CLFLG
F9C2   0E00          MVI     C,0
F9C4   CD97FD        CALL    LOM             ; WRITE A NULL TO TRIGGER CLOSE
F9C7   C9            RET
                     ;
                     ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                     ;
                     ; END OF FILE COMMAND.
                     ;
                     ; THIS ROUTINE PRODUCES A TERMINATION RECORD WHICH PROPERLY
                     ; COMPLETES A HEXADECIMAL FILE CREATED BY 'W' COMMANDS.  IF
                     ; THE OPTIONAL HEXADECIMAL PARAMETER IS SPECIFIED, A START
                     ; RECORD IS PRODUCED. A SUBSEQUENT 'R' COMMAND WILL LOAD
                     ; THE FILE CREATED AND IF A START RECORD IS ENCOUNTERED,
                     ; THE ENTRY POINT ADDRESS READ WILL REPLACE THE STORED
                     ; VALUE OF THE PROGRAM COUNTER.
                     ;
F9C8             EOF:
F9C8   0D            DCR     C               ; GET ONE PARAMETER
F9C9   CD7FFE        CALL    EXPR
F9CC   CD6BFD        CALL    POC
F9CF   3A            DB      ':'             ; OUTPUT RECORD MARK
F9D0   AF            XRA     A               ; ZERO CHECKSUM
F9D1   57            MOV     D,A
F9D2   CD09FF        CALL    PBYTE           ; OUTPUT RECORD LENGTH
F9D5   E1            POP     H               ; RETRIEVE PARAMETER
F9D6   CD01FF        CALL    PADR            ; OUTPUT IT.
F9D9   3E01          MVI     A,1             ; RECORD TYPE
F9DB   CD09FF        CALL    PBYTE           ; OUTPUT IT.
F9DE   AF            XRA     A
F9DF   92            SUB     D               ; COMPUTE CHECKSUM
F9E0   CD09FF        CALL    PBYTE           ; OUTPUT IT.
F9E3   C3ABFA        JMP     NUO             ; 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.
                     ;
F9E6             FILL:
F9E6   0C            INR     C               ; GET 3 PARAMETERS
```

```
F9E7   CD7FFE          CALL    EXPR
F9EA   C1              POP     B               ; GET DATA IN C
F9EB   D1              POP     D               ; GET HIGH ADDRESS
F9EC   E1              POP     H               ; GET LOW ADDRESS
F9ED             FIO:
F9ED   71              MOV     M,C             ; STORE CONSTANT IN MEMORY
F9EE   CD92FE          CALL    HILO            ; TEST FOR COMPLETION
F9F1   D2EDF9          JNC     FIO             ; CONTINUE LOOPING
F9F4   C9              RET
                ;
                ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                ;
                ; 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.
                ;
F9F5            GOTO:
    1           +       GET     EXIT            ; POINT RETURN ADDRESS AT USER CODE
F9F5 1 2A0400   +       LHLD    MEMTOP
F9F8 1 2ED2     +       MVI     L,EXIT AND 0FFH
F9FA   E3               XTHL                    ; REPLACE RETURN TO COMMAND LOOP
F9FB   CD1FFF           CALL    PCHK            ; GET A CHARACTER, SET Z,C
F9FE   CA0DFA           JZ      GO0             ; DON'T MODIFY PC
FA01   CDD1FE           CALL    PA0             ; GET NEW PC VALUE
FA04   EB               XCHG                    ; DE = NEW PC
    1           +       GET     PLOC
FA05 1 2A0400   +       LHLD    MEMTOP
FA08 1 2EE1     +       MVI     L,PLOC AND 0FFH
FA0A   72               MOV     M,D             ; STORE MODIFIED PC IN RAM
FA0B   2B               DCX     H
FA0C   73               MOV     M,E
FA0D            GO0:
FA0D   DA3AFA           JC      GO4             ; NO TRAPS TO BE SET
FA10   110200           LXI     D,2             ; SET COUNTER(S), D=0, E=2
FA13            GO1:
FA13   CD64FD           CALL    COMC            ; ISSUE A PROMPT FOR A TRAP
```

```
FA16   2D              DB      '-'
FA17   CDCBFE          CALL    PARAM           ; GET A TRAP
FA1A   E5              PUSH    H               ; STACK IT
FA1B   14              INR     D               ; UP 1 COUNTER
FA1C   DA23FA          JC      GO2             ; TERMINATE IF CR ENTERED
FA1F   1D              DCR     E               ; DOWN THE OTHER
FA20   C213FA          JNZ     GO1             ; GET ONE MORE TRAP
FA23                GO2:
FA23   D226F8          JNC     ERROR           ; LAST TRAP NOT FOLLOWED BY CR
     1            +     GET     TLOC            ; LOCATE TRAP AREA
FA26 1 2A0400     +     LHLD    MEMTOP
FA29 1 2EE2       +     MVI     L,TLOC AND 0FFH
FA2B                GO3:
FA2B   C1              POP     B               ; GET ONE TRAP
FA2C   71              MOV     M,C             ; STORE TRAP ADDRESS IN TRAP AREA.
FA2D   23              INX     H
FA2E   70              MOV     M,B
FA2F   23              INX     H
FA30   0A              LDAX    B               ; FETCH OPCODE BYTE
FA31   77              MOV     M,A             ; PUT IN TRAP AREA
FA32   23              INX     H
FA33   3EC7            MVI     A,(RST 0)       ; SET TRAP OPCODE
FA35   02              STAX    B
FA36   15              DCR     D
FA37   C22BFA          JNZ     GO3
FA3A                GO4:
FA3A   CD40FE          CALL    CRLF
FA3D   C9              RET                     ; RETURN TO USER CODE
                    ;
                    ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                    ;
                    ; 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>
                    ;
FA3E                HEXN:
FA3E   CD7FFE          CALL    EXPR            ; GET TWO NUMBERS
FA41   CD40FE          CALL    CRLF
FA44   D1              POP     D
FA45   E1              POP     H
FA46   E5              PUSH    H
FA47   19              DAD     D               ; COMPUTE HL+DE
FA48   CDA2FE          CALL    LADR            ; DISPLAY SUM
FA4B   CD2EFD          CALL    BLK             ; TYPE A SPACE
FA4E   E1              POP     H
FA4F   7D              MOV     A,L             ; COMPUTE HL-DE
FA50   93              SUB     E
```

```
FA51  6F            MOV     L,A
FA52  7C            MOV     A,H
FA53  9A            SBB     D
FA54  67            MOV     H,A
FA55  CDA2FE        CALL    LADR            ; DISPLAY DIFFERENCE
FA58  C9            RET
                ;
                ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                ;
                ; 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.
                ;
FA59            LOAD:
FA59  CD7FFE        CALL    EXPR            ; GET TWO ADDRESSES
FA5C  D1            POP     D               ; GET HIGH ADDRESS
FA5D  E1            POP     H               ; GET LOW ADDRESS
FA5E  CD12FE        CALL    IOCON
FA61  07            DB      ORFLG
FA62            LOO:
FA62  CDADFF        CALL    RIX             ; READ TAPE
FA65  FE42          CPI     'B'             ; SCAN FOR 'B'
FA67  C262FA        JNZ     LOO
FA6A  3601          MVI     M,1             ; INITIALIZE MEMORY
FA6C            LO1:
FA6C  CDADFF        CALL    RIX             ; GET DATA
FA6F  FE4E          CPI     'N'             ; CHECK FOR 'N'
FA71  CA79FA        JZ      LO2             ; FOUND IT, CARRY = 0
FA74  C6B0          ADI     -'P'
FA76  C226F8        JNZ     ERROR           ; ERROR
                                            ; CARRY IS SET
FA79            LO2:
FA79  7E            MOV     A,M             ; SHIFT IN DATA BIT
FA7A  17            RAL
FA7B  77            MOV     M,A
FA7C  D26CFA        JNC     LO1             ; IF CARRY IS SET, 8 BITS READ
FA7F  CDADFF        CALL    RIX             ; TEST FOR REQ'D 'F'
FA82  FE46          CPI     'F'
FA84  C226F8        JNZ     ERROR
FA87  CD92FE        CALL    HILO            ; TEST FOR COMPLETION
FA8A  D262FA        JNC     LOO             ; KEEP GOING
FA8D  C9            RET
                ;
```

```
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ; 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.
                        ;
FA8E                    MOVE:
FA8E   0C                       INR     C               ; GET THREE ADDRESSES
FA8F   CD7FFE                   CALL    EXPR
FA92   C1                       POP     B               ; DESTINATION
FA93   D1                       POP     D               ; SOURCE END
FA94   E1                       POP     H               ; SOURCE BEGIN
FA95                    MV0:
FA95   7E                       MOV     A,M             ; GET A DATA BYTE
FA96   02                       STAX    B               ; STORE AT DESTINATION
FA97   03                       INX     B               ; MOVE DESTINATION POINTER
FA98   CD92FE                   CALL    HILO            ; TEST FOR COMPLETION
FA9B   D295FA                   JNC     MV0
FA9E   C9                       RET
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ; 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.
                        ;
FA9F                    NULL:
FA9F   CDE8FF                   CALL    TI              ; REQUIRE CR
FAA2   FE0D                     CPI     CR
FAA4   C226F8                   JNZ     ERROR
FAA7   CD12FE                   CALL    IOCON
FAAA   0B                       DB      OPFLG
FAAB                    NU0:
FAAB   CDC0FE                   CALL    LEAD
FAAE   CD12FE                   CALL    IOCON
FAB1   0E                       DB      CPFLG
FAB2   CD6BFD                   CALL    POC
FAB5   00                       DB      0
FAB6   C9                       RET
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ; DRIVER PROGRAM FOR INTELLEC 80 PROM PROGRAMMING PERIPHERAL.
                        ;
                        ; THIS ROUTINE EXPECTS (1) AN OPTION CHARACTER (X,Y,Z)
                        ; WHICH SPECIFIES WHICH SOCKET OPTION IS SELECTED, (2) A
```

```
                        ; TRUE/FALSE CHARACTER (T OR F) WHICH SPECIFIES IF A DATA
                        ; INVERSION IS TO TAKE PLACE, AND (3) THREE HEXADECIMAL
                        ; PARAMETERS, THE FIRST TWO OF WHICH ARE INTERPRETED AS THE
                        ; MEMORY AREA TO BE USED AS THE PROGRAMMING SOURCE DATA,
                        ; THE THIRD ADDRESS IS THE FIRST ADDRESS IN THE PROM. NO
                        ; TIMING IS PERFORMED BY THIS ROUTINE, ALL PROM DEPENDENT
                        ; TIMING IS HANDLED BY THE PERIPHERAL.  UPON RECEIPT OF AN
                        ; ERROR STATUS FROM THE DEVICE, THIS ROUTINE DISPLAYS THE
                        ; ADDRESS WHICH FAILED TO PROGRAM AND TERMINATES WITH AN
                        ; ERROR CONDITION.
                        ;
FAB7                    PROG:
FAB7    CDB6FF                CALL    SELECT          ; SELECT SOCKET
FABA    C5                    PUSH    B               ; SAVE PROM OPCODE
FABB    0E03                  MVI     C,3
FABD    CD7FFE                CALL    EXPR            ; GET THREE PARAMETERS
FAC0    CD40FE                CALL    CRLF
FAC3    C1                    POP     B               ; PROM ADDRESS
FAC4    D1                    POP     D               ; HIGH ADDRESS
FAC5    E1                    POP     H               ; LOW ADDRESS
FAC6    3E0F                  MVI     A,0FH
FAC8    A0                    ANA     B               ; MASK 4 MSB OF B
FAC9    47                    MOV     B,A
FACA    F1                    POP     PSW             ; OPCODE IS IN A
FACB    B0                    ORA     B
FACC    47                    MOV     B,A             ; BC CONTAINS PROM OPCODE AND ADDRESS
FACD                    PR0:
FACD    78                    MOV     A,B             ; OUTPUT PROM ADDRESS AND OPCODE
FACE    D3F1                  OUT     PHI
FAD0    79                    MOV     A,C
FAD1    D3F2                  OUT     PLO
FAD3    03                    INX     B               ; INCREMENT PROM ADDRESS
FAD4    3A0400                LDA     MSK             ; FETCH T/F FLAG
FAD7    AE                    XRA     M               ; XOR DATA FROM MEMORY
FAD8    D3F0                  OUT     PDATA           ; OUTPUT PROM DATA
FADA                    PR1:
FADA    DBF1                  IN      PSTAT
FADC    E601                  ANI     PGRDY           ; TEST FOR COMPLETION
FADE    C2DAFA                JNZ     PR1
FAE1    DBF1                  IN      PSTAT
FAE3    E602                  ANI     PCOMP
FAE5    CAEFFA                JZ      PR2             ; PROGRAMMING ERROR, ABORT
FAE8    CD92FE                CALL    HILO            ; TEST FOR PROGRAMMING COMPLETION
FAEB    D2CDFA                JNC     PR0
FAEE    C9                    RET                     ; RETURN TO COMMAND MODE
FAEF                    PR2:
FAEF    CDA2FE                CALL    LADR            ; DISPLAY BAD ADDRESS
FAF2    C326F8                JMP     ERROR
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
```

```
                        ; SYSTEM QUERY COMMAND.
                        ;
                        ; THIS COMMAND IS INVOKED BY TYPING THE LETTER Q.  THIS
                        ; COMMAND PRODUCES A LISTING OF LOGICAL I/O DEVICES AND
                        ; THEIR CORRESPONDING PHYSICAL DEVICE ASSIGNMENTS.  THE
                        ; DATA DISPLAYED IS EQUIVALENT TO THE CURRENT VALUE OF IOBYT.
                        ;
FAF5                    QUERY:
FAF5    CDE8FF              CALL    TI              ; REQUIRE CR
FAF8    FE0D                CPI     CR
FAFA    C226F8              JNZ     ERROR
FAFD    0604                MVI     B,4             ; SET UP OUTER LOOP COUNTER.
FAFF    21DCF8              LXI     H,LTBL          ; POINT HL AT LOGICAL DEVICE TABLE.
FB02                    Q0:
FB02    CD40FE              CALL    CRLF            ; START A NEW LINE.
FB05    4E                  MOV     C,M             ; DISPLAY LOGICAL DEVICE IDENTIFIER.
FB06    CD30FD              CALL    COM
FB09    CD64FD              CALL    COMC            ; DISPLAY '='.
FB0C    3D                  DB      '='
FB0D    23                  INX     H               ; POINT AT MASK FOR LOGICAL DEVICE.
FB0E    7E                  MOV     A,M             ; FETCH MASK.
FB0F    2F                  CMA                     ; INVERT IT
FB10    4F                  MOV     C,A             ; PUT IN C
FB11    23                  INX     H               ; POINT AT PHYSICAL DEVICE TABLE
FB12    5E                  MOV     E,M             ; ADDRESS OF SUBORDINATE
FB13    23                  INX     H               ; TABLE
FB14    56                  MOV     D,M
FB15    23                  INX     H
FB16    EB                  XCHG                    ; HL <- PHYSICAL DEVICE TABLE
FB17    3A0300              LDA     IOBYT
FB1A    A1                  ANA     C               ; PHYSICAL SELECTION
FB1B    C5                  PUSH    B               ; SAVE OUTER LOOP COUNTER
FB1C    0604                MVI     B,4             ; SET UP INNER LOOP COUNTER
FB1E                    Q1:
FB1E    4E                  MOV     C,M             ; GET PHYSICAL DEVICE IDENTIFIER
FB1F    23                  INX     H
FB20    BE                  CMP     M               ; TEST FOR EQUALITY
FB21    CA29FB              JZ      Q2
FB24    23                  INX     H               ; POINT AT NEXT ENTRY
FB25    05                  DCR     B               ; DECREMENT INNER LOOP
FB26    C21EFB              JNZ     Q1
FB29                    Q2:
FB29    CD30FD              CALL    COM             ; DISPLAY PHYSICAL DEVICE
FB2C    EB                  XCHG                    ; POINT AT MASTER TABLE
FB2D    C1                  POP     B
FB2E    05                  DCR     B               ; DECREMENT OUTER LOOP
FB2F    C202FB              JNZ     Q0
FB32    C9                  RET
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
```

```
                        ; 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 BASE 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 AN EOF RECORD IS ENCOUNTERED. THE LOAD
                        ; ADDRESS FIELD REPLACES THE USER'S STORED PC VALUE.
                        ; A TRANSFER TO THE PROGRAM MAY THEN BE ACCOMPLISHED BY A 'G<CR>'.
                        ;
FB33                    READ:
FB33   0D                       DCR     C               ; GET ONE ADDRESS
FB34   CD7FFE                   CALL    EXPR
FB37   CD40FE                   CALL    CRLF
FB3A   CD12FE                   CALL    IOCON
FB3D   07               DB      ORFLG
FB3E                    RED0:
FB3E   CDADFF           RED0:   CALL    RIX
FB41   FE3A             RED0:   CPI     ':'
FB43   C23EFB           RED0:   JNZ     RED0            ; SCAN TO RECORD MARK
FB46   AF               RED0:   XRA     A
FB47   57               RED0:   MOV     D,A             ; CLEAR CHECKSUM
FB48   CD1DFE           RED0:   CALL    BYTE
FB4B   CA83FB           RED0:   JZ      RED3            ; ZERO RECORD LENGTH, ALL DONE
FB4E   5F               RED0:   MOV     E,A             ; E <- RECORD LENGTH
FB4F   CD1DFE           RED0:   CALL    BYTE            ; GET MSB OF LOAD ADDRESS
FB52   67               RED0:   MOV     H,A
FB53   CD1DFE           RED0:   CALL    BYTE            ; GET LSB OF LOAD ADDRESS
FB56   6F               RED0:   MOV     L,A
FB57   CD1DFE           RED0:   CALL    BYTE            ; GET RECORD TYPE
FB5A   4B               RED0:   MOV     C,E             ; C <- RECORD LENGTH
FB5B   E5               RED0:   PUSH    H
     1                  RED0:+  FETCH   -256            ; COMPUTE BUFFER POINTER
FB5C 1 2100FF           RED0:+  LXI     H,0FF00H
FB5F 1 39               RED0:+  DAD     SP
FB60                    RED1:
FB60   CD1DFE           RED1:   CALL    BYTE            ; READ DATA
FB63   77               RED1:   MOV     M,A             ; PUT IN BUFFER
FB64   23               RED1:   INX     H
FB65   1D               RED1:   DCR     E
FB66   C260FB           RED1:   JNZ     RED1            ; LOOP UNTIL DONE
FB69   CD1DFE           RED1:   CALL    BYTE            ; READ CHECKSUM
FB6C   C226F8           RED1:   JNZ     ERROR           ; CHECKSUM ERROR
FB6F   D1               RED1:   POP     D               ; DE = LOAD ADDRESS
FB70   E3               RED1:   XTHL                    ; HL = BIAS ADDRESS, TOS = BUFFER
FB71   EB               RED1:   XCHG                    ; DE = BIAS ADDRESS, HL = LOAD ADDRESS
```

```
FB72   19              DAD     D               ; HL = BIAS + LA
FB73   0600            MVI     B,0
FB75   09              DAD     B               ; HL = BIAS + LA + RL
FB76   EB              XCHG                    ; DE = BIAS + LA + RL, HL = BIAS
FB77   E3              XTHL                    ; HL = BUFFER, TOS = BIAS
FB78                RED2:
FB78   2B              DCX     H               ; DECREMENT BUFFER POINTER
FB79   7E              MOV     A,M
FB7A   1B              DCX     D               ; DECREMENT MEMORY POINTER
FB7B   12              STAX    D               ; PUT DATA IN DESIGNATED ADDRESS
FB7C   0D              DCR     C               ; UNTIL COUNT IS EXHAUSTED
FB7D   C278FB          JNZ     RED2
FB80   C33EFB          JMP     RED0            ; GET ANOTHER RECORD
FB83                RED3:
     1            +     GET     PLOC            ; POINT HL AT PC UPPER
FB83 1 2A0400     +     LHLD    MEMTOP
FB86 1 2EE1       +     MVI     L,PLOC AND 0FFH
FB88   CD1DFE          CALL    BYTE            ; GET MSB OF TRANSFER ADDRESS
FB8B   77              MOV     M,A
FB8C   2B              DCX     H               ; POINT HL AT PC LOWER
FB8D   CD1DFE          CALL    BYTE
FB90   77              MOV     M,A
FB91   E1              POP     H
FB92   C9              RET
                    ;
                    ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                    ;
                    ; 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.
                    ;
FB93                SUBS:
FB93   CDCBFE          CALL    PARAM           ; GET MEMORY ADDRESS
FB96   D8              RC                      ; CR ENTERED, RETURN TO COMMAND MODE
FB97                SU0:
FB97   7E              MOV     A,M
FB98   CDAAFE          CALL    LBYTE           ; DISPLAY DATA
FB9B   CD64FD          CALL    COMC
FB9E   2D '            DB      '-'
FB9F   CD1FFF          CALL    PCHK
FBA2   D8              RC                      ; CR ENTERED, RETURN TO COMMAND MODE
FBA3   CAADFB          JZ      SU1             ; SPACE ENTERED, SPACE BY
FBA6   EB              XCHG                    ; SAVE MEMORY ADDRESS
```

```
FBA7   CDD1FE          CALL    PA0             ; GET NEW VALUE
FBAA   EB              XCHG                    ; E = VALUE
FBAB   73              MOV     M,E             ; STORE NEW VALUE
FBAC   D8              RC                      ; CR ENTERED AFTER VALUE, RETURN
FBAD                SU1:
FBAD   23              INX     H
FBAE   C397FB          JMP     SU0
                       ;
                       ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                       ;
                       ; TRANSFER CONTENTS OF A PROM TO MEMORY.
                       ;
                       ; THIS ROUTINE EXPECTS (1) AN OPTION CHARACTER (X,Y,Z)
                       ; WHICH SPECIFIES WHICH SOCKET OPTION IS SELECTED, AND
                       ; (2) TWO HEXADECIMAL PARAMETERS WHICH IT INTERPRETS
                       ; AS THE RAM MEMORY AREA WHERE THE CONTENTS OF THE
                       ; PROM IN THE SELECTED PROGRAMMING SOCKET IS TO BE STORED.
                       ;
FBB1                TRAN:
FBB1   CDB6FF          CALL    SELECT          ; SELECT SOCKET
FBB4   0E00            MVI     C,0             ; BC = PROM ADDRESS + COMMANDS
FBB6   C5              PUSH    B               ; STACK PROM ADDRESS
FBB7   0E02            MVI     C,2
FBB9   CD7FFE          CALL    EXPR            ; GET MEMORY AREA
FBBC   D1              POP     D               ; HIGH ADDRESS
FBBD   E1              POP     H               ; LOW ADDRESS
FBBE   C1              POP     B               ; PROM ADDRESS
FBBF                TR0:
FBBF   78              MOV     A,B
FBC0   D3F1            OUT     PHI             ; OUTPUT COMMAND AND 4 MSB
FBC2   79              MOV     A,C
FBC3   D3F2            OUT     PLO             ; OUTPUT 8 LSB OF ADDRESS
FBC5   03              INX     B
FBC6   DBF0            IN      PDATA
FBC8   E5              PUSH    H
FBC9   210400          LXI     H,MSK           ; FETCH I/F FLAG
FBCC   AE              XRA     M
FBCD   E1              POP     H
FBCE   77              MOV     M,A             ; PUT DATA IN RAM
FBCF   DBF1            IN      PSTAT
FBD1   E602            ANI     PCOMP
FBD3   CA26F8          JZ      ERROR
FBD6   CD92FE          CALL    HILO
FBD9   D2BFFB          JNC     TR0             ; CONTINUE LOOPING
FBDC   C9              RET
                       ;
                       ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                       ;
                       ; 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.
                        ;
FBDD                    WRITE:
FBDD    CD7FFE                  CALL    EXPR            ; GET ADDRESS RANGE
FBE0    D1                      POP     D               ; HIGH ADDRESS
FBE1    E1                      POP     H               ; LOW ADDRESS
FBE2    CD12FE                  CALL    IOCON
FBE5    0B                      DB      OPFLG
FBE6                    WR0:
FBE6    CD6BFD                  CALL    PUC             ; EMIT RECORD MARK
FBE9    3A                      DB      ':'
FBEA    011000                  LXI     B,16            ; INITIALIZE B=0, C=16
FBED    E5                      PUSH    H               ; SAVE HL
FBEE                    WR1:
FBEE    04                      INR     B
FBEF    0D                      DCR     C
FBF0    CAF9FB                  JZ      WR2             ; TERMINATE ON COUNT OF 16 BYTES
FBF3    CD92FE                  CALL    HILO            ; OR END OF RANGE
FBF6    D2EEFB                  JNC     WR1             ; WHICHEVER OCCURS FIRST
FBF9                    WR2:
FBF9    E1                      POP     H               ; RESTORE LOW ADDRESS
FBFA    D5                      PUSH    D               ; SAVE HIGH ADDRESS
FBFB    1600                    MVI     D,0             ; INITIALIZE CHECKSUM
FBFD    78                      MOV     A,B             ; PUT RECORD LENGTH IN A
FBFE    CD09FF                  CALL    PBYTE           ; EMIT RECORD LENGTH
FC01    CD01FF                  CALL    PADR            ; EMIT LOW ADDRESS
FC04    AF                      XRA     A               ; EMIT RECORD TYPE
FC05    CD09FF                  CALL    PBYTE
FC08                    WR3:
FC08    7E                      MOV     A,M             ; FETCH CODE
FC09    CD09FF                  CALL    PBYTE           ; EMIT IT
FC0C    23                      INX     H               ; INCREMENT MEMORY ADDRESS
FC0D    05                      DCR     B               ; DECREMENT COUNT
FC0E    C208FC                  JNZ     WR3
FC11    AF                      XRA     A
FC12    92                      SUB     D               ; PUNCH -CHECKSUM
FC13    CD09FF                  CALL    PBYTE
FC16    D1                      POP     D               ; RESTORE HIGH ADDRESS
FC17    2B                      DCX     H               ; BACKUP MEMORY POINTER
FC18    CD2EFF                  CALL    PEOL            ; PUNCH CR,LF
FC1B    CD92FE                  CALL    HILO            ; TEST FOR TERMINATION
FC1E    D2E6FB                  JNC     WR0
FC21    C9                      RET
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ; 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.
                        ;
FC22                    X:
FC22  217DFC                LXI     H,ACTBL         ; POINT TO ACCESS TABLE
FC25  CD1FFF                CALL    PCHK            ; GET REGISTER IDENTIFIER
FC28  DA66FC                JC      X5              ; IF CARRY = 1, CR ENTERED
FC2B  0E0C                  MVI     C,NREGS
FC2D                    X0:
FC2D  BE                    CMP     M
FC2E  CA3BFC                JZ      X1              ; MATCHED REGISTER IDENTIFIER
FC31  23                    INX     H               ; POINT TO NEXT TABLE ENTRY
FC32  23                    INX     H
FC33  23                    INX     H
FC34  0D                    DCR     C               ; DECREMENT REGISTER COUNTER
FC35  C22DFC                JNZ     X0              ; TRY AGAIN
FC38  C326F8                JMP     ERROR           ; NOT IN TABLE, ERROR
FC3B                    X1:
FC3B  CD2EFD                CALL    BLK
FC3E                    X2:
FC3E  CD6BFE                CALL    DREG            ; DISPLAY THE REGISTER
FC41  CD64FD                CALL    COMC
FC44  2D                    DB      '-'             ; TYPE PROMPT
FC45  CD1FFF                CALL    PCHK            ; SKIP IF NULL ENTRY
FC48  D8                    RC                      ; CR ENTERED, RETURN TO COMMAND MODE
FC49  CA5CFC                JZ      X4
FC4C  E5                    PUSH    H               ; SAVE POINTER TO ACTBL
FC4D  C5                    PUSH    B               ; SAVE PRECISION
FC4E  CDD1FE                CALL    PA0             ; GET NEW REG VALUE
FC51  7D                    MOV     A,L
FC52  12                    STAX    D               ; STORE LSB IN REGISTER AREA
FC53  F1                    POP     PSW             ; RETRIEVE PRECISION (A)
FC54  B7                    ORA     A               ; SET SIGN
FC55  FA5BFC                JM      X3              ; 8 BITS ONLY
FC58  13                    INX     D
FC59  7C                    MOV     A,H
FC5A  12                    STAX    D               ; STORE MSB IN REGISTER AREA
FC5B                    X3:
FC5B  E1                    POP     H               ; RETRIEVE ACTBL POINTER
FC5C                    X4:
FC5C  AF                    XRA     A
FC5D  B6                    ORA     M
FC5E  F8                    RM                      ; END OF TABLE, RETURN TO COMMAND MODE
FC5F  78                    MOV     A,B             ; TEST DELIMITER
FC60  FE0D                  CPI     CR
FC62  C8                    RZ                      ; CR ENTERED, RETURN TO COMMAND MODE
FC63  C33EFC                JMP     X2
FC66                    X5:                         ; FULL REGISTER DISPLAY
FC66  CD40FE                CALL    CRLF
```

```
FC69                X6:
FC69   CD2EFD              CALL    BLK             ; OUTPUT A SPACE
FC6C   AF                  XRA     A               ; CLEAR A
FC6D   B6                  ORA     M               ; SET CONDITION CODES
FC6E   F8                  RM                      ; ALL DONE, RETURN TO COMMAND MODE
FC6F   4E                  MOV     C,M
FC70   CD30FD              CALL    COM             ; PRINT CHARACTER
FC73   CD64FD              CALL    COMC            ; PRINT EQUAL SIGN
FC76   3D                  DB      '='
FC77   CD6BFE              CALL    DREG            ; DISPLAY REGISTER CONTENTS
FC7A   C369FC              JMP     X6              ; CONTINUE
                    ;
                    ; TABLE FOR ACCESSING REGISTERS
                    ; TABLE CONTAINS:
                    ;     (1) REGISTER IDENTIFIER
                    ;     (2) LOCATION ON STORAGE PAGE
                    ;     (3) PRECISION
                    ;
FC7D                ACTBL:
FC7D   41CF00              DB      'A',    ALOC,   0
FC80   42CB00              DB      'B',    BLOC,   0
FC83   43CA00              DB      'C',    CLOC,   0
FC86   44C900              DB      'D',    DLOC,   0
FC89   45C800              DB      'E',    ELOC,   0
FC8C   46CE00              DB      'F',    FLOC,   0
FC8F   48DD00              DB      'H',    HLOC,   0
FC92   49CD00              DB      'I',    ILOC,   0
FC95   4CDC00              DB      'L',    LLOC,   0
FC98   4DDD01              DB      'M',    HLOC,   1
FC9B   50E101              DB      'P',    PLOC,   1
FC9E   53D101              DB      'S',    SLOC,   1
FCA1   FF                  DB      -1
000C                NREGS EQU     ($-ACTBL)/3     ; LENGTH OF ACCESS TABLE
                    ;
                    ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                    ;
                    ; END OF MONITOR COMMANDS, BEGINNING OF I/O SUBROUTINES
                    ;
                    ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                    ;
                    ; EXTERNALLY REFERENCED ROUTINE
                    ; CONSOLE INPUT CODE, VALUE RETURNED IN A
                    ; A, FLAGS MODIFIED
                    ; STACK USAGE: 2 BYTES
                    ;
FCA2                CI:                             ; CONSOLE INPUT
FCA2   3A0300              LDA     IOBYT           ; GET STATUS BYTE
FCA5   E603                ANI     NOT CMSK        ; GET CONSOLE BITS
FCA7   C2B4FC              JNZ     CIO             ; TEST FOR CRT
FCAA                TTYIN:
FCAA   DBF5                IN      TTS             ; TTY STATUS PORT
```

```
FCAC   E602          ANI     RBR             ; CHECK FOR RECEIVE BUFFER READY
FCAE   CAAAFC        JZ      TTYIN
FCB1   DBF4          IN      TTI             ; READ THE CHARACTER
FCB3   C9            RET                     ; RETURN
FCB4             CI0:
FCB4   FE01          CPI     CCRT            ; CONSOLE = CRT?
FCB6   C2C3FC        JNZ     CI1             ; TEST FOR BATCH
FCB9             CRTIN:
FCB9   DBF7          IN      CRTS            ; CRT STATUS PORT
FCBB   E602          ANI     RBR             ; CHECK FOR RECEIVE BUFFER READY
FCBD   CAB9FC        JZ      CRTIN           ; NOT READY, CONTINUE LOOPING
FCC0   DBF6          IN      CRTI            ; READ THE CHARACTER
FCC2   C9            RET
FCC3             CI1:
FCC3   FE02          CPI     BATCH
FCC5   CAD5FC        JZ      RI              ; BATCH MODE, INPUT = READER
FCC8   3EE8          MVI     A,CILOC         ; USER DEFINE CONSOLE INPUT
FCCA   C327FD        JMP     @USER
                     ;
                     ; TEST FOR OPERATOR INTERRUPTION OF COMMAND
                     ; BY DEPRESSING BREAK KEY
                     ;
                     ; REGISTER USAGE
                     ;
                     ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                     ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                     ; U = USED AS INPUT.
                     ;
                     ;     A - X
                     ;     B -             C -
                     ;     D -             E -
                     ;     H -             L -
                     ;     CARRY - X       ZERO - X
                     ;     SIGN - X        PARITY - X
                     ;     SP -            PC -
                     ;     STACK USAGE: 4 BYTES
                     ;
FCCD             BREAK:
FCCD   CDC1FD        CALL    CSTS            ; SEE IF A KEY WAS DEPRESSED
FCD0   B7            ORA     A
FCD1   C8            RZ                      ; NO CHARACTER READY
FCD2   C3E8FF        JMP     TI              ; GET THE CHARACTER
                     ;
                     ; EXTERNALLY REFERENCED ROUTINE
                     ; READER INPUT CODE
                     ; VALUE RETURNED IN A, FLAGS MODIFIED
                     ; STACK USAGE: 8 BYTES
                     ;
FCD5             RI:                         ; READER INPUT
FCD5   E5            PUSH    H               ; SAVE HL
FCD6   210300        LXI     H,IOBYT         ; POINT HL AT IOBYT
```

```
FCD9  7E              MOV     A,M
FCDA  E60C            ANI     NOT RMSK        ; READER = PTR?
FCDC  C2FCFC          JNZ     RI3             ; BRANCH TO PTR ROUTINE
FCDF  3E02            MVI     A,TTYADV        ; READER = TTY
FCE1  D3F9            OUT     PTRC
FCE3  26FA            MVI     H,TOUT          ; SET READER TIMEOUT TIMER
FCE5            RI0:
FCE5  DBF5            IN      TTS
FCE7  E602            ANI     RBR
FCE9  C2F7FC          JNZ     RI2             ; DATA IS READY
FCEC  CD63FE          CALL    DELAY           ; DELAY 10 MS
FCEF  25              DCR     H
FCF0  C2E5FC          JNZ     RI0
FCF3            RI1:
FCF3  AF              XRA     A
FCF4  37              STC                     ; SET CARRY INDICATING EOF
FCF5  E1              POP     H
FCF6  C9              RET                     ; RETURN
FCF7            RI2:
FCF7  DBF4            IN      TTI
FCF9  B7              ORA     A               ; CLEAR CARRY
FCFA  E1              POP     H
FCFB  C9              RET                     ; RETURN
FCFC            RI3:                          ; PTR ROUTINE
FCFC  FE04            CPI     RPTR
FCFE  C21DFD          JNZ     RI6
FD01  3E08            MVI     A,PTRADV        ; START PTR
FD03  D3F9            OUT     PTRC
FD05  26FA            MVI     H,TOUT          ; SET READER TIMEOUT TIMER
FD07            RI4:
FD07  DBF9            IN      PTRS
FD09  E601            ANI     PTRDY
FD0B  C218FD          JNZ     RI5
FD0E  CD63FE          CALL    DELAY
FD11  25              DCR     H
FD12  C207FD          JNZ     RI4
FD15  C3F3FC          JMP     RI1
FD18            RI5:
FD18  DBF8            IN      PTRI            ; GET THE DATA
FD1A  B7              ORA     A
FD1B  E1              POP     H
FD1C  C9              RET                     ; RETURN
FD1D            RI6:
FD1D  E1              POP     H
FD1E  FE08            CPI     RUSE1
FD20  3EEE            MVI     A,R1LOC
FD22  CA27FD          JZ      @USER
FD25  3EF1            MVI     A,R2LOC
                ;
                ; USER DEFINED I/O ENTRY POINT TRANSFER LOGIC
                ;
```

```
FD27                @USER:
FD27    E5                  PUSH    H               ; SAVE HL, CREATE A STACK ENTRY
FD28    2A0400              LHLD    MEMTOP
FD2B    6F                  MOV     L,A
FD2C    E3                  XTHL
FD2D    C9                  RET
                    ;
                    ; EXTERNALLY REFERENCED ROUTINE
                    ; CONSOLE OUTPUT CODE, VALUE EXPECTED IN C
                    ; A,FLAGS,C MODIFIED
                    ; STACK USAGE: 2 BYTES
                    ;
FD2E                BLK:                            ; PRINT A BLANK
FD2E    0E20                MVI     C,' '
FD30                COM:                            ; CONSOLE OUTPUT
FD30    3A0300              LDA     IOBYT
FD33    E603                ANI     NOT CMSK
FD35    FE02                CPI     BATCH           ; DON'T HONOR BREAK KEY IN BATCH MODE
FD37    C4CDFC              CNZ     BREAK           ; TEST FOR BREAK KEY
FD3A                CO:                             ; EXTERNAL ENTRY POINT
FD3A    3A0300              LDA     IOBYT           ; GET STATUS BYTE
FD3D    E603                ANI     NOT CMSK        ; GET CONSOLE BITS
FD3F    C24DFD              JNZ     COO             ; TEST FOR CRT
FD42                TTYOUT:
FD42    DBF5                IN      TIS             ; CONSOLE = TTY
FD44    E601                ANI     TRDY
FD46    CA42FD              JZ      TTYOUT          ; LOOP UNTIL READY
FD49    79                  MOV     A,C
FD4A    D3F4                OUT     TTO             ; OUTPUT CHARACTER
FD4C    C9                  RET                     ; RETURN
FD4D                COO:
FD4D    FE02                CPI     BATCH
FD4F    CAA1FD              JZ      LO
FD52    FE01                CPI     CCRT            ; CONSOLE = CRT?
FD54    3EEB                MVI     A,COLOC
FD56    C227FD              JNZ     @USER           ; USER DEFINED CONSOLE OUTPUT
FD59                CRTOUT:
FD59    DBF7                IN      CRTS            ; CONSOLE = CRT
FD5B    E601                ANI     TRDY
FD5D    CA59FD              JZ      CRTOUT          ; LOOP UNTIL READY
FD60    79                  MOV     A,C
FD61    D3F6                OUT     CRTO
FD63    C9                  RET
                    ;
                    ; CONSOLE OUTPUT OF CONSTANT DATA
                    ;
FD64                COMC:
FD64    E3                  XTHL                    ; GET RETURN ADDRESS
FD65    4E                  MOV     C,M             ; GET PARAMETER
FD66    23                  INX     H               ; BUMP RETURN ADDRESS
FD67    E3                  XTHL
```

```
FD68   C330FD          JMP     COM             ; OUTPUT IT
                       ;
                       ; EXTERNALLY REFERENCED ROUTINE
                       ; PUNCH OUTPUT CODE, VALUE EXPECTED IN C
                       ; A, FLAGS, AND C MODIFIED
                       ; STACK USAGE: 2 BYTES
                       ;
FD6B                   POC:                    ; PUNCH A CONSTANT
FD6B   E3                      XTHL
FD6C   4E                      MOV     C,M
FD6D   23                      INX     H
FD6E   E3                      XTHL
FD6F                   PO:                     ; PUNCH OUTPUT
FD6F   3A0300                  LDA     IOBYT           ; GET STATUS BYTE
FD72   E630                    ANI     NOT PMSK        ; GET PUNCH BITS
FD74   CA42FD                  JZ      TTYOUT          ; NO, PUNCH = TTY
FD77   FE10                    CPI     PPTP            ; TEST FOR PTP
FD79   C28BFD                  JNZ     PO1             ; TEST FOR USER DEVICE(S)
FD7C                   PO0:                    ; PUNCH = PTP
FD7C   DBF9                    IN      PTPS            ; GET STATUS
FD7E   E604                    ANI     PTPRY           ; CHECK STATUS
FD80   CA7CFD                  JZ      PO0             ; LOOP UNTIL READY
FD83   79                      MOV     A,C
FD84   D3F8                    OUT     PTPO
FD86   3E20                    MVI     A,PTPADV        ; START PUNCH
FD88   D3F9                    OUT     PTPC
FD8A   C9                      RET
FD8B                   PO1:
FD8B   FE20                    CPI     PUSE1
FD8D   3EF4                    MVI     A,P1LOC
FD8F   CA27FD                  JZ      @USER           ; USER DEFINED PUNCH 1
FD92   3EF7                    MVI     A,P2LOC
FD94   C327FD                  JMP     @USER           ; USER DEFINED PUNCH 2
                       ;
                       ; EXTERNALLY REFERENCED ROUTINE
                       ; LIST OUTPUT CODE
                       ; VALUE EXPECTED IN C, A AND FLAGS MODIFIED
                       ; STACK USAGE: 2 BYTES
                       ;
FD97                   LOM:                    ; LIST OUTPUT
FD97   3A0300                  LDA     IOBYT
FD9A   E603                    ANI     NOT CMSK
FD9C   FE02                    CPI     BATCH           ; DON'T HONOR BREAK KEY IN BATCH MODE
FD9E   C4CDFC                  CNZ     BREAK           ; TEST FOR BREAK KEY
FDA1                   LO:                     ; LIST OUTPUT
FDA1   3A0300                  LDA     IOBYT           ; GET STATUS BYTE
FDA4   E6C0                    ANI     NOT LMSK        ; GET LIST BITS
FDA6   CA42FD                  JZ      TTYOUT          ; LIST = TTY
FDA9   FE40                    CPI     LCRT
FDAB   CA59FD                  JZ      CRTOUT          ; LIST = CRT
FDAE   FEC0                    CPI     LUSE            ; TEST FOR USER DEFINED LIST DEVICE
```

```
FDB0   3EFA          MVI     A,L1LOC
FDB2   CA27FD        JZ      @USER           ; USER DEFINED LIST
FDB5           LP0:
FDB5   DBFB          IN      LPTS
FDB7   E601          ANI     LPTRY
FDB9   CAB5FD        JZ      LP0
FDBC   79            MOV     A,C
FDBD   2F            CMA
FDBE   D3FA          OUT     LPTO
FDC0   C9            RET
               ;
               ; EXTERNALLY REFERENCED ROUTINE
               ; CONSOLE INPUT STATUS CODE
               ; A, FLAGS MODIFIED
               ; STACK USAGE: 2 BYTES
               ;
FDC1           CSTS:                        ; CONSOLE INPUT STATUS
FDC1   3A0300        LDA     IOBYT           ; GET STATUS BYTE
FDC4   E603          ANI     NOT CMSK        ; CONSOLE = TTY?
FDC6   C2CEFD        JNZ     CS0             ; CONSOLE = CRT
FDC9   DBF5          IN      TTS             ; GET TTY STATUS
FDCB   C3D5FD        JMP     CS1
FDCE           CS0:
FDCE   FE01          CPI     CCRT
FDD0   C2DCFD        JNZ     CS3
FDD3   DBF7          IN      CRTS            ; GET CRT STATUS
FDD5           CS1:
FDD5   E602          ANI     RBR
FDD7   3E00          MVI     A,FALSE         ; RETURN FALSE IF NO DATA AVAILABLE
FDD9           CS2:
FDD9   C8            RZ
FDDA   2F            CMA
FDDB   C9            RET                     ; RETURN
FDDC           CS3:
FDDC   FE02          CPI     BATCH
FDDE   3EFF          MVI     A,TRUE
FDE0   CAD9FD        JZ      CS2
FDE3   3EFD          MVI     A,CSLOC         ; USER DEFINE CONSOLE STATUS
FDE5   C327FD        JMP     @USER
               ;
               ; EXTERNALLY REFERENCED ROUTINE
               ; I/O SYSTEM STATUS CODE
               ; STATUS BYTE RETURNED IN A
               ; STACK USAGE: 2 BYTES
               ;
FDE8           IOCHK:
FDE8   3A0300        LDA     IOBYT           ; GET STATUS BYTE
FDEB   C9            RET                     ; RETURN
               ;
               ; EXTERNALLY REFERENCED ROUTINE
               ; SET I/O CONFIGURATION
```

```
                        ; VALUE EXPECTED IN C, A MODIFIED
                        ; STACK USAGE: 2 BYTES
                        ;
FDEC                    IOSET:
FDEC    79                      MOV     A,C
FDED    320300                  STA     IOBYT           ; PUT NEW IOBYT IN MEMORY
FDF0    C9                      RET                     ; RETURN
                        ;
                        ; EXTERNALLY REFERENCED ROUTINE
                        ; RETURN ADDRESS OF END OF MEMORY TO USER
                        ; VALUE RETURNED IN (B,A)
                        ; STACK USAGE: 8 BYTES
                        ;
FDF1                    MEMCK:
FDF1    3A0500                  LDA     MEMTOP+1
FDF4    3D                      DCR     A
FDF5    47                      MOV     B,A
FDF6    3EC0                    MVI     A,USER
FDF8    C9                      RET
                        ;
                        ; EXTERNALLY REFERENCED ROUTINE
                        ; DEFINE USER I/O ENTRY POINTS
                        ; SELECTION CODE IN C
                        ; ENTRY POINT IN D,E
                        ; A, FLAGS MODIFIED
                        ; STACK USAGE: 8 BYTES
                        ;
FDF9                    IODEF:
FDF9    E5                      PUSH    H
FDFA    C5                      PUSH    B
     1                  +       GET     XTBL+1
FDFB 1  2A0400          +       LHLD    MEMTOP
FDFE 1  2EE9            +       MVI     L,000E9H AND 0FFH
FE00    79                      MOV     A,C
FE01    FE08                    CPI     UCS+1
FE03    D226F8                  JNC     ERROR           ; INVALID SELECTION CODE
FE06    81                      ADD     C               ; DOUBLE INDEX
FE07    81                      ADD     C               ; TRIPLE INDEX
FE08    4F                      MOV     C,A
FE09    0600                    MVI     B,0
FE0B    09                      DAD     B               ; COMPUTE LOCATION OF I/O OPERAND
FE0C    73                      MOV     M,E             ; STORE BRANCH OPERAND IN INSTRUCTION
FE0D    23                      INX     H
FE0E    72                      MOV     M,D
FE0F    C1                      POP     B
FE10    E1                      POP     H
FE11    C9                      RET
                        ;
                        ; EXTERNALLY REFERENCED ROUTINE
                        ;
FE12                    IOCON:                          ; INTERNAL ENTRY POINT FOR I/O CONTROL
```

```
FE12   E3              XTHL
FE13   4E              MOV     C,M
FE14   23              INX     H
FE15   E3              XTHL
FE16                SPCL:
FE16   E5              PUSH    H
FE17   2600            MVI     H,0
FE19   69              MOV     L,C
FE1A   74              MOV     M,H
FE1B   E1              POP     H
FE1C   C9              RET
                    ;
                    ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                    ;
                    ; END OF I/O SUBROUTINES, BEGINNING OF MONITOR SUBROUTINES
                    ;
                    ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                    ;
                    ; READ TWO ASCII CHARACTERS, DECODE INTO 8 BITS BINARY
                    ;
                    ; REGISTER USAGE
                    ;
                    ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                    ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                    ; U = USED AS INPUT.
                    ;
                    ;       A -
                    ;       B -                 C -
                    ;       D -                 E -
                    ;       H -                 L -
                    ;       CARRY - X           ZERO - X
                    ;       SIGN - X            PARITY - X
                    ;       SP -                PC -
                    ;       STACK USAGE:
                    ;
FE1D                BYTE:
FE1D   C5              PUSH    B
FE1E   CDADFF          CALL    RIX             ; READ CHAR FROM TAPE
FE21   CDEFFE          CALL    NIBBLE          ; CONVERT ASCII TO HEX
FE24   07              RLC
FE25   07              RLC
FE26   07              RLC
FE27   07              RLC                     ; SHIFT FOUR PLACES
FE28   4F              MOV     C,A
FE29   CDADFF          CALL    RIX
FE2C   CDEFFE          CALL    NIBBLE          ; GET LOWER NIBBLE
FE2F   B1              ORA     C
FE30   4F              MOV     C,A
FE31   82              ADD     D               ; UPDATE CHECKSUM
FE32   57              MOV     D,A
FE33   79              MOV     A,C
```

```
FE34   C1              POP     B
FE35   C9              RET                     ; RETURN
                ;
                ; CONVERT 4 BIT HEX VALUE TO ASCII CHARACTER
                ; INPUT - 0...0FH
                ; OUTPUT - 30H...39H, 41H...46H
                ;
                ; REGISTER USAGE
                ;
                ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                ; U = USED AS INPUT.
                ;
                ;       A - U,X,S
                ;       B -             C - S
                ;       D -             E -
                ;       H -             L -
                ;       CARRY - X       ZERO - X
                ;       SIGN - X        PARITY - X
                ;       SP -            PC -
                ;       STACK USAGE:
                ;
FE36            CONV:
FE36   E60F            ANI     0FH             ; ONLY 4 LSB ARE SIGNIFICANT
FE38   C690            ADI     90H             ; SET UP A SO THAT A-F CAUSE CARRY
FE3A   27              DAA
FE3B   CE40            ACI     40H             ; ADD IN CARRY AND ADJUST UPPER NIBBLE
FE3D   27              DAA
FE3E   4F              MOV     C,A
FE3F   C9              RET                     ; RETURN
                ;
                ; TYPE CARRIAGE RETURN AND LINE FEED ON CONSOLE
                ;
FE40            CRLF:
FE40   CD64FD          CALL    COMC
FE43   0D              DB      CR
FE44   CD64FD          CALL    COMC
FE47   0A              DB      LF
FE48   C9              RET
                ;
                ; PRINT CONTENTS OF HL IN HEX ON LIST DEVICE
                ;
                ; REGISTER USAGE
                ;
                ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                ; U = USED AS INPUT.
                ;
                ;       A - X
                ;       B -             C -
                ;       D -             E -
```

```
                        ;       H - U                   L - U
                        ;       CARRY - X               ZERO - X
                        ;       SIGN - X                PARITY - X
                        ;       SP -                    PC -
                        ;       STACK USAGE:
                        ;
FE49                    DADR:
FE49    7C                      MOV     A,H             ; PRINT MSB
FE4A    CD51FE                  CALL    DBYTE
FE4D    7D                      MOV     A,L             ; PRINT LSB
FE4E    C351FE                  JMP     DBYTE
                        ;
                        ; LIST A BYTE ON THE LIST DEVICE AS 2 ASCII CHARACTERS
                        ;
                        ; REGISTER USAGE
                        ;
                        ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                        ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                        ; U = USED AS INPUT.
                        ;
                        ;       A - U,X
                        ;       B -                     C - X
                        ;       D -                     E -
                        ;       H -                     L -
                        ;       CARRY - X               ZERO - X
                        ;       SIGN - X                PARITY - X
                        ;       SP -                    PC -
                        ;       STACK USAGE:
                        ;
FE51                    DBYTE:
FE51    F5                      PUSH    PSW             ; SAVE A COPY OF A
FE52    0F                      RRC
FE53    0F                      RRC
FE54    0F                      RRC
FE55    0F                      RRC
FE56    CD36FE                  CALL    CONV
FE59    CD97FD                  CALL    LOM
FE5C    F1                      POP     PSW             ; RETRIEVE ORIGINAL VALUE
FE5D    CD36FE                  CALL    CONV
FE60    C397FD                  JMP     LOM
                        ;
                        ; 1.0 MS. DELAY
                        ;
                        ; REGISTER USAGE
                        ;
                        ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                        ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                        ; U = USED AS INPUT.
                        ;
                        ;       A -
                        ;       B -                     C -
```

```
                        ;       D -             E -
                        ;       H -             L -
                        ;       CARRY - X       ZERO - X
                        ;       SIGN - X        PARITY - X
                        ;       SP -            PC -
                        ;       STACK USAGE:
                        ;
FE63                    DELAY:
FE63    DBFF                    IN      RTC             ; LOOP UNTIL REAL TIME
FE65    E601                    ANI     RTCS            ; CLOCK CHANGES PHASE
FE67    CA63FE                  JZ      DELAY
FE6A    C9                      RET                     ; RETURN
                        ;
                        ; DISPLAY THE CONTENTS OF A USER REGISTER
                        ;
                        ; HL POINTS TO CHARACTER IN ACTBL,
                        ; RETURNS POINTING TO NEXT CHARACTER
                        ; DE IS RETURNED WITH ADDRESS OF REGISTER LOCATION
                        ; B IS RETURNED WITH THE REGISTER PRECISION
                        ;
FE6B                    DREG:
FE6B    23                      INX     H
FE6C    5E                      MOV     E,M             ; INCREMENT HL TO POINT AT DISPLACEMENT
FE6D    3A0500                  LDA     MEMTOP+1        ; LOCATE REGISTER CONTENTS
FE70    57                      MOV     D,A             ; IN TOP OF MEMORY
FE71    23                      INX     H
FE72    46                      MOV     B,M             ; PRECISION, 0=8 BITS, 1=16 BITS
FE73    23                      INX     H               ; POINT AT NEXT REGISTER IDENTIFIER
FE74    1A                      LDAX    D               ; 8/16 BIT DISPLAY AND MODIFICATION
FE75    CDAAFE                  CALL    LBYTE           ; MSB OF 16 BIT REG, ALL OF 8 BIT REG
FE78    05                      DCR     B               ; TEST PRECISION
FE79    F8                      RM                      ; 8 BIT DISPLAY, RETURN
FE7A    1B                      DCX     D
FE7B    1A                      LDAX    D
FE7C    C3AAFE                  JMP     LBYTE           ; LSB OF 16 BIT REG
                        ;
                        ; EVALUATE EXPRESSION: <EXPR>,<EXPR>,<EXPR>
                        ;
                        ; THE C REGISTER CONTAINS THE NUMBER OF PARAMETERS REQUIRED
                        ; (1,2, OR 3).  PARAMETERS ARE RETURNED ON THE STACK.
                        ;
                        ; REGISTER USAGE
                        ;
                        ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                        ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                        ; U = USED AS INPUT.
                        ;
                        ;       A -
                        ;       B -             C -
                        ;       D -             E -
                        ;       H -             L -
```

```
                        ;       CARRY - X       ZERO - X
                        ;       SIGN - X        PARITY - X
                        ;       SP -            PC -
                        ;       STACK USAGE:
                        ;
FE7F                    EXPR:
FE7F    CDCBFE          	CALL    PARAM           ; GET A PARAMETER
FE82    E3              	XTHL                    ; GET RETURN ADDRESS OFF STACK,
                                                    ; PUT HL ON
FE83    E5              	PUSH    H               ; REPLACE RETURN ADDRESS
FE84    0D              	DCR     C               ; DECREMENT COUNT
FE85    D28CFE          	JNC     EX0             ; COMMA ENTERED
FE88    C226F8          	JNZ     ERROR           ; INCORRECT PARAM COUNT
FE8B    C9              	RET
FE8C                    EX0:
FE8C    C27FFE          	JNZ     EXPR            ; GET ANOTHER PARAMETER
FE8F    C326F8          	JMP     ERROR           ; NOT TERMINATED WITH CR
                        ;
                        ; COMPARE HL WITH DE:
                        ; IF HL <= DE THEN CARRY = 0;
                        ; IF HL > DE THEN CARRY = 1;
                        ;
                        ; REGISTER USAGE
                        ;
                        ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                        ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                        ; U = USED AS INPUT.
                        ;
                        ;       A -
                        ;       B -             C -
                        ;       D -             E -
                        ;       H -             L -
                        ;       CARRY - X       ZERO - X
                        ;       SIGN - X        PARITY - X
                        ;       SP -            PC -
                        ;       STACK USAGE:
                        ;
FE92                    HILO:
FE92    23              	INX     H               ; BUMP HL
FE93    7C              	MOV     A,H             ; TEST FOR HL = 0
FE94    B5              	ORA     L
FE95    37              	STC
FE96    C8              	RZ
FE97    7B              	MOV     A,E             ; DE - HL, SET/RESET CARRY
FE98    95              	SUB     L
FE99    7A              	MOV     A,D
FE9A    9C              	SBB     H
FE9B    C9              	RET                     ; RETURN
                        ;
                        ; CONVERT NIBBLE IN A-REGISTER TO ASCII IN A-REGISTER
                        ; AND PRINT ON CONSOLE DEVICE
```

```
                        ;
                        ; REGISTER USAGE
                        ;
                        ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                        ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                        ; U = USED AS INPUT.
                        ;
                        ;       A -
                        ;       B -                 C -
                        ;       D -                 E -
                        ;       H -                 L -
                        ;       CARRY - X           ZERO - X
                        ;       SIGN - X            PARITY - X
                        ;       SP -                PC -
                        ;       STACK USAGE:
                        ;
FE9C                    HXD:
FE9C    CD36FE                  CALL    CONV
FE9F    C330FD                  JMP     COM
                        ;
                        ; PRINT CONTENTS OF HL IN HEX ON CONSOLE DEVICE
                        ;
                        ; REGISTER USAGE
                        ;
                        ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                        ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                        ; U = USED AS INPUT.
                        ;
                        ;       A -
                        ;       B -                 C -
                        ;       D -                 E -
                        ;       H -                 L -
                        ;       CARRY - X           ZERO - X
                        ;       SIGN - X            PARITY - X
                        ;       SP -                PC -
                        ;       STACK USAGE:
                        ;
FEA2                    LADR:
FEA2    7C                      MOV     A,H             ; PRINT MSB
FEA3    CDAAFE                  CALL    LBYTE
FEA6    7D                      MOV     A,L             ; PRINT LSB
FEA7    C3AAFE                  JMP     LBYTE
                        ;
                        ; LIST A BYTE AS 2 ASCII CHARACTERS
                        ;
                        ; REGISTER USAGE
                        ;
                        ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                        ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                        ; U = USED AS INPUT.
                        ;
```

```
                        ;       A -
                        ;       B -             C -
                        ;       D -             E -
                        ;       H -             L -
                        ;       CARRY - X       ZERO - X
                        ;       SIGN - X        PARITY - X
                        ;       SP -            PC -
                        ;       STACK USAGE:
                        ;
FEAA                    LBYTE:
FEAA    F5                      PUSH    PSW             ; SAVE A COPY OF A
FEAB    0F                      RRC
FEAC    0F                      RRC
FEAD    0F                      RRC
FEAE    0F                      RRC
FEAF    CD9CFE                  CALL    HXD
FEB2    F1                      POP     PSW             ; RETRIEVE ORIGINAL VALUE
FEB3    C39CFE                  JMP     HXD
                        ;
                        ; PRINT CR, LF ON LIST DEVICE
                        ;
FEB6                    LCRLF:
FEB6    0E0D                    MVI     C,CR
FEB8    CD97FD                  CALL    LOM
FEBB    0E0A                    MVI     C,LF
FEBD    C397FD                  JMP     LOM
                        ;
                        ; PUNCH 6 INCHES OF LEADER
                        ;
                        ; REGISTER USAGE
                        ;
                        ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                        ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                        ; U = USED AS INPUT.
                        ;
                        ;       A - X
                        ;       B - X           C - X
                        ;       D -             E -
                        ;       H -             L -
                        ;       CARRY - X       ZERO - X
                        ;       SIGN - X        PARITY - X
                        ;       SP -            PC -
                        ;       STACK USAGE:
                        ;
FEC0                    LEAD:
FEC0    063C                    MVI     B,60            ; SET TO PUNCH 6 INCHES OF NULLS
FEC2                    LE0:
FEC2    CD6BFD                  CALL    POC
FEC5    00                      DB      0
FEC6    05                      DCR     B
FEC7    C2C2FE                  JNZ     LE0
```

```
FECA  C9              RET                     ; RETURN
                ;
                ; COLLECT A HEXADECIMAL PARAMETER
                ;
FECB            PARAM:
FECB  CD1FFF          CALL    PCHK            ; GET FIRST CHARACTER
FECE  CA26F8          JZ      ERROR           ; DISALLOW NULL PARAMETERS
FED1            PA0:
FED1  210000          LXI     H,0             ; CLEAR ACCUMULATOR
FED4            PA1:
FED4  47              MOV     B,A             ; SAVE DELIMITER CHARACTER
FED5  CDEFFE          CALL    NIBBLE          ; CONVERT TO HEX
FED8  DAE7FE          JC      PA2             ; NOT LEGAL CHAR, TREAT AS DELIMITER
FEDB  29              DAD     H               ; *2
FEDC  29              DAD     H               ; *4
FEDD  29              DAD     H               ; *8
FEDE  29              DAD     H               ; *16
FEDF  B5              ORA     L
FEE0  6F              MOV     L,A
FEE1  CDE8FF          CALL    TI              ; GET SUBSEQUENT CHARACTERS
FEE4  C3D4FE          JMP     PA1             ; DECODE NEXT CHARACTER
FEE7            PA2:
FEE7  78              MOV     A,B
FEE8  CD22FF          CALL    P2C
FEEB  C226F8          JNZ     ERROR
FEEE  C9              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.
                ;
                ; REGISTER USAGE
                ;
                ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                ; U = USED AS INPUT.
                ;
                ;       A -
                ;       B -             C -
                ;       D -             E -
                ;       H -             L -
                ;       CARRY - X       ZERO - X
                ;       SIGN - X        PARITY - X
                ;       SP -            PC -
                ;       STACK USAGE:
                ;
FEEF            NIBBLE:
FEEF  D630            SUI     '0'
FEF1  D8              RC                      ; FILTER OUT 0-2FH
FEF2  C6E9            ADI     '0'-'G'
FEF4  D8              RC                      ; FILTER OUT 47H-0FFH
```

```
FEF5   C606            ADI     6
FEF7   F2FDFE          JP      NIO             ; TAKE BRANCH FOR A-F
FEFA   C607            ADI     7
FEFC   D8              RC                      ; FILTER OUT 3AH-40H
FEFD             NIO:
FEFD   C60A            ADI     10
FEFF   B7              ORA     A               ; CLEAR ERROR FLAG
FF00   C9              RET                     ; RETURN
                 ;
                 ; PUNCH CONTENTS OF HL IN HEX ON PUNCH DEVICE
                 ;
                 ; REGISTER USAGE
                 ;
                 ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                 ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                 ; U = USED AS INPUT.
                 ;
                 ;     A -
                 ;     B -             C -
                 ;     D -             E -
                 ;     H -             L -
                 ;     CARRY - X       ZERO - X
                 ;     SIGN - X        PARITY - X
                 ;     SP -            PC -
                 ;     STACK USAGE:
                 ;
FF01             PADR:
FF01   7C              MOV     A,H
FF02   CD09FF          CALL    PBYTE
FF05   7D              MOV     A,L
FF06   C309FF          JMP     PBYTE
                 ;
                 ; PUNCH A BYTE AS 2 ASCII CHARACTERS
                 ;
                 ; REGISTER USAGE
                 ;
                 ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                 ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                 ; U = USED AS INPUT.
                 ;
                 ;     A -
                 ;     B -             C -
                 ;     D -             E -
                 ;     H -             L -
                 ;     CARRY - X       ZERO - X
                 ;     SIGN - X        PARITY - X
                 ;     SP -            PC -
                 ;     STACK USAGE:
                 ;
FF09             PBYTE:
FF09   5F              MOV     E,A             ; SAVE VALUE
```

```
FF0A   0F            RRC
FF0B   0F            RRC
FF0C   0F            RRC
FF0D   0F            RRC
FF0E   CD36FE        CALL    CONV
FF11   CD6FFD        CALL    PO
FF14   7B            MOV     A,E
FF15   CD36FE        CALL    CONV
FF18   CD6FFD        CALL    PO
FF1B   7B            MOV     A,E
FF1C   82            ADD     D
FF1D   57            MOV     D,A
FF1E   C9            RET                     ; RETURN
                ;
                ; TEST FOR NULL INPUT PARAMETER
                ;
                ; REGISTER USAGE
                ;
                ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                ; U = USED AS INPUT.
                ;
                ;       A -
                ;       B -             C -
                ;       D -             E -
                ;       H -             L -
                ;       CARRY - X       ZERO - X
                ;       SIGN - X        PARITY - X
                ;       SP -            PC -
                ;       STACK USAGE:
                ;
FF1F            PCHK:
FF1F   CDE8FF        CALL    TI              ; GET A CHARACTER
FF22            P2C:
FF22   FE20          CPI     ' '
FF24   C8            RZ
FF25   FE2C          CPI     ','
FF27   C8            RZ
FF28   FE0D          CPI     CR
FF2A   37            STC
FF2B   C8            RZ
FF2C   3F            CMC
FF2D   C9            RET
                ;
                ; PUNCH CR,LF
                ;
                ; REGISTER USAGE
                ;
                ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                ; U = USED AS INPUT.
```

```
                        ;
                        ;       A -
                        ;       B -                 C -
                        ;       D -                 E -
                        ;       H -                 L -
                        ;       CARRY - X           ZERO - X
                        ;       SIGN - X            PARITY - X
                        ;       SP -                PC -
                        ;       STACK USAGE:
                        ;
FF2E                    PEOL:
FF2E    CD6BFD                  CALL    POC
FF31    0D                      DB      CR
FF32    CD6BFD                  CALL    POC
FF35    0A                      DB      LF
FF36    C9                      RET
                        ;
                        ; RESTART 2 CODE, (PROGRAMMED BREAKPOINT).
                        ;
                        ; THIS ROUTINE IS ENTERED VIA A RESTART 0 (RST 0) 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.
                        ;
FF37                    RESTART:
FF37    F3                      DI                      ; DISABLE IF SOFTWARE TRAP
FF38    E5                      PUSH    H               ; SAVE MACHINE STATE
FF39    D5                      PUSH    D
FF3A    C5                      PUSH    B
FF3B    F5                      PUSH    PSW
FF3C    E1                      POP     H               ; GET A,F IN H,L
FF3D    DBFC                    IN      MASK
FF3F    F5                      PUSH    PSW             ; SAVE INTERRUPT MASK
FF40    E5                      PUSH    H               ; STACK A,F ABOVE MASK
FF41    3EFE                    MVI     A,NOT INT0
FF43    D3FC                    OUT     MASK            ; SET DEFAULT INTERRUPT MASK
     1                  +       GET     EXIT            ; FIND TOP OF MEMORY
FF45 1  2A0400          +       LHLD    MEMTOP
FF48 1  2ED2            +       MVI     L,EXIT AND 0FFH
                                                        ; HL = NEW STACK POINTER
FF4A    EB                      XCHG
     1                  +       FETCH   12              ; COMPUTE ORIGINAL STACK POINTER
FF4B 1  210C00          +       LXI     H,0000CH
FF4E 1  39              +       DAD     SP
FF4F    0605                    MVI     B,5             ; COUNT FOR TRANSFER OF MACHINE STATE
                                                        ; TO STORAGE (MOVE THE STACK)
FF51    EB                      XCHG
FF52                    RST0:
FF52    2B                      DCX     H
FF53    72                      MOV     M,D
```

```
FF54   2B              DCX     H
FF55   73              MOV     M,E
FF56   D1              POP     D
FF57   05              DCR     B
FF58   C252FF          JNZ     RST0
FF5B   C1              POP     B               ; GET OLD PC = B,C; OLD HL = D,E
FF5C   0B              DCX     B               ; DECREMENT TO POINT AT TRAPPED CODE
FF5D   F9              SPHL                    ; NEW STACK VALUE
     1           +     GET     TLOC
FF5E 1 2A0400    +     LHLD    MEMTOP
FF61 1 2EE2      +     MVI     L,TLOC AND 0FFH
FF63   7E              MOV     A,M             ; TEST IF THIS IS A PROGRAMMED RESTART
FF64   91              SUB     C               ; OR A CONSOLE RESTART
FF65   23              INX     H
FF66   7E              MOV     A,M
FF67   98              SBB     B
FF68   CA7AFF          JZ      RST1            ; PC MATCHES TRAP1, PROGRAMMED RESTART
FF6B   23              INX     H
FF6C   23              INX     H
FF6D   7E              MOV     A,M
FF6E   91              SUB     C
FF6F   23              INX     H
FF70   7E              MOV     A,M
FF71   98              SBB     B
FF72   CA7AFF          JZ      RST1            ; PC MATCHES TRAP2, PROGRAMMED RESTART
FF75   3E20            MVI     A,EOI           ; END OF INTERRUPT
FF77   D3FD            OUT     REVRT           ; REVERT INTERRUPT SYSTEM
FF79   03              INX     B               ; ADJUST PC FOR CONSOLE RESTART
FF7A             RST1:
     1           +     GET     LLOC
FF7A 1 2A0400    +     LHLD    MEMTOP
FF7D 1 2EDC      +     MVI     L,LLOC AND 0FFH
FF7F   73              MOV     M,E
FF80   23              INX     H
FF81   72              MOV     M,D             ; SAVE OLD HL
FF82   2EE0            MVI     L,PLOC-1
FF84   71              MOV     M,C             ; SAVE OLD PC
FF85   23              INX     H
FF86   70              MOV     M,B
FF87   C5              PUSH    B
FF88   CD64FD          CALL    COMC
FF8B   2A              DB      '*'
FF8C   E1              POP     H               ; RETRIEVE OLD PC FOR DISPLAY
FF8D   CDA2FE          CALL    LADR            ; DISPLAY PC
     1           +     GET     TLOC            ; CLEAR TRAPS
FF90 1 2A0400    +     LHLD    MEMTOP
FF93 1 2EE2      +     MVI     L,TLOC AND 0FFH
FF95   1602            MVI     D,2             ; SET COUNT FOR TWO TRAPS
FF97             RST2:
FF97   4E              MOV     C,M             ; GET LSB OF ADDRESS
FF98   AF              XRA     A
```

```
FF99   77            MOV     M,A
FF9A   23            INX     H
FF9B   46            MOV     B,M             ; GET MSB OF ADDRESS
FF9C   77            MOV     M,A
FF9D   23            INX     H
FF9E   79            MOV     A,C
FF9F   B0            ORA     B               ; TEST FOR VALID TRAP
FFA0   CAA5FF        JZ      RST3            ; ADDRESS = 0, NO TRAP TO RESTORE
FFA3   7E            MOV     A,M             ; GET OPCODE BYTE
FFA4   02            STAX    B               ; REPLACE IT
FFA5   .         RST3:
FFA5   23            INX     H               ; POINT TO NEXT TRAP ADDRESS
FFA6   15            DCR     D
FFA7   C297FF        JNZ     RST2            ; REPEAT FOR TRAP 2
FFAA   C330F8        JMP     START           ; ENTER MONITOR
                     ;
                     ; GET CHARACTER FROM READER, MASK OFF PARITY BIT
                     ;
                     ; REGISTER USAGE
                     ;
                     ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                     ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                     ; U = USED AS INPUT.
                     ;
                     ;     A -
                     ;     B -             C -
                     ;     D -             E -
                     ;     H -             L -
                     ;     CARRY - X       ZERO - X
                     ;     SIGN - X        PARITY - X
                     ;     SP -            PC -
                     ;     STACK USAGE:
                     ;
FFAD                 RIX:
FFAD   CDD5FC        CALL    RI
FFB0   DA26F8        JC      ERROR           ; READER TIMEOUT ERROR
FFB3   E67F          ANI     7FH
FFB5   C9            RET                     ; RETURN
                     ;
                     ; GET A CHARACTER FROM THE CONSOLE DEVICE, ASSUME
                     ; THAT IT'S A SOCKET SELECT OPTION CHARACTER, AND TURN
                     ; IT INTO A SOCKET SELECTION MASK. IF THE CHARACTER IS
                     ; IN ERROR, TAKE THE STANDARD ERROR EXIT.
                     ;
                     ; REGISTER USAGE
                     ;
                     ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                     ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                     ; U = USED AS INPUT.
                     ;
                     ;     A -
```

```
                        ;       B -             C -
                        ;       D -             E -
                        ;       H -             L -
                        ;       CARRY - X       ZERO - X
                        ;       SIGN - X        PARITY - X
                        ;       SP -            PC -
                        ;       STACK USAGE:
                        ;
FFB6                    SELECT:
FFB6    DBF1                    IN      PSTAT           ; READ PROM PROGRAMMER STATUS
FFB8    B7                      ORA     A               ; PROM PERIPHERAL CONNECTED?
FFB9    CA26F8                  JZ      ERROR           ; NO, ABORT
FFBC    CDE8FF                  CALL    TI
FFBF    FE54                    CPI     'T'
FFC1    C2C8FF                  JNZ     SE0
FFC4    AF                      XRA     A
FFC5    C3CFFF                  JMP     SE1
FFC8                    SE0:
FFC8    FE46                    CPI     'F'
FFCA    C226F8                  JNZ     ERROR
FFCD    3EFF                    MVI     A,0FFH
FFCF                    SE1:
FFCF    320400                  STA     MSK
FFD2    CDE8FF                  CALL    TI
FFD5    D658                    SUI     'X'
FFD7    01FF00                  LXI     B,0FFH          ; OPTION X SELECTED
FFDA    C8                      RZ
FFDB    3D                      DCR     A
FFDC    01F030                  LXI     B,(PSOCK OR PNIB) SHL 8 + 0F0H
FFDF    C8                      RZ
FFE0    3D                      DCR     A
FFE1    C226F8                  JNZ     ERROR
FFE4    010F20                  LXI     B,PSOCK SHL 8 + 0FH     ; OPTION Z SELECTED
FFE7    C9                      RET
                        ;
                        ; INPUT FROM CONSOLE, ECHOED AND RETURNED IN A
                        ;
                        ; REGISTER USAGE
                        ;
                        ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
                        ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
                        ; U = USED AS INPUT.
                        ;
                        ;       A -
                        ;       B -             C -
                        ;       D -             E -
                        ;       H -             L -
                        ;       CARRY - X       ZERO - X
                        ;       SIGN - X        PARITY - X
                        ;       SP -            PC -
                        ;       STACK USAGE:
```

```
                        ;
  FFE8                  TI:
  FFE8    C5                    PUSH    B
  FFE9    CDA2FC                CALL    CI
  FFEC    E67F                  ANI     7FH             ; MASK PARITY
  FFEE    FE03                  CPI     ETX             ; TEST FOR BREAK
  FFF0    CA26F8                JZ      ERROR           ; ABORT COMMAND
  FFF3    4F                    MOV     C,A
  FFF4    CD3AFD                CALL    CO              ; ECHO
  FFF7    79                    MOV     A,C
  FFF8    C1                    POP     B
  FFF9    C9                    RET                     ; RETURN
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ; END OF PROGRAM
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                              END
NO PROGRAM ERRORS
```

SYMBOL TABLE

* 01

@USER	FD27	A	0007	ACT	F8EC	ACTBL	FC7D
ALOC	00CF	ALT	F904	APT	F8FC	ART	F8F4
AS0	F897	AS1	F8A6	AS2	F8BE	AS3	F8CB
ASSIG	F88F	B	0000	BASE	F800	BATCH	0002
BEGIN	F800	BLK	FD2E	BLOC	00CB	BN0	F918
BN1	F922	BNPF	F90C	BOOT	0002	BREAK	FCCD
BYTE	FE1D	C	0001	CASE	0F0D	CCRT	0001
CI	FCA2	CI0	FCB4	CI1	FCC3	CILOC	00E8
CL5	0000 *	CL6	0004 *	CL7	0008 *	CL8	000C
CLERR	0010 *	CLFLG	000F	CLOC	00CA	CM0	F958
CM1	F98B	CMSK	00FC	CO	FD3A	CO0	FD4D
COLOC	00EB	COM	FD30	COMC	FD64	COMP	F949
CONV	FE36	CPFLG	000E	CR	000D	CRLF	FE40
CRTC	00F7	CRT1	00F6	CRTIN	FCB9	CRTO	00F6
CRTOU	FD59	CRTS	00F7	CS0	FDCE	CS1	FDD5
CS2	FDD9	CS3	FDDC	CSLOC	00FD	CSTS	FDC1
CTBL	F85F	CTTY	0000	CUSE	0003	D	0002
DADR	FE49	DATE	2608	DBYTE	FE51	DEBUG	0000
DELAY	FE63	DI0	F99D	DI1	F9A3	DI2	F9BB
DISP	F994	DLOC	00C9	DREG	FE6B	DSR	0080 *
DTR	0002	E	0003	ELOC	00C8	ENDX	0100 *
EOF	F9C8	EOI	0020	ERROP	F826	ETX	0003
EX0	FE8C	EXIT	00D2	EXPR	FE7F	FALSE	0000
FETCH	0F9C	FI0	F9ED	FILL	F9E6	FLOC	00CE
GET	0F95	GO0	FA0D	GO1	FA13	GO2	FA23
GO3	FA2B	GO4	FA3A	GOTO	F9F5	H	0004
HEXN	FA3E	HILO	FE92	HLOC	00DD	HXD	FE9C
ICON	00F3	ICRTI	0020 *	ICRTO	0010 *	ILOC	00CD
ILPT	0040 *	INIT	0003 *	INITI	0006	INT0	0001
INT1	0002 *	INT2	0004 *	INT3	0008 *	INT4	0010 *
INT5	0020 *	INT6	0040 *	INT7	0080 *	IOBYT	0003
IOCHK	FDE8	IOCON	FE12	IODEF	FDF9	IOSET	FDEC
IPTP	0004 *	IPTR	0008 *	ISTAT	00FA *	ITTYI	0002 *
ITTYO	0001 *	L	0005	L1LOC	00FA	LADR	FEA2
LBYTE	FEAA	LCRLF	FEB6	LCRT	0040	LCT	0018
LE0	FEC2	LEAD	FEC0	LF	000A	LLOC	00DC
LLPT	0080	LMSK	003F	LO	FDA1	LO0	FA62
LO1	FA6C	LO2	FA79	LOAD	FA59	LOCK	00FE *
LOM	FD97	LP0	FDB5	LPTC	00FB *	LPTO	00FA
LPTRY	0001	LPTS	00FB	LTBL	F8DC	LTTY	0000
LUSE	00C0	LVER	0015	M	0006	MASK	00FC
MEMCK	FDF1	MEMIO	0004	MENB	0080 *	MOVE	FA8E
MSK	0004	MV0	FA95	NI0	FEFD	NIBBL	FEEF
NREGS	000C	NU0	FAAB	NULL	FA9F	OLFLG	000C
OPFLG	000B	ORFLG	0007	P1LOC	00F4	P2C	FF22
P2LOC	00F7	PA0	FED1	PA1	FED4	PA2	FEE7
PADR	FF01	PARAM	FECB	PBYTE	FF09	PCHK	FF1F
PCOMP	0002	PDATA	00F0	PENB	0010 *	PEOL	FF2E
PEVEN	0020 *	PGRDY	0001	PHI	00F1	PLO	00F2
PLOC	00E1	PMSK	00CF	PNIB	0010	PO	FD6F
PO0	FD7C	PO1	FD8B	POC	FD6B	PPTP	0010
PR0	FACD	PR1	FADA	PR2	FAEF	PROG	FAB7

```
PSOCK   0020       PSTAT   00F1       PSW     0006       PTPAD   0020
PTPC    00F9       PTPO    00F8       PTPRE   0010 *     PTPRY   0004
PTPS    00F9       PTRAD   0008       PTRC    00F9       PTRDY   0001
PTRI    00F8       PTRRE   0004 *     PTRS    00F9       PTTY    0000
PUSE1   0020       PUSE2   0030       Q0      FB02       Q1      FB1E
Q2      FB29       QUERY   FAF5       R110    0002       R12@2   0001 *
R1LOC   00EE       R24@1   0003       R2LOC   00F1       R3@2    0003 *
R48@1   0002 *     R6@2    0002 *     R96@1   0001 *     RBR     0002
READ    FB33       RED0    FB3E       RED1    FB60       RED2    FB78
RED3    FB83       RESET   0000       RESTA   FF37       REVRT   00FD
RFR     0020 *     RI      FCD5       RI0     FCE5       RI1     FCF3
RI2     FCF7       RI3     FCFC       RI4     FD07       RI5     FD18
RI6     FD1D       RIX     FFAD       RMSK    00F3       ROV     0010 *
RPAR    0008 *     RPTR    0004       RRFLG   000D *     RST0    FF52
RST1    FF7A       RST2    FF97       RST3    FFA5       RTC     00FF
RTCS    0001       RTS     0020       RTTY    0000       RUSE1   0008
RUSE2   000C       RXEN    0004       SBASE   0000       SE0     FFC8
SE1     FFCF       SELEC   FFB6       SH0     0006       SH1     002C
SH2     006B       SH3     0082       SH4     008E       SH5     00A5
SH6     00AE       SIZE    0F40       SLOC    00D1       SP      0006
SPCL    FE16       ST1     0040       ST15    0080 *     ST2     00C0
START   F830       SU0     FB97       SU1     FBAD       SUBS    FB93
TBE     0004 *     TI      FFE8       TLOC    00E2       TOS     00C8
TOUT    00FA       TR0     FBBF       TRAN    FBB1       TRDY    0001
TRUE    FFFF       TTC     00F5       TTI     00F4       TTO     00F4
TTS     00F5       TTYAD   0002       TTYIN   FCAA       TTYOU   FD42
TXEN    0001       UCI     0000 *     UCO     0001 *     UCS     0007
UL1     0006 *     UNTIL   0F86       UP1     0004 *     UP2     0005 *
UR1     0002 *     UR2     0003 *     USER    00C0       USRST   0040 *
VER     000B       VERS    00B1       WHILE   0F8E       WR0     FBE6
WR1     FBEE       WR2     FBF9       WR3     FC08       WRITE   FBDD
X       FC22       X0      FC2D       X1      FC3B       X2      FC3E
X3      FC5B       X4      FC5C       X5      FC66       X6      FC69
XTBL    00E8

* 02

LOOP    0019

* 03

* 04

* 05

LOOP    F8AC

* 06

LOOP    F8B4

* 07

LOOP    F8CB

* 08
```

W. ZOLNEROVICH


```
;
;          INTELLEC/MDS MONITOR
;             VERSION 1.2
;
;          COPYRIGHT (C) 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>
;                     <QUERY STATUS 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<T/F><SOCKET><NUMBER>,<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<T/F><SOCKET><NUMBER>,<NUMBER>,<NUMBER>
```

```
                        ;
                        ; <QUERY STATUS COMMAND> ::= Q
                        ;
                        ; <READ HEXADECIMAL FILE COMMAND> ::= R<NUMBER>
                        ;
                        ; <SUBSTITUTE MEMORY COMMAND> ::= S<NUMBER>...
                        ;
                        ; <TRANSFER COMMAND> ::= T<T/F><SOCKET><NUMBER>,<NUMBER>
                        ;
                        ; <WRITE HEXADECIMAL RECORD COMMAND> ::= W<NUMBER>,<NUMBER>
                        ;
                        ; <REGISTER MODIFY COMMAND> ::= X<REGISTER IDENTIFIER><NUMBER>...
                        ;
                        ; <LOGICAL DEVICE> ::= CONSOLE!READER!LIST!PUNCH
                        ;
                        ; <PHYSICAL DEVICE> ::= CRT!TTY!PTR!PTP!LPT!BATCH!1!2
                        ;
                        ; <REGISTER IDENTIFIER> ::= A!B!C!D!E!F!H!I!L!M!P!S
                        ;
                        ; <SOCKET> ::= X!Y!Z
                        ;
                        ; <T/F> ::= T!F
                        ;
                        ; <NUMBER> ::=        <HEX DIGIT>
                        ;             <NUMBER><HEX DIGIT>
                        ;
                        ; <HEX DIGIT> ::= 0!1!2!3!4!5!6!7!8!9!A!B!C!D!E!F
                        ;
000C                    VER   EQU     12              ; VERSION 1.2
                              TITLE   ' INTELLEC/MDS MONITOR, VERSION 1.2, 15 SEPTEMBER 1975 '
1509                    DATE  EQU     1509H           ; CREATION DATE, 15 SEPTEMBER 1975
                        ;
                        ; NOTE:
                        ;
                        ; THE DATE SHOWN ABOVE IS ENCODED INTO A TWO BYTE FIELD
                        ; IN BOTH THE BOOTSTRAP PROM AND THE MONITOR ROM IN ORDER
                        ; TO CONTROL NEW RELEASES OF THIS PROGRAM.
                        ;
                        ; IN THE BOOTSTRAP PROM, THE DATE CODE IS LOCATED AT
                        ; ADDRESSES 4 AND 5.
                        ;
                        ; IN THE MONITOR ROM, THE DATE CODE IS LOCATED AT ADDRESSES
                        ; 0F824H AND 0F825H.
                        ;
                        ; IF AND WHEN A NEW RELEASE IS ISSUED, PLEASE CHANGE THE
                        ; DATE CODE.
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ; INTELLEC/MDS SYSTEM CONSTANTS
                        ;
```

```
                ; TTY AND CRT STATUS BITS
                ;
0001            TRDY    EQU     00000001B       ; TRANSMIT READY
0002            RBR     EQU     00000010B       ; RECEIVE BUFFER READY
0004            TBE     EQU     00000100B       ; TRANSMIT EMPTY
0008            RPAR    EQU     00001000B       ; RECEIVE PARITY ERROR
0010            ROV     EQU     00010000B       ; RECEIVE OVERRUN ERROR
0020            RFR     EQU     00100000B       ; RECEIVE FRAMING ERROR
0080            DSR     EQU     10000000B       ; DATA SET READY
                ;
                ; TTY AND CRT INITIALIZATION CONTROLS
                ;
0002            R48@1   EQU     00000010B       ; 4800 BAUD @ JUMPER 1
0001            R96@1   EQU     00000001B       ; 9600 BAUD @ JUMPER 1
0003            R24@1   EQU     00000011B       ; 2400 BAUD @ JUMPER 1
0002            R6@2    EQU     00000010B       ; 600 BAUD @ JUMPER 2
0001            R12@2   EQU     00000001B       ; 1200 BAUD @ JUMPER 2
0003            R3@2    EQU     00000011B       ; 300 BAUD @ JUMPER 2
0002            R110    EQU     00000010B       ; 110 BAUD @ JUMPER 3
0008            CL7     EQU     00001000B       ; CHARACTER LENGTH = 7
000C            CL8     EQU     00001100B       ; CHARACTER LENGTH = 8
0004            CL6     EQU     00000100B       ; CHARACTER LENGTH = 6
0000            CL5     EQU     00000000B       ; CHARACTER LENGTH = 5
0040            ST1     EQU     01000000B       ; 1 STOP BIT
0080            ST15    EQU     10000000B       ; 1.5 STOP BITS
00C0            ST2     EQU     11000000B       ; 2 STOP BITS
0010            PENB    EQU     00010000B       ; PARITY ENABLE
0020            PEVEN   EQU     00100000B       ; EVEN PARITY
0001            TXEN    EQU     00000001B       ; TRANSMIT ENABLE
0002            DTR     EQU     00000010B       ; DATA TERMINAL READY
0004            RXEN    EQU     00000100B       ; RECEIVE ENABLE
0010            CLERR   EQU     00010000B       ; CLEAR ERROR
0040            USRST   EQU     01000000B       ; USART RESET
0020            RTS     EQU     00100000B       ; REQUEST TO SEND
                ;
                ; PTR, PTP, AND TTY READER CONTROLS
                ;
0010            PTPREV          EQU     00010000B       ; PUNCH REVERSE DIRECTION
0020            PTPADV          EQU     00100000B       ; PUNCH ADVANCE
0004            PTRREV          EQU     00000100B       ; READ REVERSE DIRECTION
0008            PTRADV          EQU     00001000B       ; READER ADVANCE
0002            TTYADV          EQU     00000010B       ; TTY ADVANCE
                ;
                ; LPT, PTR AND PTP STATUS BITS
                ;
0001            LPTRY   EQU     00000001B       ; LPT READY
0001            PTRDY   EQU     00000001B       ; PTR READY WITH DATA
0004            PTPRY   EQU     00000100B       ; PTP READY FOR DATA
                ;
                ; TTY I/O CONSTANTS
                ;
```

```
00F4            TTI   EQU     0F4H            ; TTY INPUT DATA PORT
00F4            TTO   EQU     0F4H            ; TTY OUTPUT DATA PORT
00F5            TTS   EQU     0F5H            ; TTY INPUT STATUS PORT
00F5            TTC   EQU     0F5H            ; TTY OUTPUT CONTROL PORT
                ;
                ; CRT I/O CONSTANTS
                ;
00F6            CRTI  EQU     0F6H            ; CRT INPUT DATA PORT
00F7            CRTS  EQU     0F7H            ; CRT INPUT STATUS PORT
00F6            CRTO  EQU     0F6H            ; CRT OUTPUT DATA PORT
00F7            CRTC  EQU     0F7H            ; CRT OUTPUT CONTROL PORT
                ;
                ; PTR I/O CONSTANTS
                ;
00F8            PTRI  EQU     0F8H            ; PTR INPUT DATA PORT
00F9            PTRS  EQU     0F9H            ; PTR INPUT STATUS PORT
00F9            PTRC  EQU     0F9H            ; PTR OUTPUT COMMAND PORT
                ;
                ; PTP I/O CONSTANTS
                ;
00F8            PTPO  EQU     0F8H            ; PTP OUTPUT DATA PORT
00F9            PTPS  EQU     0F9H            ; PTP INPUT STATUS PORT
00F9            PTPC  EQU     0F9H            ; PTP OUTPUT COMMAND PORT
                ;
                ; LPT I/O CONSTANTS
                ;
00FA            LPTO  EQU     0FAH            ; LPT OUTPUT DATA PORT
00FB            LPTS  EQU     0FBH            ; LPT INPUT STATUS PORT
00FB            LPTC  EQU     0FBH            ; LPT OUTPUT COMMAND PORT
                ;
                ; REAL TIME CLOCK CONSTANTS
                ; EACH TICK = 1.0 MS
                ;
00FF            RTC   EQU     0FFH            ; REAL TIME CLOCK PORT
0001            RTCS  EQU     00000001B       ; REAL TIME CLOCK STATUS
0002            BOOT  EQU     00000010B       ; BOOTSTRAP MODE INDICATOR, 1 = ON
                ;
                ; PROGRAMMER I/O CONSTANTS
                ;
00F1            PHI   EQU     0F1H            ; PROM COMMAND AND MSB ADDRESS BITS
00F2            PLO   EQU     0F2H            ; PROM ADDRESS BITS (8 LSB)
00F0            PDATA EQU     0F0H            ; PROM DATA PORT
00F1            PSTAT EQU     0F1H            ; PROM STATUS PORT
0002            PCOMP EQU     00000010B       ; PROGRAMMING COMPLETE
0001            PGRDY EQU     00000001B       ; PROM READY
0020            PSOCK EQU     00100000B       ; 16 PIN SOCKET SELECTED
0010            PNIB  EQU     00010000B       ; SELECT UPPER NIBBLE
                ;
                ; FDCC CONSTANTS
                ;
0079            LOW   EQU     79H             ; LOW(IOPB)
```

```
007A            HI      EQU     7AH             ; HIGH(IOPB)
0078            DSTAT   EQU     78H             ; DISK STATUS
3000            TRK0    EQU     3000H           ; FIRST ADDRESS OF DISK BOOTSTRAP
                ;
                ;       CONDITIONAL ASSEMBLY SWITCHES
                ;
0000            FALSE   EQU     0
FFFF            TRUE    EQU     NOT FALSE
0000            DEBUG   EQU     FALSE
                ;
                ; GLOBAL CONSTANTS
                ;
00FA            TOUT    EQU     250             ; 250 MS. COUNTER FOR READER TIMEOUT
000D            CR      EQU     0Dh             ; ASCII VALUE OF CARRIAGE RETURN
000A            LF      EQU     0AH             ; ASCII VALUE OF LINE FEED
0003            ETX     EQU     03H             ; MONITOR BREAK CHARACTER (CONTROL C)
                ;
                ; 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            LLPT    EQU     10000000B       ; LIST = LPT
00C0            LUSE    EQU     11000000B       ; USER DEFINED LIST
                ;
                ; INTERRUPT SYSTEM MASKS AND VALUES
                ;
0001            INT0    EQU     00000001B       ; MASK FOR INTERRUPT LEVEL 0
0002            INT1    EQU     00000010B
0004            INT2    EQU     00000100B
0008            INT3    EQU     00001000B
0010            INT4    EQU     00010000B
0020            INT5    EQU     00100000B
0040            INT6    EQU     01000000B
```

```
0080            INT7  EQU     10000000B
                ;
00FC            MASK  EQU     0FCH            ; MASK PORT
00FD            REVRT EQU     0FDH            ; INTERRUPT REVERT
00FE            LOCK  EQU     0FEH            ; BUS OVERRIDE
00FA            ISTAT EQU     0FAH            ; INTERRUPT STATUS PORT
00F3            ICON  EQU     0F3H            ; INTERRUPT CONTROL PORT
0020            EOI   EQU     00100000B       ; END OF INTERRUPT VALUE
                ;
                ; INTERRUPT STATUS AND CONTROL BITS
                ;
0001            ITTYO EQU     00000001B       ; OUTPUT TTY
0002            ITTYI EQU     00000010B       ; INPUT TTY
0004            IPTP  EQU     00000100B       ; PTP
0008            IPTR  EQU     00001000B       ; PTR
0010            ICRTO EQU     00010000B       ; OUTPUT CRT
0020            ICRTI EQU     00100000B       ; INPUT CRT
0040            ILPT  EQU     01000000B       ; LPT
0080            MENB  EQU     10000000B       ; ENABLE MONITOR INTERRUPTS
                ;
                ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                ;
                ; PAGE 0 DEDICATED RAM LOCATIONS, INITIALIZED BY SHADOW PROM CODE.
                ;
0000                    ORG     0
0000            RESET:
0003                    DS      3               ; TRAP TO MONITOR RESTART
0003            IOBYT:
0001                    DS      1               ; I/O SYSTEM STATUS BYTE
0004            MSK:                            ; PROM PROGRAMMING T/F FLAG
0004            MEMTOP:
0002                    DS      2               ; TOP OF RAM, ONLY H SAVED
0006            INITIO:
0001                    DS      1               ; INITIAL I/O CONFIGURATION
                ;
                ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                ;
                ;       MACRO DEFINITIONS
                ;
      1         FETCH MACRO   VALUE           ; FETCH AN ADDRESS IN THE STACK
      1               LXI     H,VALUE
      1               DAD     SP
                      ENDM
                ;
      1         GET   MACRO   VALUE           ; COMPUTE RAM ADDRESS BASED ON STACK
      1               LHLD    MEMTOP
      1               MVI     L,VALUE AND 0FFH
                      ENDM
                ;
```

```
1           WHILE MACRO   CHAR            ; SCAN INPUT WHILE EQUAL
1           LOOP:
1                 CALL    TI
1                 CPI     CHAR
1                 JZ      LOOP
                  ENDM
            ;
1           UNTIL MACRO   CHAR            ; SCAN INPUT UNTIL EQUAL
1           LOOP:
1                 CALL    TI
1                 CPI     CHAR
1                 JNZ     LOOP
                  ENDM
            ;
1           SIZE  MACRO                   ; FIND TOP OF MEMORY
1                 LXI     H,0             ; INITIAL VALUE
1           LOOP:
1                 INR     H
1                 MOV     A,M             ; FETCH CONTENTS OF MEMORY
1                 CMA                     ; INVERT IT
1                 MOV     M,A             ; ATTEMPT TO WRITE INTO MEMORY
1                 CMP     M               ; IS LOCATION READ/WRITE?
1                 CMA                     ; INVERT AGAIN
1                 MOV     M,A             ; WRITE DATA BACK
1                 JZ      LOOP            ; YES, CONTINUE
1                 DCX     H               ; LAST ADDRESS IN RAM
1                 SHLD    MEMTOP          ; STORE TOP OF MEMORY
                  ENDM
            ;
            ; CASE BRANCH MACRO
            ; INPUT PARAMETERS:
            ; REGISTER A - CASE INDEX, 0...N
            ; PARAMETER 1 - ADDRESS OF BRANCH TABLE
            ; PARAMETER 2 - LENGTH OF BRANCH TABLE
            ; USES REGISTERS A,D,E,H,L
            ;
1           CASE  MACRO   TABLE,RANGE
1                 LXI     H,TABLE
1                 CPI     RANGE           ; TEST FOR OVERRUN
1                 JP      ERROR
1                 MOV     E,A             ; MOVE INDEX TO DE
1                 MVI     D,0
1                 DAD     D               ; ADD BASE + 2 * INDEX -> HL
1                 DAD     D
1                 MOV     A,M             ; GET LSB OF BRANCH LOCATION
1                 INX     H
1                 MOV     H,M             ; GET MSB OF BRANCH LOCATION
1                 MOV     L,A
1                 PCHL                    ; TAKE THE BRANCH
                  ENDM
```

```
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ;       ADDRESS CONSTANTS FOR VERSION 1.2 ONLY
                        ;       ======= ========= === ======= === ====
                        ;
                        ;       THE FOLLOWING FOUR ADDRESSES ARE INSERTED FOR ASSEMBLING
                        ;       THE VERSION 1.2 BOOTSTRAP PROM CODE ONLY. THEY SPECIFY THE
                        ;       ADDRESSES OF THE THREE ROUTINES IN THE MONITOR ROM
                        ;       WHICH ARE CALLED FROM THE BOOTSTRAP PROM. THESE ARE THE
                        ;       ADDRESSES OF THESE ROUTINES IN THE VERSION 1.X MONITOR ROM.
                        ;
FF37                    RESTART         EQU     0FF37H
FD42                    TTYOUT          EQU     0FD42H
FD59                    CRTOUT          EQU     0FD59H
F800                    BEGIN           EQU     0F800H
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ; SHADOW PROM CODE
                        ;
0000                    SBASE SET       0
0000                          ORG       SBASE
                        ;
                        ; FUNCTIONS:
                        ;
                        ;       A.      INITIALIZE INTERRUPT SYSTEM.
                        ;               1. PROGRAM INTERRUPT SUBSYSTEM (8259)
                        ;               2. MASK ALL INTERRUPTS BUT TRAP LOGIC
                        ;
0000  C30600                  JMP       SH0             ; BRANCH AROUND STATUS BYTE
0003  00              INIT:  DB        0               ; INITIALLY,
                                                       ; CONSOLE = TTY,
                                                       ; READER = TTY,
                                                       ; PUNCH = TTY,
                                                       ; LIST = TTY
0004  0915                   DW        DATE            ; DATE STAMP FOR BOOTSTRAP PROM
0006                    SH0:
0006  F3                     DI                        ; DISABLE INTERRUPT SYSTEM
0007  3E12                   MVI       A,12H           ; INITIALIZE COMMAND
0009  D3FD                   OUT       REVRT
000B  AF                     XRA       A
000C  AF                     XRA       A
000D  D3FC                   OUT       MASK
000F  3EFE                   MVI       A,NOT INTO
0011  D3FC                   OUT       MASK
0013  3E00                   MVI       A,0
0015  D3F3                   OUT       ICON
                        ;
                        ;       B.      INITIALIZE RAM.
```

```
                        ;               1. COMPUTE SIZE OF RAM MEMORY.
                        ;
      1                 +       SIZE
0017  1 210000          +       LXI     H,0             ; INITIAL VALUE
001A  1                 +LOOP:
001A  1 24              +       INR     H
0018  1 7E              +       MOV     A,M             ; FETCH CONTENTS OF MEMORY
001C  1 2F              +       CMA                     ; INVERT IT
001D  1 77              +       MOV     M,A             ; ATTEMPT TO WRITE INTO MEMORY
001E  1 BE              +       CMP     M               ; IS LOCATION READ/WRITE?
001F  1 2F              +       CMA                     ; INVERT AGAIN
0020  1 77              +       MOV     M,A             ; WRITE DATA BACK
0021  1 CA1A00          +       JZ      LOOP            ; YES, CONTINUE
0024  1 2B              +       DCX     H               ; LAST ADDRESS IN RAM
0025  1 220400          +       SHLD    MEMTOP          ; STORE TOP OF MEMORY
                        ;
                        ;               2. SET UP DEDICATED MEMORY LOCATIONS
                        ;                       USER I/O ENTRY POINTS (TOP OF MEMORY)
                        ;                       EXIT TEMPLATE
                        ;                       USER REGISTERS
                        ;                       USER INTERRUPT MASK
                        ;                       USER STACK
                        ;                       MONITOR STACK
                        ;
0028    01C800                  LXI     B,TOS           ; MOVE EXIT TEMPLATE TO RAM
002B    69                      MOV     L,C
002C    F9                      SPHL                    ; SET STACK
002D                    SH1:
002D    0A                      LDAX    B
002E    77                      MOV     M,A
002F    0C                      INR     C               ; MOVE BOTH POINTERS
0030    2C                      INR     L
0031    C22D00                  JNZ     SH1             ; END ON PAGE BOUNDARY
0034    2ED1                    MVI     L,SLOC          ; SET UP INITIAL VALUE FOR USER STACK
0036    74                      MOV     M,H             ; LOWER HALF OF STACK POINTER IS KNOWN
0037    35                      DCR     M
                                                        ; MERELY SET UPPER HALF.
                        ;
                        ;                       TRAP TO MONITOR (0-2)
                        ;
0038    3EC3                    MVI     A,(JMP RESTART)
003A    320000                  STA     RESET
003D    213FFF                  LXI     H,RESTART       ; SET UP RESTART 0 FOR BREAKPOINT
0040    220100                  SHLD    RESET+1         ; LOGIC
                        ;
                        ;       C.      PROGRAM I/O DEVICES.
                        :               1. USART FOR CRT
                        ;               2. USART FOR TTY
                        ;
0043    3E4F                    MVI     A,ST1 OR B24@1 OR CL8
0045    D3F7                    OUT     CRTC
```

```
0047  3ECE          MVI   A,ST2 OR R110 OR CL8
0049  D3F5          OUT   TTC
004B  3E27          MVI   A,TXEN OR DTR OR RXEN OR RTS
004D  D3F7          OUT   CRTC
004F  D3F5          OUT   TTC
                    ;
                    ;
                    ;           3. HIGH SPEED READER
                    ;           4. HIGH SPEED PUNCH
                    ;           5. TTY READER
                    ;
0051  AF            XRA   A
0052  D3F9          OUT   PTRC
                    ;
                    ;     D.    LOAD ISIS.TO IF DISKETTE 0 IS READY
                    ;
0054  DB78          IN    DSTAT           ; SAMPLE FDCC STATUS
0056  0F            RRC
0057  D26800        JNC   SH3
005A  3EAA          MVI   A,IOPB
005C  D379          OUT   LOW             ; LOW(IOPB)
005E  AF            XRA   A
005F  D37A          OUT   HI              ; HIGH(IOPB), START DISK I/O
0061                SH2:
0061  DB78          IN    DSTAT           ; WAIT FOR FDCC TO COMPLETE
0063  E604          ANI   4
0065  CA6100        JZ    SH2
                    ;
                    ;     E.    DETERMINE COLD START CONSOLE.
                    ;
0068                SH3:
0068  210300        LXI   H,IOBYT         ; POINT AT I/O STATUS
006B  56            MOV   D,M             ; FETCH INTO D
006C  DBF5          IN    TTS
006E  E602          ANI   RBF
0070  CA7800        JZ    SH4             ; NOT TTY
0073  DBF4          IN    TTI             ; GET CHARACTER FROM TTY
0075  C38200        JMP   SH5
0078                SH4:
0078  14            INR   D               ; IOBYTE = CRT
0079  DBF7          IN    CRTS
007B  E602          ANI   RBR
007D  CA6800        JZ    SH3             ; NOT CRT
0080  DBF6          IN    CRTI            ; GET CHARACTER FROM CRT
0082                SH5:
0082  E67F          ANI   7FH
0084  FE20          CPI   ' '
0086  C26800        JNZ   SH3
0089  72            MOV   M,D             ; REPLACE MODIFIED I/O STATUS BYTE
008A  2E06          MVI   L,INITIO
008C  72            MOV   M,D             ; SET INITIAL I/O CONFIGURATION
```

```
                        ;
                        ;       F.      IF DISK IS READY, TRANSFER TO ISIS.TO
                        ;
008D    DB78            IN      DSTAT
008F    0F              RRC
0090    DA0030          JC      TRK0
                        ;
                        ;       G.      TYPE SIGN-ON FOR RAM MONITOR
                        ;
0093    2EB1            MVI     L,VERS          ; ADDRESS OF MESSAGE
0095    0615            MVI     B,LVER          ; LENGTH OF MESSAGE
0097                    SHo:
0097    4E              MOV     C,M             ; GET A CHARACTER
0098    7A              MOV     A,D             ; TEST CONSOLE SELECTION
0099    0F              RRC
009A    D442FD          CNC     TTYOUT          ; PRINT ON TTY IF TTY IS CONSOLE
009D    7A              MOV     A,D
009E    0F              RRC
009F    DC59FD          CC      CRTOUT          ; PRINT ON CRT IF CRT IS CONSOLE
00A2    23              INX     H
00A3    05              DCR     B
00A4    C29700          JNZ     SH6
                        ;
                        ;       H.      BRANCH TO MONITOR.
                        ;
00A7    C300F8          JMP     BEGIN           ; INTERRUPTS ARE DISABLED.
                        ;
                        ;       DISK I/O PARAMETER BLOCK
                        ;
00AA                    IOPB:
00AA    80              DB      80H             ; IOCW, NO UPDATE BIT SET
00AB    04              DB      04H             ; I/O INSTRUCTION, READ DISK 0
00AC    1A              DB      26              ; READ 26 SECTORS
00AD    00              DB      0               ; TRACK 0
00AE    01              DB      1               ; SECTOR 1
00AF    0030            DW      TRK0            ; LOAD ADDRESS
                        ;
                        ;       MDS MONITOR SIGN-ON MESSAGE
                        ;
00B1    0D0A4D44 VERS:  DB      CR,LF,'MDS MONITOR, V'
00B5    53204D4F
00B9    4E49544F
00BD    522C2056
00C1    312E32          DB      VER/10+'0','.',VER MOD 10+'0'
00C4    0D0A            DB      CR,LF
0015                    LVER    EQU     $-VERS          ; LENGTH OF SIGN-ON MESSAGE
                        ;
                        ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                        ;
                        ; EXIT CODE TEMPLATE, TO BE EXECUTED IN RAM
                        ; THIS CODE IS ORIGINATED SO AS TO BE ALIGNED
```

```
                        ; AGAINST THE TOP OF A PAGE
                        ;
00C8                            ORG     SBASE+0C8H
                        ;
00C8                    TOS:                            ; BASE OF MONITOR WORK STACK
00C0                    USER    EQU     TOS-8           ; BASE OF DEFAULT USER WORK STACK
00C8    EE              ELOC:   DB      0EEH            ; E REGISTER STORAGE
00C9    DD              DLOC:   DB      0DDH            ; D REGISTER
00CA    CC              CLOC:   DB      0CCH            ; C REGISTER
00CB    BB              BLOC:   DB      0BBH            ; B REGISTER
00CC    00                      DB      0               ; UNUSED BYTE
00CD    FE              ILOC:   DB      NOT INTO        ; INTERRUPT MASK
00CE    FF              FLOC:   DB      0FFH            ; CPU FLAGS
00CF    AA              ALOC:   DB      0AAH            ; A REGISTER
00D0    C0                      DB      0SEP            ; LOW(SP)
00D1    00              SLOC:   DB      0               ; HIGH(SP)
                        ;
00D2                    EXIT:                           ; MONITOR STACK ORIGIN
00D2    F3                      DI                      ; DISABLE TO PROTECT THIS SEQUENCE
00D3    D1                      POP     D               ; RESTORE D,E
00D4    C1                      POP     B               ; RESTORE B,C
00D5    F1                      POP     PSW             ; RESTORE INTERRUPT MASK
00D6    D3FC                    OUT     MASK
00D8    F1                      POP     PSW             ; RESTORE A AND FLAGS
00D9    E1                      POP     H               ; RESTORE ORIGINAL STACK VALUE
00DA    F9                      SPHL
00DB    213412                  LXI     H,1234H         ; RESTORE H,L
00DC                    LLOC    EQU     $-2
00DD                    HLOC    EQU     $-1
00DE    FB                      EI                      ; ENABLE INTERRUPTS
00DF    C36967                  JMP     6789H           ; RETURN TO INTERRUPTED CODE
00E1                    PLOC    EQU     $-1
00E2    0000            TLOC:   DW      0               ; TRAP 1 ADDRESS
00E4    00                      DB      0               ; TRAP 1 VALUE
00E5    0000                    DW      0               ; TRAP 2 ADDRESS
00E7    00                      DB      0               ; TRAP 2 VALUE
                        ;
                        ; EXTENSIBLE I/O ENTRY POINTS
                        ;
00E8                    XTBL:
00E8                    CILOC:
00E8    C30000                  JMP     0
00EB                    COLOC:
00EB    C30000                  JMP     0
00EE                    R1LOC:
00EE    C30000                  JMP     0
00F1                    P2LOC:
00F1    C30000                  JMP     0
00F4                    P1LOC:
00F4    C30000                  JMP     0
00F7                    P2LOC:
```

```
  00F7   C30000          JMP     0
  00FA                 L1LOC:
  00FA   C30000          JMP     0
  00FD                 CSLOC:
  00FD   C30000          JMP     0
  0100                 ENDX:                          ; THIS LABEL SHOULD BE AT 100H.
                       ;
                       ; END OF SHADOW PROM CODE
                       ;
                       ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                             END
NO PROGRAM ERRORS
```

SYMBOL TABLE

* 01

A	0007	ALOC	00CF *	B	0000	BATCH	0002 *
BEGIN	F800	BLOC	00CB *	BOOT	0002 *	C	0001
CASE	0F14 *	CCRT	0001 *	CILOC	00E8 *	CL5	0000 *
CL6	0004 *	CL7	0008 *	CL9	000C	CLERR	0010 *
CLOC	00CA *	CMSK	00FC *	COLOC	00EB *	CR	000D
CRTC	00F7	CRTI	00F6	CRTO	00F6 *	CRTOU	FD59
CRTS	00F7	CSLOC	00FD *	CTTY	0000 *	CUSE	0003 *
D	0002	DATE	1509	DEBUG	0000 *	DLOC	00C9 *
DSR	0080 *	DSTAT	0078	DTR	0002	E	0003
ELOC	00C8 *	ENDX	0100 *	EOI	0020 *	ETX	0003 *
EXIT	00D2 *	FALSE	0000	FETCH	0F9C *	FLOC	00CE *
GET	0F95 *	H	0004	HI	007A	HLOC	00DD *
ICON	00F3	ICRTI	0020 *	ICRTO	0010 *	ILOC	00CD *
ILPT	0040 *	INIT	0003 *	INITI	0006	INTO	0001
INT1	0002 *	INT2	0004 *	INT3	0008 *	INT4	0010 *
INT5	0020 *	INT6	0040 *	INT7	0080 *	IOBYT	0003
IOPB	00AA	IPTP	0004 *	IPTR	0008 *	ISTAT	00FA *
ITTYI	0002 *	ITTYO	0001 *	L	0005	L1LOC	00FA *
LCRT	0040 *	LF	000A	LLOC	00DC *	LLPT	0080 *
LMSK	003F *	LOCK	00FE *	LOW	0079	LPTC	00FB *
LPTO	00FA *	LPTRY	0001 *	LPTS	00FB *	LTTY	0000 *
LUSE	00C0 *	LVER	0015	M	0006	MASK	00FC
MEMTO	0004	MENB	0080 *	MSK	0004 *	P1LOC	00F4 *
P2LOC	00F7 *	PCOMP	0002 *	PDATA	00F0 *	PENB	0010 *
PEVEN	0020 *	PGRDY	0001 *	PHI	00F1 *	PLO	00F2 *
PLOC	00E1 *	PMSK	00CF *	PNIB	0010 *	PPTP	0010 *
PSOCK	0020 *	PSTAT	00F1 *	PSW	0006	PTPAD	0020 *
PTPC	00F9 *	PTPO	00F8 *	PTPRE	0010 *	PTPRY	0004 *
PTPS	00F9 *	PTRAD	0008 *	PTRC	00F9	PTRDY	0001 *
PTRI	00F8 *	PTRRE	0004 *	PTRS	00F9 *	PTTY	0000 *
PUSE1	0020 *	PUSE2	0030 *	R110	0002	R12@2	0001 *
R1LOC	00EE *	P24@1	0003	R2LOC	00F1 *	R3@2	0003 *
R48@1	0002 *	R6@2	0002 *	R96@1	0001 *	RBR	0002
RESET	0000	RESTA	FF37	REVRT	00FD	RFR	0020 *
RMSK	00F3 *	ROV	0010 *	RPAR	0008 *	RPTR	0004 *
RTC	00FF *	RTCS	0001 *	RTS	0020	RTTY	0000 *
RUSE1	0008 *	RUSE2	000C *	RXEN	0004	SBASE	0000
SH0	0006	SH1	002D	SH2	0061	SH3	0068
SH4	0078	SH5	0082	SH6	0097	SIZE	0F47
SLOC	00D1	SP	0006	ST1	0040	ST15	0080 *
ST2	00C0	TBE	0004 *	TLOC	00E2 *	TOS	00C8
TOUT	00FA *	TRDY	0001 *	TRK0	3000	TRUE	FFFF *
TTC	00F5	TTI	00F4	TTO	00F4 *	TTS	00F5
TTYAD	0002 *	TTYOU	FD42	TXEN	0001	UNTIL	0F86 *
USER	00C0	USRST	0040 *	VER	000C	VERS	00B1
WHILE	0F8E *	XTBL	00E8 *				

* 02

LOOP	001A

* 09

* 10

* 11

* 12

* 13

* 14

* 15

* 16

* 17

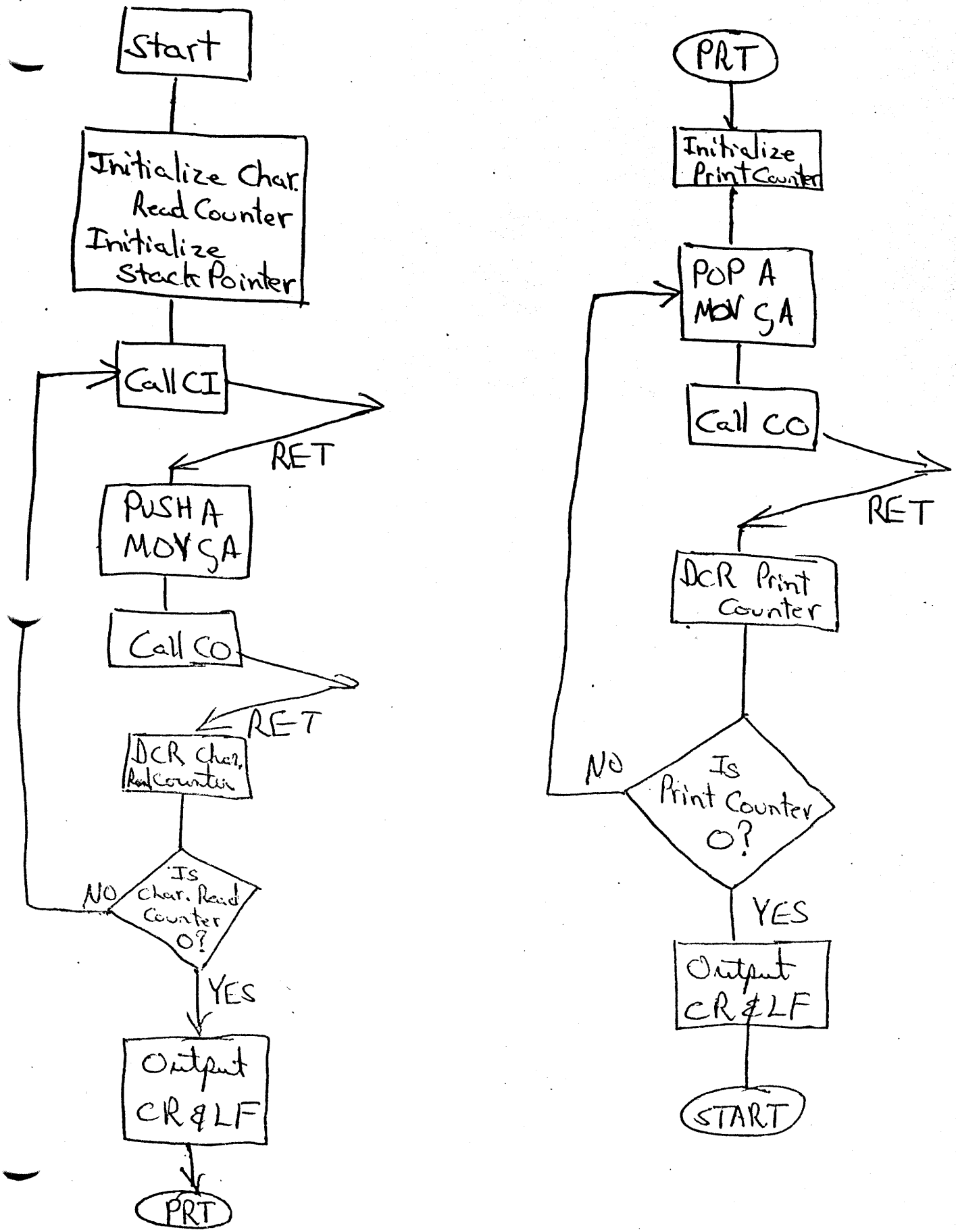
* 18

```
0020                ORG     20H
0020 DB00     LOOP: IN      0H
0022 4F             MOV     C,A
0023 DB01           IN      01H
0025 81             ADD     C
0026 D300           OUT     0H
0028 91             SUB     C
0029 91             SUB     C
002A D301           OUT     01H
002C C32000         JMP     LOOP
0020                END     LOOP
```

PROGRAM TO INPUT AN 8 BIT SWITCH SETTING FROM INPUT PORTS 1 & 2 AND DISPLAY THE SUM AND DIFFERENCE AT OUTPUT PORTS 1 & 2. 6/9/76

PROGRAM TO ACCEPT A WORD FROM THE TELETYPE (AND ECHO IT) AND THEN PRINT THE SAME WORD BACKWARDS. 6/9/76

```
50T
$S
        ORG     50H
CO      EQU     0F809H
CI      EQU     0F803H
START:  MVI     D,06H
        LXI     SP,0200H
READ:   CALL    CI
        PUSH    PSW
        MOV     C,A
        CALL    CO
        DCR     D
        JNZ     READ
        MVI     C,0DH
        CALL    CO
        MVI     C,0AH
        CALL    CO
        MVI     E,06H
PRINT:  POP     PSW
        MOV     C,A
        CALL    CO
        DCR     E
        JNZ     PRINT
        MVI     C,0DH
        CALL    CO
        MVI     C,0AH
        CALL    CO
        JMP     START
        END     50H
*
```


Start
Initialize Char. Read Counter
Initialize Stack Pointer
Call CI
RET
PUSH A
MOV C,A
Call CO
RET
DCR Char. Read Counter
Is Char. Read Counter 0?
NO
YES
Output CR&LF
PRT
PRT
Initialize Print Counter
POP A
MOV C,A
Call CO
RET
DCR Print Counter
Is Print Counter 0?
NO
YES
Output CR&LF
START

8080 MDS MACRO ASSEMBLER, V2.2 PAGE 1

```
                        0050            ORG     50H
                        F809            CO      EQU     0F809H
                        F803            CI      EQU     0F803H
                        0050 1606       START:  MVI     D,06H
                        0052 310002             LXI     SP,0200H
                        0055 CD03F8     READ:   CALL    CI
                        0058 F5                 PUSH    PSW
                        0059 4F                 MOV     C,A
                        005A CD09F8             CALL    CO
                        005D 15                 DCR     D
                        005E C25500             JNZ     READ
                        0061 0E0D               MVI     C,0DH
                        0063 CD09F8             CALL    CO
                        0066 0E0A               MVI     C,0AH
                        0068 CD09F8             CALL    CO
                        006B 1E06               MVI     E,06H
                        006D F1         PRINT:  POP     PSW
                        006E 4F                 MOV     C,A
                        006F CD09F8             CALL    CO
                        0072 1D                 DCR     E
                        0073 C26D00             JNZ     PRINT
                      I 0076 0E0D               MVI     C,0DH
                        0078 CD09F8             CALL    CO
                        007B 0E0A               MVI     C,0AH
                        007D CD09F8             CALL    CO
                        0080 C35000             JMP     START
                        0050                    END     50H
```