```
                ;*****************************************************************
                ;*                                                               *
                ;*       CUSTOMIZED  BIOS FOR MICROMATION 'DOUBLER' AS USED WITH *
                ;*                (C) OR (C1) PROM (ON DOUBLER).                 *
                ;*           SINGLE/DOUBLE DENSITY, SINGLE/DOUBLE SIDED.         *
                ;*===============================================================*
                ;* WITH DRIVERS FOR MM MULTI I/O BRD IN SINGLE USER CONFIGURATION*
                ;*===============================================================*
                ;*       COPYRIGHT (C) 1979, MICROMATION AND DIGITAL RESEARCH.   *
                ;*                                                               *
                ;*       LAST UPDATED ON FEB 12, 1980                            *
                ;*                                                               *
                ;*       REVISED FOR CAVRO STD SYSTEM                            *
                ;*       VERSION 2.0  REV 8  OCT 26,1984                         *
                ;*                                                               *
                ;*****************************************************************

                ;*******************************************************************************
0000 =          TRUE            EQU     -1      !       FALSE           EQU     0
                ;*******************************************************************************
                ;*THE FOLLOWING LABELS MAY BE USER DEFINED FOR DIFFERENT OPERATING ENVIRONMENTS:

0038 =          MSIZE           EQU     56      ;SIZE OF OPERATING SYSTEM IN KILOBYTES
                                                ;(CURRENTLY 60K).  THIS NUMBER MUST BE
                                                ;CHANGED FOR LARGER SYSTEMS.

F000 =          IOPROM          EQU     0F000H

                                                ;OFFSET FOR 56K = 4580H


0004 =          NDRIVES         EQU     4       ;NUMBER OF DISK DRIVES SUPPORTED BY
                                                ;THIS CBIOS

                ;*******************************************************************************
                ;       I/O BYTE FOR LIST DEVICE IS IMPLEMENTED AS FOLLOWS:
                ;               TTY = MULTLIST (MM MULTI I/O BOARD SERIAL PORT#1)
                ;               CRT = CONOUT   (MM DOUBLER SERIAL PORT)
                ;               LPT = CENTLIST (CENTRONICS 703/779 TYPE LIST DEVICE)
                ;               UL1 = OPTIONAL DRIVER TO BE SELECTED BY USER (SEE BELOW)
                ;*******************************************************************************
                ;                                                       ==>UPDATED:2-12-80
                ;
                ;       LIST DEVICE EQUATES: (THESE COULD BE SET FALSE TO SAVE BIOS SPACE IF
                ;                             USER LIST DRIVER IS TOO LARGE TO FIT OTHERWISE)
                ;
0000 =          MULTLIST        EQU     FALSE
0000 =          CENTLIST        EQU     FALSE
                ;-------------------------------------------------------------------------------
                ;       LIST DEVICE OPTIONS: SET ONLY ONE FLAG TRUE FOR DESIRED DRIVER AS UL1
                ;
0000 =          NONE            EQU     FALSE   ;NO UL1 FUNCTION DESIRED
0000 =          GODBIO          EQU     FALSE   ;GODBOUT I/O BOARD AS LST:
0000 =          SSMIO           EQU     FALSE   ;SOLID STATE MUSIC 2S+P AS LST:
```

```
0000 =          DPIO            EQU     FALSE   ;DELTA PRODUCTS CPU BOARD AS LST:
FFFF =          USERLST         EQU     TRUE    ;SET FLAG TO INSERT USER DEFINED LST:
                                                ;THIS CODE MUST BE INSERTED IN BIOS UNDER LIST:
                                                ;AND INITIALIZATION CODE UNDER (COLD) BOOT
                ;-----------------------------------------------------------------------
                ;
                ;       BIAS IS ADDRESS OFFSET FROM 3400H FOR MEMORY SYSTEMS
                ;       THAN 20K (REFERRED TO AS B THROUGHOUT THE TEXT).
                ;
9000 =          BIAS    EQU     (MSIZE-20)*1024
C400 =          CCP     EQU     3400H+BIAS      ;BASE OF CCP
CC06 =          BDOS    EQU     CCP+806H        ;BASE OF BDOS
DA00 =          BIOS    EQU     CCP+1600H       ;BASE OF BIOS
                ;
0003 =          IOBYTE  EQU     0003H   ;INTEL I/O BYTE
005E =          SIGNON  EQU     005EH   ;LOCATION OF MESSAGE LEFT BY
                                        ;BOOTSTRAP LOADER


                ;       CBIOS FOR MICROMATION DOUBLE DENSITY CONTROLLER
                ;

F800 =          CONTROLLER      EQU     0F800H          ;ADDRESS OF CONTROLLER
F800 =          PROM            EQU     CONTROLLER      ;ADDRESS OF DOUBLER PROM
FC00 =          BUFF            EQU     CONTROLLER+400H ;ADDRESS OF SCRATCH RAM


                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;       *                                                       *
                ;       *               HARDWARE PORT DEFINITIONS               *
                ;       *                                                       *
                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;

                ;
FE00 =          WRCONT          EQU     CONTROLLER+600H
FE00 =          RDSTAT          EQU     WRCONT
FE01 =          WRCLK           EQU     WRCONT+1
FE02 =          UARTDATA        EQU     WRCONT+2
FE05 =          RDMARK          EQU     WRCONT+5        ;LOADS THE HEAD
FE0A =          UARTSTAT        EQU     WRCONT+0AH

                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;       *                                                       *
                ;       *               RAM VARIABLE DEFINITIONS                *
                ;       *                                                       *
                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;
                ;
FC01 =          DENBYTE         EQU     BUFF+1          ;0 FOR SINGLDE DEN., 10 FOR DBL.DEN
FC03 =          CTRBYTE         EQU     BUFF+3          ;RAM IMAGE OF RDSTAT OR WRCONT
FC04 =          TRACK           EQU     BUFF+4          ;TRACK NUMBER FOR CURRENT DRIVE
FC05 =          PRESDSK         EQU     BUFF+5          ;CURRENTLY LOGGED IN DRIVE NO.
FC0A =          SECTOR          EQU     BUFF+0AH
FC0D =          NXTDISK         EQU     BUFF+0DH        ;DRIVE NO. FOR NEXT I/O (READ/WRITE)
FC10 =          STEPTIME        EQU     BUFF+10H        ;STEP TIME IN MSEC
```

```
FC16 =          DENMAP          EQU     BUFF+16H        ;FOR EACH OF 4 DRIVES,
                                                        ;00 FOR SINGLE DENSITY
                                                        ;10H FOR (C) PROM
                                                        ;04H FOR (C1)PROM FOR DOUBLE DENSITY
FC20 =          TRY1            EQU     BUFF+20H        ;CBIOS ERROR CHECK
FC21 =          RETRYCOUNT      EQU     BUFF+21H        ;CBIOS TRK./SECTOR ERROR CHK. BYTE
FC22 =          CURRDRIVE       EQU     BUFF+22H        ;CBIOS DISK, LOGGED IN BEFORE WARMBOOT

                        ****************************************
                        *                                      *
                        *        LIST DEVICE EQUATES           *
                        *                                      *
                        ****************************************

                IF (MULTLIST OR CENTLIST)
                ;ASSEMBLED ONLY FOR PORTS ON MICROMATION MULTI I/O BOARD

        *=======================================================================
        *
        * PORT EQUATES:CHANGE VALUE OF UART1C TO RE-MAP THE PORT DECODING OF I/O BOARD
        *
        *       NOTE:DECODING OF BOARD I/O SPACE OCCURS ALONG 32 PORT BOUNDRIES
        *            (I.E.:A5-A7 ARE SIGNIFICANT ADDRESS BITS FOR I/O BLOCK)
        *=======================================================================

        ***************************************************************************

        UART1C  EQU     0               ;CONTROL PORT OF 8251 #1 (IC 4B)

        ***************************************************************************

        BASE    EQU     UART1C          ;EQUATE BASE FOR REST OF TABLE
        UART1D  EQU     BASE+1          ;DATA PORT OF UART #1

        UART2C  EQU     BASE+4          ;UART #2 (IC 5B)
        UART2D  EQU     BASE+5

        UART3C  EQU     BASE+8          ;UART #3 (IC 7B)
        UART3D  EQU     BASE+9

        UART4C  EQU     BASE+0CH        ;UART #4 (IC 9B)
        UART4D  EQU     BASE+0DH

        P8255   EQU     BASE+10H        ;8255 CONTROL REG. (IC 1A)
        PORTC   EQU     BASE+11H
        PORTB   EQU     BASE+12H
        PORTA   EQU     BASE+13H

        RTC     EQU     BASE+14H        ;8253 TIMER/COUNTER CONTROL PORT (IC 3B)
        TIMER2  EQU     BASE+15H        ;COUNTER #2
        TIMER1  EQU     BASE+16H        ;COUNTER #1
        TIMER0  EQU     BASE+17H        ;COUNTER #0

        PORT0   EQU     BASE+18H        ;PARALLEL OUTPUT LATCH (IC 2A)
        PORT1   EQU     BASE+19H        ;PARALLEL OUTPUT LATCH (IC 3A)
```

```
                USERPRT EQU     BASE+1CH        ;BASE OF USER DEFINED I/O PORTS (PORTS 1C-1FH)

                ********************************************************************************
                        ENDIF                   ;STOP CONDITIONAL ASSEMBLY


                        IF CENTLIST
                        ;CONDITIONAL ASM FOR MM MULTI I/O BRD FOR USE WITH CENRONICS PRINTER

                COMMAND         EQU     P8255           ;COMMAND WORD REGISTER
                DATAPORT        EQU     PORT0           ;PARALLEL DATA OUTPUT LATCH TO PRINTER
                STATUS          EQU     PORTC           ;STATUS BITS:OBF/=BIT 7
                                                        ;            ACK/=BIT 6
                                                        ;            BUSY=BIT 5
                                ENDIF                   ;END OF PORT DEFINITIONS FOR CENTLIST


                        IF USERLST      ;AREA FOR USER LIST ROUTINE EQUATES

0020 =          TUART   EQU     20H
0020 =          TUARTAS EQU     TUART
0021 =          TUARTA  EQU     TUARTAS+1
0022 =          TUARTAC EQU     TUARTA+1
0023 =          TUARTAM EQU     TUARTAC+1
0030 =          TUARTBS EQU     TUART+10H
0031 =          TUARTB  EQU     TUARTBS+1
0032 =          TUARTBC EQU     TUARTB+1
0033 =          TUARTBM EQU     TUARTBC+1
0080 =          TUARTBE EQU     80H
0040 =          TUARTDA EQU     40H

                        ENDIF

                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;       *                                                               *
                ;       *       JUMP VECTORS TO ROUTINES IN CONTROLLER PROM             *
                ;       *                                                               *
                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;
                ;
F803 =          CHOME           EQU     PROM+3          ;HOMES THE DISK
F806 =          CSELDSK         EQU     PROM+6          ;SELECTS DRIVE POINTED TO BY C REG
                                                        ;AND LOADS HEAD
F809 =          CSETTRK         EQU     PROM+9          ;STEPS DRIVE TO TRACK (C)
F80C =          CSETSEC         EQU     PROM+0CH        ;SET SECTOR NUMBER
F80F =          CSETDMA         EQU     PROM+0FH        ;SET DMA ADDRESS
F815 =          DISKWRITE       EQU     PROM+15H        ;WRITE SECTOR
F812 =          DISKREAD        EQU     PROM+12H        ;SECTOR READ
F81B =          SETDEN          EQU     PROM+1BH        ;TEST DENSITY OF CURRENT DRIVE
F81E =          WRITEPROTECT            EQU     PROM+1EH        ;CHECK FOR WRITE PROTECT

                ;
                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;
DA00                            ORG     BIOS    ;ORIGIN OF THIS PROGRAM
                ;
```

```
                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;
                ;
002C =          NSECTS          EQU     ($-CCP)/128     ;WARM START SECTOR COUNT
                ;
0000 =          BOOTBASE        EQU     0               ;BOTTOM OF USABLE MEMORY
0004 =          CPMDRIVE        EQU     BOOTBASE+4      ;LOC. 04 OF PAGE 00. CURRENT CPM DISK
0042 =          TRACK1          EQU     BOOTBASE+42H    ;TEMPORARY STORAGE FOR TRACK NUMBER
0049 =          DENSIDE         EQU     BOOTBASE+49H    ;INFO ON DENSITY AND SIDE FOR UTILITIES
004D =          SIDE            EQU     BOOTBASE+4DH
                ;
                ;
                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;       *                                                       *
                ;       *       JUMP VECTORS FOR INDIVIDUAL SUBROUTINES         *
                ;       *               USED BY ENTIRE SYSTEM                   *
                ;       *                                                       *
                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

DA00 C3CADA             JMP     BOOT            ;COLD START
DA03 C3F0DA     BWBOOT  JMP     WBOOT           ;WARM START
DA06 C31BDD             JMP     CONST           ;CONSOLE STATUS
DA09 C324DD     CONIN1  JMP     CONIN           ;CONSOLE CHARACTER IN
DA0C C330DD     CONOUT1 JMP     CONOUT          ;CONSOLE CHARACTER OUT
DA0F C33DDD             JMP     LIST            ;LIST CHARACTER OUT
DA12 C362DD             JMP     PUNCH           ;PUNCH CHARACTER OUT
DA15 C362DD             JMP     READER          ;READER CHARACTER OUT
DA18 C3DDDC             JMP     HOME            ;BIOS MOVE HEAD TO HOME POSITION
DA1B C37EDC             JMP     SELDSK          ;BIOS SELECT DISK (DOESN'T ACTUALLY SELECTS)
DA1E C3DFDC             JMP     SETTRK          ;BIOS SET TRACK
DA21 C3FBDC             JMP     SETSEC          ;BIOS SET SECTOR
DA24 C30FF8             JMP     CSETDMA         ;CONTROLLER SET DMA ADDRESS
DA27 C3C7DB             JMP     READ            ;BIOS READ DISK
DA2A C3A8DB             JMP     WRITE           ;BIOS WRITE DISK
DA2D C361DD             JMP     LISTST          ;RETURN LIST STATUS
DA30 C3E4DC             JMP     SECTRAN         ;SECTOR TRANSLATE
                ;
                ;
                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;       *                                                       *
                ;       *       FIXED DISK PARAMETER BLOCK TABLES FOR FOUR      *
                ;       *                DRIVE SYSTEM                           *
                ;       *                                                       *
                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;
                ;
DA33 =          DPBASE  EQU     $       ;BASE OF DISK PARAMETER BLOCKS
                ;
                ;       DISK PARAMETER BLOCK FOR DRIVE NO. 00
                ;
DA33 73DA0000   DPE0:   DW      TRANS,0000H     ;TRANSLATE TABLE
DA37 00000000           DW      0000H,0000H     ;SCRATCH AREA
DA3B 6FDD8DDA           DW      DIRBUF,DPB0     ;DIR BUFF, PARM BLOCK
DA3F 0EDEEFDD           DW      CSV0,ALV0       ;CHECK, ALLOC VECTORS
                ;
                ;       DISK PARAMETER BLOCK FOR DRIVE NO. 01
```

```
                        ;
DA43 73DA0000   DPE1:   DW      TRANS,0000H     ;TRANSLATE TABLE
DA47 00000000           DW      0000H,0000H     ;SCRATCH AREA
DA4B 6FDD8DDA           DW      DIRBUF,DPB0     ;DIR BUFF, PARM BLOCK
DA4F 4DDE2EDE           DW      CSV1,ALV1       ;CHECK, ALLOC VECTORS
                        ;
                        ;       DISK PARAMETER BLOCK FOR DRIVE NO. 02
                        ;
DA53 73DA0000   DPE2:   DW      TRANS,0000H     ;TRANSLATE TABLE
DA57 00000000           DW      0000H,0000H     ;SCRATCH AREA
DA5B 6FDD8DDA           DW      DIRBUF,DPB0     ;DIR BUFF, PARM BLOCK
DA5F 8CDE6DDE           DW      CSV2,ALV2       ;CHECK, ALLOC VECTORS
                        ;
                        ;       DISK PARAMETER BLOCK FOR DRIVE NO. 03
                        ;
DA63 73DA0000   DPE3:   DW      TRANS,0000H     ;TRANSLATE TABLE
DA67 00000000           DW      0000H,0000H     ;SCRATCH AREA
DA6B 6FDD8DDA           DW      DIRBUF,DPB0     ;DIR BUFF, PARM BLOCK
DA6F CBDEACDE           DW      CSV3,ALV3       ;CHECK, ALLOC VECTORS
                        ;
                        ;
                        ;       SINGLE DENSITY SECTOR TRANSLATE TABLE
                        ;
                        ;
DA73 01070D13   TRANS:  DB      1,7,13,19       ;SECTORS 1,2,3,4
DA77 19050B11           DB      25,5,11,17      ;SECTORS 5,6,7,8
DA7B 1703090F           DB      23,3,9,15       ;SECTORS 9,10,11,12
DA7F 1502080E           DB      21,2,8,14       ;SECTORS 13,14,15,16
DA83 141A060C           DB      20,26,6,12      ;SECTORS 17,18,19,20
DA87 1218040A           DB      18,24,4,10      ;SECTORS 21,22,23,24
DA8B 1016               DB      16,22           ;SECTORS 25,26
                        ;
                        ;
                                ;DISK PARAMETER BLOCK FOR SINGLE DENSITY SINGLE SIDED DISKS
                        ;               WITH BLOCK SIZE BLKSZ = 1024 BYTES / BLOCK
                        DPB0:
                        ;
DA8D 1A00               DW      26              ;SECTORS PER TRACK
DA8F 03                 DB      3               ;BLOCK SHIFT FACTOR
DA90 07                 DB      7               ;BLOCK MASK
DA91 00                 DB      0               ;NULL MASK
DA92 F200               DW      242             ;DISK SIZE-1 (NO. OF BLOCKS/DISK-1)
DA94 3F00               DW      63              ;NO. OF DIRECTORY ENTRIES MAX.-1
DA96 C0                 DB      192             ;DIRECTORY ALOCATION SPACE MASK, 1 ST BYTE
DA97 00                 DB      0               ;SAME AS ABOVE, 2 ND BYTE
DA98 1000               DW      16              ;CHECK SIZE - (64 DIR ENTRIES DIV BY 4)
DA9A 0200               DW      2               ;NO. OF SYSTEM (NOT ACCESSABLE) TRACKS
                        ;
                        DPB1:           ;DISK PARAMETER BLOCK FOR DOUBLE DEN., SINGLE SIDED DISKS
                        ;               WITH BLOCK SIZE BLKSZ = 2048 BYTES / BLOCK
                        ;
DA9C 3400               DW      52              ;SECTORS PER TRACK
DA9E 04                 DB      4               ;BLOCK SHIFT FACTOR
DA9F 0F                 DB      15              ;BLOCK MASK
DAA0 01                 DB      1               ;EXTENT MASK
DAA1 F200               DW      242             ;DISK SIZE-1 (NO. OF BLOCKS/DISK-1)
```

```
DAA3 7F00           DW      127             ;NO. OF DIRECTORY ENTRIES MAX.-1
DAA5 C0             DB      192             ;DIRECTORY ALOCATION SPACE MASK, 1 ST BYTE
DAA6 00             DB      0               ;SAME AS ABOVE, 2 ND BYTE
DAA7 2000           DW      32              ;CHECK SIZE
DAA9 0200           DW      2               ;NO. OF SYSTEM (NOT ACCESSABLE) TRACKS
                ;
                DPB2:       ;DISK PARAMETER BLOCK FOR SINGLE DEN., DOUBLE SIDED DISKS
                ;                   WITH BLOCK SIZE BLKSZ = 2048 BYTES / BLOCK
                ;
DAAB 3400           DW      52              ;SECTORS PER TRACK
DAAD 04             DB      4               ;BLOCK SHIFT FACTOR
DAAE 0F             DB      15              ;BLOCK MASK
DAAF 01             DB      1               ;EXTENT MASK
DAB0 F200           DW      242             ;DISK SIZE-1 (NO. OF BLOCKS/DISK-1)
DAB2 7F00           DW      127             ;NO. OF DIRECTORY ENTRIES MAX.-1
DAB4 C0             DB      192             ;DIRECTORY ALOCATION SPACE MASK, 1 ST BYTE
DAB5 00             DB      0               ;SAME AS ABOVE, 2 ND BYTE
DAB6 2000           DW      32              ;CHECK SIZE
DAB8 0200           DW      2               ;NO. OF SYSTEM (NOT ACCESSABLE) TRACKS
                ;
                DPB3:       ;DISK PARAMETER BLOCK FOR DOUBLE DEN., DOUBLE SIDED DISKS
                ;                   WITH BLOCK SIZE BLKSZ = 4096 BYTES / BLOCK

DABA 6800           DW      104             ;SECTORS PER TRACK
DABC 05             DB      5               ;BLOCK SHIFT FACTOR
DABD 1F             DB      31              ;BLOCK MASK
DABE 03             DB      3               ;EXTENT MASK
DABF F200           DW      242             ;DISK SIZE-1 (NO. OF BLOCKS/DISK-1)
DAC1 7F00           DW      127             ;NO. OF DIRECTORY ENTRIES MAX.-1
DAC3 80             DB      128             ;DIRECTORY ALOCATION SPACE MASK, 1 ST BYTE
DAC4 00             DB      0               ;SAME AS ABOVE, 2 ND BYTE
DAC5 2000           DW      32              ;CHECK SIZE
DAC7 0200           DW      2               ;NO. OF SYSTEM (NOT ACCESSABLE) TRACKS
                ;
                ;
                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;       *                                                       *
                ;       *               END OF FIXED TABLES                     *
                ;       *                                                       *
                ;       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                ;
                ;
                PAGE
```

```
                ;       ********************************************************
                ;       *       *       *       *       *       *       *      *
                ;       ********************************************************
                ;
DAC9 FF         BOOTFLAG:       DB      0FFH    ;SET TO 00 ON COLDBOOT


                BOOT:   ;SIMPLEST CASE IS TO JUST PERFORM PARAMETER INITIALIZATION

DACA 3EC1               MVI     A,0C1H          ;SET IOBYTE TO: LST:=UL1:
                                                ;               CON:=CRT:
DACC 320300             STA     IOBYTE          ;INITIALIZE THE IOBYTE
                                                ;(LST:=LPT:, CON:=CRT:)
DACF AF                 XRA     A               ;ZERO IN THE ACCUM
DAD0 320400             STA     CPMDRIVE        ;LOC 0004 - CPM CURRENT DISK NO.
DAD3 32C9DA             STA     BOOTFLAG

                ;-------------------------------------------------------------------------
                ;       INITIALIZATION ROUTINES FOR LIST DEVICES FOLLOW:
                ;-------------------------------------------------------------------------
                ;       IF GODBIO               ;START CONDITIONAL FOR GODBOUT I/O BOARD
                ;       XRA     A               ;INITIALIZE THE I/O BOARD
                ;       OUT     3               ;BY SENDING A 00 TO THE STATUS PORT
                ;       ENDIF                   ;STOP ASSEMBLING FOR THE GODBOUT ONLY
                ;ABOVE NOT NEEDED FOR CURRENT BOARD

                        IF DPIO                 ;UART INIT FOR DELTA PRODUCTS CPU BOARD
                        MVI     A,0EH           ;BRING UART TO COMAND INSTRUCTION MODE
                        OUT     01              ;PORT 1 IS UART A STATUS
                        OUT     03              ;PORT 3 IS UART B STATUS (ALSO TIME FILL)
                        MVI     A,40H           ;COMAND INSTRUCTION MODE - RESET UART
                        OUT     01
                        OUT     03
                        MVI     A,0EEH          ;MODE - BAUD RATE FACTOR = 16X
                        OUT     01              ;DISABLE PARITY, 8 BIT CHARACTER LENGTH
                        OUT     03              ;
                        MVI     A,37H           ;COMAND INSTRUCTION MODE-TRANSM ENABLE,
                        OUT     01              ;DATA TERM READY, RECEIVE ENABLE, ERROR
                        OUT     03              ;RESET, REQUEST TO SEND (IF NEEDED)
                        ENDIF                   ;STOP ASSEMBLING FOR THE DELTA PRODUCTS BRD.

                        IF MULTLIST
                        ;CONDITIONAL ASSEMBLY FOR MICROMATION MULTI I/O BOARD FOLLOWS

                        ;8251 UART INITIALIATION:
                        MVI     A,0EH           ;THESE FIRST TWO BYTES PUT 8251'S IN MODE SET
                        OUT     UART1C          ;
                        NOP                     ;WASTE TIME AT 4MHZ....
                        NOP
                        MVI     A,40H           ;
                        OUT     UART1C          ;
                        NOP                     ;WASTE TIME AT 4MHZ....
                        NOP
                        MVI     A,0EEH          ;MODE SET:ASYNC,8 DATA & 2 STOP BITS,NO PARITY
```

```
                        OUT     UART1C          ;
                        NOP                     ;WASTE TIME AT 4MHZ....
                        NOP
                        MVI     A,37H           ;COMMAND WORD:ENABLE RXD & TXD, RTS/ & DTR/
                                                ;FLAGS SET, ERROR FLAGS RESET...
                        OUT     UART1C          ;

                        ;8253 INITIALIZATION:
                        MVI     A,0B6H          ;SET MODE OF 8253 COUNTER #2 (BAUD CLOCK)
                        OUT     RTC             ;
                        MVI     A,13            ;DIVIDE BY 13 OF MASTER 2MHZ. CLOCK
                        OUT     TIMER2          ;
                        XRA     A               ;
                        OUT     TIMER2          ;
                        ENDIF                   ;END OF CONDITIONAL ASSEMBLY


                        IF CENTLIST
                        ;8255 INITIALIZATION: SETS UP CENTRONICS TYPE PRINTER DRIVER

                        MVI     A,0A8H          ;DEFINE MODE: PORTA= MODE1 (OUTPUT)
                                                ;PORTC= STATUS A AND MODE0 (INPUT)
                                                ;PORTB= DON'T CARE (SET TO MODE0 OUTPUT)
                        OUT     COMMAND         ;
                        MVI     A,0FFH          ;TURN OFF DATA STROBE
                        OUT     DATAPORT        ;
                        ENDIF                   ;STOP CONDITIONAL ASSEMBLY FOR CENTLIST


                        IF USERLST    ;INSERT CUSTOM LIST DEVICE INITIALIZATION IN THIS AREA

DAD6 AF                 XRA     A
DAD7 D323               OUT     TUARTAM
DAD9 D333               OUT     TUARTBM
DADB 3E01               MVI     A,01
DADD D322               OUT     TUARTAC
DADF D332               OUT     TUARTBC
DAE1 3E10               MVI     A,10H
DAE3 D322               OUT     TUARTAC
DAE5 3E90               MVI     A,90H
DAE7 D320               OUT     TUARTAS
DAE9 3EC0               MVI     A,0C0H
DAEB D330               OUT     TUARTBS

                        ENDIF


DAED C33DDB             JMP     GOCPM           ;INITIALIZE AND GO TO CP/M
                ;
                WBOOT:  ;SIMPLEST CASE IS TO READ THE DISK UNTIL ALL SECTORS LOADED

DAF0 3EFF               MVI     A,0FFH
DAF2 32C9DA             STA     BOOTFLAG        ;TELL THAT WE ARE WARM

DAF5 3100C4             LXI     SP,CCP          ;SET UP STACK BELOW CP/M
DAF8 0E00               MVI     C,0             ;SELECT DISK 0
```

```
DAFA CD06F8           CALL    CSELDSK
DAFD CD03F8           CALL    CHOME           ;GO TO TRACK 00

DB00 062C             MVI     B,NSECTS        ;B COUNTS # OF SECTORS TO LOAD
DB02 0E00             MVI     C,0             ;C HAS THE CURRENT TRACK NUMBER
DB04 1602             MVI     D,2             ;D HAS THE NEXT SECTOR TO READ

                ;     NOTE THAT WE BEGIN BY READING TRACK 0, SECTOR 2 SINCE SECTOR 1
                ;     CONTAINS THE COLD START LOADER, WHICH IS SKIPPED IN A WARM START

DB06 2100C4           LXI     H,CCP           ;BASE OF CP/M (INITIAL LOAD POINT)
                LOAD1:  ;LOAD ONE MORE SECTOR
DB09 C5               PUSH    B               ;SAVE SECTOR COUNT, CURRENT TRACK
DB0A D5               PUSH    D               ;SAVE NEXT SECTOR TO READ
DB0B E5               PUSH    H               ;SAVE DMA ADDRESS
DB0C 4A               MOV     C,D             ;GET SECTOR ADDRESS TO REGISTER C
DB0D CD0CF8           CALL    CSETSEC         ;CONTROLLER SET SECTOR ADDRESS FROM REGISTER C
DB10 C1               POP     B               ;RECALL DMA ADDRESS TO B,C
DB11 C5               PUSH    B               ;REPLACE ON STACK FOR LATER RECALL
DB12 CD0FF8           CALL    CSETDMA         ;SET DMA ADDRESS FROM B,C
                ;
                ;     DRIVE SET TO 0, TRACK SET, SECTOR SET, DMA ADDRESS SET
DB15 CD12F8           CALL    DISKREAD        ;DIRECT DISK READ OF ONE SECTOR
DB18 B7               ORA     A               ;ANY ERRORS?
DB19 C2F0DA           JNZ     WBOOT           ;RETRY THE ENTIRE BOOT IF AN ERROR OCCURS
                ;
                ;     NO ERROR, MOVE TO NEXT SECTOR
DB1C E1               POP     H               ;RECALL DMA ADDRESS
DB1D 118000           LXI     D,128           ;DMA=DMA+128
DB20 19               DAD     D               ;NEW DMA ADDRESS IS IN H,L
DB21 D1               POP     D               ;RECALL SECTOR ADDRESS
DB22 C1               POP     B               ;RECALL NUMBER OF SECTORS REMAINING
                                              ;AND CURRENT TRK
DB23 05               DCR     B               ;SECTORS=SECTORS-1
DB24 CA3DDB           JZ      GOCPM           ;TRANSFER TO CP/M IF ALL HAVE BEEN LOADED
                ;
                ;     MORE SECTORS REMAIN TO LOAD, CHECK FOR TRACK CHANGE
DB27 14               INR     D
DB28 7A               MOV     A,D             ;SECTOR=27?, IF SO, CHANGE TRACKS
DB29 FE1B             CPI     27
DB2B DA09DB           JC      LOAD1           ;CARRY GENERATED IF SECTOR<27
                ;
                ;     END OF CURRENT TRACK, GO TO NEXT TRACK

                NXTTRK:
DB2E 1601             MVI     D,1             ;BEGIN WITH FIRST SECTOR OF NEXT TRACK
DB30 0C               INR     C               ;TRACK=TRACK+1
                ;
                ;     SAVE REGISTER STATE, AND CHANGE TRACKS
DB31 C5               PUSH    B
DB32 D5               PUSH    D
DB33 E5               PUSH    H
DB34 CD09F8           CALL    CSETTRK ;TRACK ADDRESS SET FROM REGISTER C
DB37 E1               POP     H
DB38 D1               POP     D
DB39 C1               POP     B
```

```
DB3A C309DB          JMP     LOAD1          ;FOR ANOTHER SECTOR
                     ;
                     ;       *       *       *       *       *       *       *       *

                     ;       END OF LOAD OPERATION, SET PARAMETERS AND GO TO CP/M
                     GOCPM:
DB3D 3EC3            MVI     A,0C3H         ;C3 IS A JMP INSTRUCTION
DB3F 320000          STA     0              ;FOR JMP TO WBOOT
DB42 2103DA          LXI     H,BWBOOT       ;CBIOS WBOOT ENTRY POINT
DB45 220100          SHLD    1              ;SET ADDRESS FIELD FOR JMP AT 0
                     ;
DB48 320500          STA     5              ;FOR JMP TO BDOS
DB4B 2106CC          LXI     H,BDOS         ;BDOS ENTRY POINT
DB4E 220600          SHLD    6              ;ADDRESS FIELD OF JUMP AT 5 TO BDOS
                     ;
DB51 018000          LXI     B,80H          ;DEFAULT DMA ADDRESS IS 80H
DB54 CD0FF8          CALL    CSETDMA

DB57 3AC9DA          LDA     BOOTFLAG
DB5A FE00            CPI     0              ;ARE WE COLD?
DB5C C2A1DB          JNZ     GOCCP          ;NO, WE ARE WARM, SKIP INIT CYCLE

                     ;*****************************************************

                             ;INIT THE IOPROM IN THE SYSTEM
                             ;BUT FIRST WE MUST CHECK IF THERE IS AN I/O PROM PRESENT,
                             ;AND MOVE THE JUMP TABLE AS REQUIRED

                     PROMCHK:
DB5F 2100F0          LXI     H,IOPROM
DB62 7E              MOV     A,M
DB63 FEC3            CPI     0C3H           ;IS THE BYTE A JMP?
DB65 C2A1DB          JNZ     GOCCP          ;NO!
DB68 2C              INR     L              ;MAYBE
DB69 2C              INR     L
DB6A 2C              INR     L
DB6B 7E              MOV     A,M
DB6C FEC3            CPI     0C3H
DB6E C274DB          JNZ     HELLO          ;NO!
DB71 CD7FDB          CALL    MOVECTORS      ;YES
DB74 0E09    HELLO   MVI     C,09           ;CPM SEND MESS COMMAND
DB76 115E00          LXI     D,SIGNON       ;LOAD ADDR OF MESSAGE
DB79 CD06CC          CALL    BDOS           ;SEND IT
DB7C C3A1DB          JMP     GOCCP

                     MOVECTORS:
                             *********************************************************
                             *
                             *  CHECKS PROM FOR JUMP VECTOR ADDRESS AND MOVES THEM
                             *  TO BIOS IF NON-ZERO
                             *
                             *********************************************************

DB7F 1107F0          LXI     D,IOPROM + 7H  ;ADDR OF NEW TABLE
DB82 2107DA          LXI     H,BIOS + 7H    ;WHERE THIS BIOS WANTS IT
DB85 1A      AGAIN:  LDAX    D
```

```
DB86 FE00           CPI     0               ;IS THE HI BYTE 0? (IF IT IS, IT CANNOT
DB88 CA94DB         JZ      SKIPIT          ;YES                BE A REAL ADDRESS)
DB8B 1A             LDAX    D               ;NO, MOVE 2 BYTES
DB8C 77             MOV     M,A
DB8D 13             INX     D
DB8E 23             INX     H
DB8F 1A             LDAX    D
DB90 77             MOV     M,A
DB91 C396DB         JMP     SKIP2           ;MOVE POINTERS TO NEXT INSTRUCTION
DB94 23       SKIPIT: INX   H
DB95 13               INX   D
DB96 23       SKIP2:  INX   H
DB97 23               INX   H
DB98 13               INX   D
DB99 13               INX   D
DB9A 7B       NEXT:   MOV   A,E             ;WHERE ARE WE?
DB9B FE30             CPI   30H
DB9D DA85DB           JC    AGAIN           ;NOT DONE YET - LOOP AGAIN
DBA0 C9               RET                   ;FINISHED - EXIT

              ;
              GOCCP:
              ;               EI              ;ENABLE THE INTERRUPT SYSTEM
DBA1 3A0400           LDA     CPMDRIVE        ;CPM BOOT DRIVE
DBA4 4F               MOV     C,A             ;SEND TO THE CCP
DBA5 C300C4           JMP     CCP             ;GO TO CP/M FOR FURTHER PROCESSING
              ;
              ;       *       *       *       *       *       *       *       *
              ;
              ;       ERROR CHECKING READ AND WRITE RTNS FOR
              ;       MICROMATION CBIOS
              ;
              ;               AUGUST 1,1978
              ;
              ;
              WRITE
DBA8 CD5ADC           CALL    SETUP           ;SELECT DISK, STEP TO CORRECT TRACK
DBAB CD1EF8           CALL    WRITEPROTECT    ;SETS Z IF DRIVE IS PROTECTED
DBAE C2C2DB           JNZ     WRTOK           ;IF ENABLED, CONTINUE
              ;
DBB1 1102DC           LXI     D,PROTMES       ;WRITE PROTECT MESAGE
DBB4 CD63DD           CALL    PRINT           ;PRINT IT
              ;
DBB7 CD09DA           CALL    CONIN1          ;WAIT FOR A CHAR & UNLOAD HEAD
DBBA FE03             CPI     3               ;CONTROL C?
DBBC CAF0DA           JZ      WBOOT           ;REBOOT IF SO, OTHERWISE
DBBF C3A8DB           JMP     WRITE           ;LOOP UNTIL NOT PROTECTED

              WRTOK
DBC2 3E15             MVI     A,DISKWRITE AND 0FFH    ;LOW ADDRESS OF WRITE ROUTINE IN PROM
DBC4 C3CCDB           JMP     READWRITE

              READ
DBC7 CD5ADC           CALL    SETUP           ;SELECT DRIVE, STEP TO CORRECT TRACK
DBCA 3E12             MVI     A,DISKREAD AND 0FFH     ;LOW ADDRESS OF READ ROUTINE IN PROM
```

```
                READWRITE
DBCC 32D8DB             STA     RW+1            ;STORE ADDRESS IN CALL INSTRUCTION
DBCF AF                 XRA     A
DBD0 3220FC             STA     TRY1            ;ZERO OUT ERROR COUNTERS
DBD3 3221FC             STA     RETRYCOUNT
                RETRY:
DBD6 F3                 DI
DBD7 CD12F8     RW      CALL    DISKREAD        ;READ OR WRITE SECTOR
                                                ;SETS Z ON SUCCESSFUL READ
                                                ;SETS C ON TRACK ERROR
DBDA FB                 EI
DBDB 3E00               MVI     A,0
DBDD C8                 RZ                      ;SUCCESSFUL READ. RETURN

                                                ;READ OR WRITE ERROR HANDLERS
DBDE DAEBDB             JC      TRACKERROR      ;READ AND WRITE SET C ON TRACK ERROR
DBE1 2121FC             LXI     H,RETRYCOUNT
DBE4 34                 INR     M               ;INCREMENT RETRYCOUNT
DBE5 7E                 MOV     A,M
DBE6 E60F               ANI     0FH             ;16 TRIES
DBE8 C2D6DB             JNZ     RETRY           ;IF MORE THAN 16 RETRIES, TRY RESEEKING

                TRACKERROR
DBEB 2120FC             LXI     H,TRY1
DBEE 34                 INR     M               ;INCREMENT NO OF TRACK ERRORS
DBEF 7E                 MOV     A,M
DBF0 FE0A               CPI     10              ;ALLOW ONLY 10 TRACK ERRORS
DBF2 3E01               MVI     A,1             ;CP/M CONVENTION FOR PERMANENT ERROR
DBF4 C8                 RZ                      ;IF >10, RETURN A FAILURE

DBF5 CD03F8             CALL    CHOME           ;ELSE, HOME THE HEAD
DBF8 3A4200             LDA     TRACK1
DBFB 4F                 MOV     C,A
DBFC CD09F8             CALL    CSETTRK ;RESEEK TO CORRECT TRACK
DBFF C3D6DB             JMP     RETRY

                PROTMES:
DC02 0D0A0A5052         DB      CR,LF,LF,'PROTECTED - INSERT'
DC17 2041204E4F         DB      ' A NON-PROTECTED DISK'
DC2C 20414E4420         DB      ' AND TYPE RETURN'
DC3C 0D0A0A284F         DB      CR,LF,LF,'(OR TYPE CTL C TO ABORT)',CR,LF,'$'
                ;
                *---------------------------------------------------------------
                SETUP:
                *---------------------------------------------------------------
                ;SELECTS REQUESTED DRIVE IF IT IS NOT ALREADY SELECTED
                ;STEPS TO REQUESTED TRACK
                ;SETS UP CONTROLLER TO TRANSFER DATA TO/FROM PROPER SIDE OF DISK

DC5A 3A0DFC             LDA     NXTDISK         ;GET NEXT DISK NO.
DC5D 4F                 MOV     C,A
DC5E 3A05FC             LDA     PRESDSK         ;GET PRES DSK.
DC61 B9                 CMP     C               ;EQUAL?
DC62 C406F8             CNZ     CSELDSK         ;IF NO, SELECT NEXT DISK
                ;**********************************
```

```
                ;FIX FOR BUG IN C.1 PROM
                ;IF YOUR DOUBLER HAS ANY PROM VERSION OTHER THAN C.1, YOU DON'T NEED THIS
                ;       LDA     RDSTAT          ;GET DRIVE STATUS
                ;       ANI     20H             ;HEAD LOADED?
                ;       CNZ     CSELDSK         ;IF NO, CALL CSELDSK TO LOAD HEAD
                ;                               ;(C REG ALREADY HAS PROPER DISK NUMBER)
                ;END OF FIX FOR BUG IN C.1 PROM
                ;*********************************

DC65 3A4200             LDA     TRACK1          ;GET REQUESTED TRACK NUMBER
DC68 4F                 MOV     C,A
DC69 3A04FC             LDA     TRACK           ;GET TRACK # WE'RE ON
DC6C B9                 CMP     C               ;ARE WE ON REQUESTED TRACK?
DC6D C409F8             CNZ     CSETTRK         ;IF NO, CALL CONTROLLER SETTRK ROUTINE

DC70 3A4D00             LDA     SIDE            ;SIDE = 00 FOR SIDE 0
DC73 0F                 RRC                     ;SHIFT SIDE BIT INTO CARRY
DC74 3F                 CMC                     ;COMPLEMENT CARRY
DC75 3A03FC             LDA     CTRBYTE
DC78 17                 RAL                     ;SHIFT CARRY INTO BIT 0, SET CARRY = BIT 7
DC79 0F                 RRC                     ;ROTATE IT BACK TO NORMAL
DC7A 3203FC             STA     CTRBYTE
DC7D C9                 RET

                ;       *       *       *       *       *       *       *       *

                ;
                ;       I/O DRIVERS FOR THE DISK FOLLOW
                ;       FOR NOW, WE WILL SIMPLY STORE THE PARAMETERS AWAY FOR USE
                ;       IN THE READ AND WRITE SUBROUTINES
                ;

                ;---------------------------------------------------------------
                SELDSK: ;SELECT DISK GIVEN BY REGISTER C
                ;---------------------------------------------------------------
                ;ON ENTRY, BIT 0 OF REGISTER E CONTAINS THE DRIVE STATUS
                ;BIT 0 IS 0 IF THE SELECTED DRIVE IS OFF LINE (NEVER BEEN SELECTED
                ;               SINCE LAST WARM BOOT)
                ;WE WANT TO TEST THE DENSITY OF THE DISK IF IT IS OFF LINE
DC7E 210000             LXI     H,0000H         ;ERROR RETURN CODE
DC81 79                 MOV     A,C
DC82 FE04               CPI     NDRIVES         ;VALID DRIVE NUMBER?
DC84 D0                 RNC                     ;NO CARRY IF 4,5,...

DC85 320DFC             STA     NXTDISK         ;DOUBLER SCRATCH LOC. FOR NEXT DISKOP
                ;
                ;       DISK NUMBER IS IN THE PROPER RANGE
                ;
DC88 7B                 MOV     A,E             ;GET DRIVE STATUS FOR EVALUATION
DC89 1F                 RAR                     ;GET BIT 0 - LOGIN STAT INTO CARRY
DC8A D49CDC             CNC     SETPARM         ;CARRY IS CLEAR IF DRIVE IS JUST BEING
                                                ;SELECTED FOR THE FIRST TIME
                                                ;SELECT DESIRED DRIVE, TEST DENSITY
                ;
                ;       COMPUTE PROPER DISK PARAMETER HEADER ADDRESS
                ;
```

```
                CALCDPE
DC8D 3A0DFC             LDA     NXTDISK         ;GET DISK, WHICH BACAME PRESENT AFTER SEL.
DC90 6F                 MOV     L,A             ;L=DISK NUMBER 0,1,2,3
DC91 2600               MVI     H,0             ;HIGH ORDER ZERO
DC93 29                 DAD     H               ;*2
DC94 29                 DAD     H               ;*4
DC95 29                 DAD     H               ;*8
DC96 29                 DAD     H               ;*16 (SIZE OF EACH HEADER)
DC97 1133DA             LXI     D,DPBASE
DC9A 19                 DAD     D               ;HL=.DPBASE(DISKNO*16)
DC9B C9                 RET

                ;---------------------------------------------------------------
                SETPARM:        ;TEST DISK WHICH IS BEING SELECTED FOR THE FIRST TIME
                ;---------------------------------------------------------------

                        ;ON ENTRY, C REG CONTAINS DISK NUMBER TO BE SELECTED

DC9C CD06F8             CALL    CSELDSK         ;SELECT DISK ROUTINE IN PROM
DC9F CD03F8             CALL    CHOME           ;HOME THE DRIVE
DCA2 0E02               MVI     C,2             ;STEP TO TRACK 2
DCA4 CD09F8             CALL    CSETTRK         ;WHEN THE PROM STEPS FROM TRACKS 0 OR 1
                                                ;TO TRACK 2, IT TESTS THE DENSITY
                                                ;DENSITY INFO IN DENBYTE IS NOW VALID
DCA7 CD8DDC             CALL    CALCDPE         ;CALCULATE DISK PARAMETER HEADER ADDRESS
DCAA 0600               MVI     B,0             ;FOR DAD
DCAC 70                 MOV     M,B             ;ZERO OUT SECTOR TRANSLATE POINTER IN DPE
DCAD 23                 INX     H               ;(IF DISK IS SINGLE DENSITY, WE'LL REPLACE IT)
DCAE 70                 MOV     M,B
DCAF 3A00FE             LDA     RDSTAT          ;CHECK FOR DOUBLE SIDED DISK
DCB2 E640               ANI     40H             ;IF BIT 6 IS LOW, DISK IS DOUBLE SIDED
DCB4 0E1E               MVI     C,DPB2-DPB0     ;OFFSET IN PARM TABLE FOR DOUBLE SIDED
DCB6 CABBDC             JZ      DBLSIDD

DCB9 0E00               MVI     C,0             ;NO OFFSET FOR SINGLE SIDED
                DBLSIDD
DCBB 3A01FC             LDA     DENBYTE         ;NOW FIND DENSITY
                                                ;FOR <C> PROM, DENBYTE = 0 FOR SD, 10H FOR DD
                                                ;FOR <C1> PROM, DENBYTE = 0 FOR SD, 4 FOR DD
DCBE B7                 ORA     A               ;DENBYTE = 0?
DCBF 3E0F               MVI     A,DPB1-DPB0     ;OFFSET IN PARM TABLE FOR DOUBLE DENSITY
DCC1 C2CBDC             JNZ     DBLDEN

                                                ;SINGLE DENSITY.
DCC4 AF                 XRA     A               ;NO OFFSET FOR SINGLE DENSITY
DCC5 2B                 DCX     H               ;POINT BACK TO BEGINNING OF DP HEADER
DCC6 3673               MVI     M,TRANS AND 0FFH ;PUT IN LOW BYTE OF TRANSLATE TABLE POINTER
DCC8 23                 INX     H
DCC9 36DA               MVI     M,TRANS SHR 8   ;PUT IN HIGH BYTE OF TRANSLATE TABLE POINTER

DCCB 81         DBLDEN  ADD     C               ;ADD OFFSET FOR # SIDES
DCCC 324900             STA     DENSIDE         ;STORE IN SCRATCH FOR USE BY UTILITIES
DCCF 0E09               MVI     C,9             ;INDEX INTO PARM HEADER TO DPB FIELD
DCD1 09                 DAD     B               ;HL NOW CONTAINS ADDRESS OF DPB ENTRY OF HEADER
DCD2 EB                 XCHG                    ;SAVE IT IN DE
```

```
DCD3 4F             MOV     C,A             ;MOVE DPB OFFSET TO C FOR DAD
DCD4 218DDA         LXI     H,DPB0          ;POINT TO BEGINNING OF PARM BLOCKS
DCD7 09             DAD     B               ;NOW HL POINT TO CORRECT PARM BLOCK
DCD8 EB             XCHG                    ;NOW DE POINT TO CORRECT PARM BLOCK
                                            ;RESTORE ADDRESS OF DPB ENTRY OF HEADER TO HL
DCD9 73             MOV     M,E             ;PUT LOW BYTE OF DPB ADDRESS INTO DP HEADER
DCDA 23             INX     H
DCDB 72             MOV     M,D             ;PUT HIGH BYTE OF DPB ADDRESS INTO DP HEADER
DCDC C9             RET
            ;
            *--------------------------------------------------------------
            HOME:   ;MOVE TO THE TRACK 00 POSITION OF CURRENT DRIVE
            ;       TRANSLATE THIS CALL INTO A SETTRK CALL WITH PARAMETER 00
            *--------------------------------------------------------------

DCDD 0E00           MVI     C,0

            SETTRK: ;SET TRACK GIVEN BY REGISTER C
DCDF 79             MOV     A,C
DCE0 324200         STA     TRACK1          ;JUST STORE TRACK NUMBER
DCE3 C9             RET


                    ********************************************************
                    *       *       *       *       *       *       *      *
                    ********************************************************


            SECTRAN:

                    ;SECTOR TRANSLATE ROUTINE
                    ;THIS ROUTINE HANDLES ONLY SINGLE BYTE SECTOR VALUES
                    ;AND SINGLE BYTE TRANSLATE TABLES
                    ;ON ENTRY, B=0, C=LOGICAL SECTOR NUMBER, AND DE POINT TO
                    ;       THE TRANSLATE TABLE
                    ;ON EXIT, H=0 AND L CONTAINS THE PHYSICAL SECTOR NUMBER

                    ;IF DE=0000H, DO NOT TRANSLATE. JUST PUT MOVE BC INTO HL,
                    ;       INCREMENT (TO REFERENCE TO SECTOR 1, NOT 0), AND RETURN

                    ;IF DE <> 0000, WE ASSUME THE DISK IS SINGLE DENSITY
                    ;NORMALIZE C TO THE RANGE OF <0 TO 25> (BY SUBTRACTING 26 IF
                    ;       NECESSARY)
                    ;TRANSLATE WITH THE TABLE POINTED TO BY DE, AND, IF 26 WAS
                    ;       SUBTRACTED, ADD IT BACK IN
                    ;RESULT IS RETURNED IN HL (IT'S IN THE RANGE <1 TO 52>)

DCE4 69             MOV     L,C             ;MOVE BC (LOG SECT) TO HL (PHYS SECT)
DCE5 60             MOV     H,B             ;H IS NOW ZERO (SINCE B WAS)
DCE6 23             INX     H               ;INCREMENT FOR PHYSICAL SECTOR NUMBER
DCE7 7A             MOV     A,D
DCE8 B3             ORA     E               ;DE=0000H?
DCE9 C8             RZ                      ;IF YES, RETURN

DCEA 2B             DCX     H               ;DECR HL BACK TO LOGICAL SECT #
DCEB 79             MOV     A,C
```

```
DCEC D61A            SUI     26             ;SECTOR NUMBER < 26?
DCEE DAF4DC          JC      UNDER26

DCF1 061A            MVI     B,26           ;IF NO, PUT 26 IN B (TO ADD IN LATER)
DCF3 6F              MOV     L,A            ;PUT (C)-26 IN L REG

                     UNDER26                ;NOW L IS IN RANGE OF 0-25
DCF4 19              DAD     D              ;INDEX INTO TRANSLATE TABLE
DCF5 7E              MOV     A,M            ;GET PHYSICAL SECTOR NUMBER FROM TABLE
DCF6 80              ADD     B              ;ADD OFFSET FOR SIDE 1
DCF7 2600            MVI     H,0
DCF9 6F              MOV     L,A            ;PUT BACK IN L REG
DCFA C9              RET


                     *--------------------------------------------------------------
                     SETSEC:        ;SET SECTOR ROUTINE
                     *--------------------------------------------------------------

                                    ;DECIDES IF DISK IS SINGLE OR DOUBLE SIDED
                                    ;NORMALIZES THE SECTOR NUMBER (1-26 IN SD, 1-52 IN DD)
                                    ;UPDATES SIDE
                                    ;JUMPS TO CONTROLLER SET SECTOR ROUTINE
DCFB 2116FC          LXI     H,DENMAP       ;POINT TO DENSITY TABLE
DCFE 3A0DFC          LDA     NXTDISK        ;GET DISK NUMBER
DD01 85              ADD     L              ;INDEX INTO TABLE
DD02 6F              MOV     L,A
DD03 7E              MOV     A,M            ;GET DENSITY BYTE
DD04 214D00          LXI     H,SIDE
DD07 3600            MVI     M,0            ;INITIALIZE SIDE TO 0
DD09 B7              ORA     A              ;SINGLE DENSITY?
DD0A 79              MOV     A,C            ;MOVE SECTOR NUMBER TO ACCUMULATOR
DD0B CA10DD          JZ      SDSECT
                                            ;DOUBLE DENSITY
DD0E D61A            SUI     26             ;IF DOUBLE DENSITY, SECTOR # < 53?

DD10 D61B            SDSECT  SUI     27     ;IF SINGLE DENSITY, SECTOR # < 27?
DD12 FA18DD          JM      SIDE0          ;IF YES, SIDE=0, USE SECTOR NUMBER IN C
                                            ;SIDE 1
DD15 34              INR     M              ;SET SIDE=1
DD16 3C              INR     A
DD17 4F              MOV     C,A            ;USE SECTOR NUMBER IN ACCUMULATOR

DD18 C30CF8          SIDE0   JMP     CSETSEC        ;GO TO CONTROLLER SET SECTOR ROUTINE
                                            ;(SECTOR NUMBER IS IN C REG)

                     ;
                     ;      ************************************************************
                     ;      *                                                          *
                     ;      *   SIMPLE I/O HANDLERS (MUST BE FILLED IN BY USER)        *
                     ;      *                                                          *
                     ;      ************************************************************
                     ;
                     ;              HARDWARE UART CONSOLE ROUTINES
000D =               CR      EQU     0DH    ;ASCII <CR>
000A =               LF      EQU     0AH    ;<LF>
```

```
                ;
                CONST:  ;CONSOLE STATUS, RETURN 0FFH IF CHARACTER READY, 00H IF NOT
                ;
DD1B 3A0AFE     CONSTAT LDA     UARTSTAT
DD1E E602               ANI     2
DD20 C8                 RZ
DD21 3EFF               MVI     A,0FFH
DD23 C9                 RET
                ;
                ;       *       *       *       *       *       *       *       *
                ;
                CONIN:  ;CONSOLE CHARACTER INTO REGISTER A
DD24 CD1BDD             CALL    CONSTAT
DD27 CA24DD             JZ      CONIN
DD2A 3A02FE             LDA     UARTDATA
DD2D E67F               ANI     7FH
DD2F C9                 RET
                ;
                ;       *       *       *       *       *       *       *       *
                ;
                CONOUT: ;CONSOLE CHARACTER OUTPUT FROM REGISTER C

DD30 3A0AFE             LDA     UARTSTAT
DD33 E601               ANI     01
DD35 CA30DD             JZ      CONOUT
DD38 79                 MOV     A,C
DD39 3202FE             STA     UARTDATA
DD3C C9                 RET
                ;
                PAGE
```

```
                ;       *       *       *       *       *       *       *       *
                ;
                LIST:   ;THIS ROUTINE IMPLEMENTS THE I/O BYTE FUNCTION FOR LST: DEVICES
                *=============================================================================
                *       I/O BYTE FOR LIST DEVICE IS IMPLEMENTED AS FOLLOWS:
                *               TTY = MULTLIST (MM MULTI I/O BOARD SERIAL PORT#1)
                *               CRT = CONOUT   (MM DOUBLER SERIAL PORT)
                *               LPT = CENTLIST (CENTRONICS 703/779 TYPE LIST DEVICE)
                *               UL1 = OPTIONAL DRIVER SELECTED BY USER
                *=============================================================================

                        IF NOT (GODBIO OR SSMIO OR DPIO OR USERLST OR MULTLIST OR CENTLIST)
                        RET                     ;THEN RETURN BECAUSE IT IS NOT SUPPORTED
                        ENDIF

                        IF (GODBIO OR SSMIO OR DPIO OR USERLST OR MULTLIST OR CENTLIST)
DD3D 3A0300             LDA     IOBYTE          ;GET THE CURRENT I/O STATUS
DD40 E6C0               ANI     0C0H
DD42 CA5FDD             JZ      MULTLST         ;MM MULTI I/O BRD SERIAL PORT AS LIST
DD45 17                 RAL
DD46 D230DD             JNC     CONOUT          ;DOUBLER CONSOLE AS LIST
DD49 17                 RAL
DD4A D260DD             JNC     CENTLST         ;CENTRONICS PARALLEL PRINTER DRIVER

                LIST1:
                        IF GODBIO               ;CONDITIONAL FOR THE GODBOUT I/O BOARD FOLLOWS:
                        IN      21H
                        ANI     41H             ;BIT 0 IS TRANSMITTER BUFFER EMPTY
                        CPI     1               ;BIT 6 IS -DATA SET READY FROM PRINTER
                        JNZ     LIST1           ;NOT READY YET
                        MOV     A,C             ;LIST GETS CHARACTER IN C REG.
                        OUT     20H             ;DATA PORT
                        ENDIF                   ;STOP ASSEMBLING FOR THE GODBOUT ONLY


                        IF SSMIO                ;LIST ROUTINE FOR SSM 2P+2S I/O BOARD
                        IN      2
                        ANI     81H             ;PRINTER READY AND TX READY
                        CPI     80H
                        JNZ     LIST1
                        MOV     A,C
                ;       CPI     0AH             ;FOR AN AUTO LF DEVICE
                ;       RZ
                        OUT     3
                        ENDIF

                        IF DPIO       ;LIST ROUTINE FOR DELTA PRODUCTS CPU BOARD WITH 8251 UART
                        IN      01              ;GET PRINTER STATUS
                        ANI     81H             ;PRINTER READY AND TX READY
                        CPI     81H             ;UART READY, DATA SET READY (BOTH)
                        JNZ     LIST1           ;NOT READY, WAIT TILL READY
                        MOV     A,C
                        OUT     00              ;PRINTER READY, OUTPUT DATA BYTE
                        ENDIF                   ;STOP CONDITIONAL ASSEMBLY
```

```
                        IF USERLST              ;AREA FOR USER DEFINED LIST ROUTINE

                TUARTBOUT:
DD4D DB30               IN      TUARTBS         ;CHK STATUS
DD4F E680               ANI     TUARTBE         ;BUFFER EMPTY?
DD51 CA4DDD             JZ      TUARTBOUT       ;LOOP IF NOT
DD54 79                 MOV     A,C             ;FETCH OUTPUT CHR
DD55 B7                 ORA     A               ;TEST PARITY
DD56 E25BDD             JPO     TUARTBO1        ;SKIP IF ALREADY ODD
DD59 F680               ORI     80H             ;INSERT PARITY BIT
                TUARTBO1:
DD5B D331               OUT     TUARTB          ;WRITE CHR
DD5D C9                 RET                     ;EXIT

                        ENDIF

DD5E C9                 RET                     ;HEY WATCH OUT..DON'T GET RID OF THIS <RET>

                MULTLST
                        IF MULTLIST   ;BEGIN CONDITIONAL ASSEMBLY FOR MM I/O BOARD SERIAL OUT

                        IN      UART1C          ;GET STATUS
                        ANI     81H             ;CHECK TO SEE IF TRANSMITTER BUFFER IS EMTPY--
                        CPI     81H             ;AND IF DSR IS ACTIVE
                        JNZ     MULTLST         ;WAIT TO SEND IT IF NOT...
                        MOV     A,C             ;GET CHARACTER TO OUTPUT
                        OUT     UART1D          ;SEND IT TO LIST DEVICE
                        ENDIF                   ;STOP ASSEMBLING FOR THE MICROMATION I/O BOARD
DD5F C9                 RET

                CENTLST
                        IF CENTLIST             ;LIST ROUTINE FOR CENTRONICS 703/779 PRINTER
                        XRA     A               ;CLEAR CARRY FLAG
                        IN      STATUS          ;GET STATUS BYTE
                        RAL                     ;
                        RAL                     ;
                        RAL                     ;CHECK BIT 5:PRINTER BUSY???
                        JC      CENTLST         ;
                AKWAIT  IN      STATUS          ;
                        ANI     80H             ;CHECK BIT 7:/OBF??? (EMPTY=HIGH)
                        CPI     80H             ;
                        JNZ     AKWAIT          ;
                        MOV     A,C             ;GET CHARACTER FROM C REG.
                        CPI     0AH             ;PRINTER IS AUTO LINEFEED
                        RZ
                        ORI     80H             ;RESET DATA STROBE BIT FIRST...
                        OUT     DATAPORT        ;OUT TO PRINTER
                        OUT     PORTA           ;THIS SETS THE /OBF FLAG OF STATUS PORT
                        ANI     7FH             ;GENERATE DATA STROBE:BIT 7 OF PORT0
                        OUT     DATAPORT        ;
                        ORI     80H             ;
                        OUT     DATAPORT        ;
                        ENDIF                   ;END OF LIST ROUTINE FOR CENTRONICS 779
DD60 C9                 RET
```

```
                ENDIF                   ;END OF LIST DRIVER BLOCK...(DO NOT REMOVE.)

                ;
                ;       *       *       *       *       *       *       *       *
                ;
                LISTST: ;RETURN LIST STATUS (0 IF NOT READY, 1 IF READY)
                ;
DD61 AF         XRA     A       ;0 IS ALWAYS OK TO RETURN
                PUNCH:
DD62 C9         READER: RET


                PRINT:
DD63 1A         LDAX    D               ;GET NEXT BYTE TO PRINT
DD64 FE24       CPI     '$'             ;END?
DD66 C8         RZ

DD67 4F         MOV     C,A
DD68 CD0CDA     CALL    CONOUT1         ;OUTPUT CHARACTER TO CONSOLE
DD6B 13         INX     D               ;POINT TO NEXT CHARACTER
DD6C C363DD     JMP     PRINT
                ;
                ;
                ;*****************************************************************
                ;
                ;               WARNING
                ;
                ;       END OF CODE MUST BE LESS THAN
                ;
                ;          THE VALUE OF ENDCODE
                ;
                ;
DD80 =          ENDCODE EQU     BIOS + 380H

                *****************************************************************


                ;       ************************************************************
                ;       *                                                          *
                ;       *  THE REMAINDER OF THE CBIOS IS RESERVED UNINITIALIZED    *
                ;       *  DATA AREA, AND DOES NOT NEED TO BE A PART OF THE        *
                ;       *  SYSTEM MEMORY IMAGE (THE SPACE MUST BE AVAILABLE,       *
                ;       *  HOWEVER, BETWEEN BEGDAT AND ENDDAT).                    *
                ;       *                                                          *
                ;       ************************************************************
                ;
                ;
                ;       SCRATCH RAM AREA FOR BDOS USE
                ;
DD6F =          BEGDAT  EQU     $       ;BEGINING OF DATA AREA
DD6F            DIRBUF  DS      128     ;DIRECTORY ACCESS BUFFER

                ;
DDEF            ALV0            DS      31      ;ALLOCATION VECTOR 0
DE0E            CSV0            DS      32      ;CHECK VECTOR 0
                ;
```

```
DE2E            ALV1            DS      31      ;ALLOCATION VECTOR 1
DE4D            CSV1            DS      32      ;CHECK VECTOR 1
                ;
DE6D            ALV2            DS      31      ;ALLOCATION VECTOR 2
DE8C            CSV2            DS      32      ;CHECK VECTOR 2
                ;
                ;
DEAC            ALV3            DS      31      ;ALLOCATION VECTOR 3
DECB            CSV3            DS      32      ;CHECK VECTOR 3
                ;
DEEB =          ENDDAT          EQU     $               ;END OF DATA AREA
017C =          DATSIZ          EQU     $-BEGDAT        ;SIZE OF DATA AREA
                ;
                ;-----------------------------------
DEEB                    END
```

```
                ;*****************************************************************
                ;*                                                               *
                ;*       CUSTOMIZED  BIOS FOR MICROMATION 'DOUBLER' AS USED WITH *
                ;*                <C> OR <C1> PROM (ON DOUBLER).                 *
                ;*            SINGLE/DOUBLE DENSITY, SINGLE/DOUBLE SIDED.        *
                ;*===============================================================*
                ; WITH DRIVERS FOR MM MULTI I/O BRD IN SINGLE USER CONFIGURATION*
                ;*===============================================================*
                ;*       COPYRIGHT (C) 1979, MICROMATION AND DIGITAL RESEARCH.   *
                ;*                                                               *
                ;*       LAST UPDATED ON FEB 12, 1980                            *
                ;*                                                               *
                ;*       REVISED FOR CAVRO STD SYSTEM                            *
                ;*       VERSION 2.0  REV 6  OCT 27,1981                         *
                ;*                                                               *
                ;*****************************************************************

                ;*************************************************************************
0000 =          TRUE            EQU     -1      !       FALSE           EQU     0
                ;*************************************************************************
                ;THE FOLLOWING LABELS MAY BE USER DEFINED FOR DIFFERENT OPERATING ENVIRONMENTS:

0038 =          MSIZE           EQU     56      ;SIZE OF OPERATING SYSTEM IN KILOBYTES
                                                ;(CURRENTLY 48K).  THIS NUMBER MUST BE
                                                ;CHANGED FOR LARGER SYSTEMS.


F000 =          IOPROM          EQU     0F000H



0004 =          NDRIVES         EQU     4       ;NUMBER OF DISK DRIVES SUPPORTED BY
                                                ;THIS CBIOS

                ;=================================================================================
                ;       I/O BYTE FOR LIST DEVICE IS IMPLEMENTED AS FOLLOWS:
                ;               "TTY" = MULTLIST (MM MULTI I/O BOARD SERIAL PORT#1)
                ;               "CRT" = CONOUT   (MM DOUBLER SERIAL PORT)
                ;               "LPT" = CENTLIST (CENTRONICS 703/779 TYPE LIST DEVICE)
                ;               "UL1" = OPTIONAL DRIVER TO BE SELECTED BY USER (SEE BELOW)
                ;=================================================================================
                ;                                                       ==>UPDATED:2-12-80
                ;
                ;       LIST DEVICE EQUATES: (THESE COULD BE SET FALSE TO SAVE BIOS SPACE IF
                ;                             USER LIST DRIVER IS TOO LARGE TO FIT OTHERWISE)
                ;
0000 =          MULTLIST        EQU     FALSE
0000 =          CENTLIST        EQU     FALSE
                ;---------------------------------------------------------------------------------
                ;       LIST DEVICE OPTIONS: SET ONLY ONE FLAG TRUE FOR DESIRED DRIVER AS "UL1"
                ;
0000 =          NONE            EQU     FALSE   ;NO "UL1" FUNCTION DESIRED
FFFF =          GODBIO          EQU     TRUE    ;GODBOUT I/O BOARD AS LST:
0000 =          SSMIO           EQU     FALSE   ;SOLID STATE MUSIC 2S+P AS LST:
0000 =          DPIO            EQU     FALSE   ;DELTA PRODUCTS CPU BOARD AS LST:
0000 =          USERLST         EQU     FALSE   ;SET FLAG TO INSERT USER DEFINED LST:
                                                ;THIS CODE MUST BE INSERTED IN BIOS UNDER LIST:
```

```
                                    ;AND INITIALIZATION CODE UNDER (COLD) BOOT
                ;-------------------------------------------------------------------------
                ;
                ;       "BIAS" IS ADDRESS OFFSET FROM 3400H FOR MEMORY SYSTEMS
                ;       THAN 20K (REFERRED TO AS "B" THROUGHOUT THE TEXT).
                ;
9000 =          BIAS    EQU     (MSIZE-20)*1024
C400 =          CCP     EQU     3400H+BIAS      ;BASE OF CCP
CC06 =          BDOS    EQU     CCP+806H        ;BASE OF BDOS
DA00 =          BIOS    EQU     CCP+1600H       ;BASE OF BIOS
                ;
0003 =          IOBYTE  EQU     0003H   ;INTEL I/O BYTE


                ;       CBIOS FOR MICROMATION DOUBLE DENSITY CONTROLLER
                ;

F800 =          CONTROLLER      EQU     0F800H          ;ADDRESS OF CONTROLLER
F800 =          PROM            EQU     CONTROLLER      ;ADDRESS OF DOUBLER PROM
FC00 =          BUFF            EQU     CONTROLLER+400H ;ADDRESS OF SCRATCH RAM


                ;       *************************************
                ;       *                                   *
                ;       *       HARDWARE PORT DEFINITIONS   *
                ;       *                                   *
                ;       *************************************
                ;

                ;
FE00 =          WRCONT          EQU     CONTROLLER+600H
FE00 =          RDSTAT          EQU     WRCONT
FE01 =          WRCLK           EQU     WRCONT+1
FE02 =          UARTDATA        EQU     WRCONT+2
FE05 =          RDMARK          EQU     WRCONT+5        ;LOADS THE HEAD
FE0A =          UARTSTAT        EQU     WRCONT+0AH

                ;       *************************************
                ;       *                                   *
                ;       *       RAM VARIABLE DEFINITIONS    *
                ;       *                                   *
                ;       *************************************
                ;
                ;
FC01 =          DENBYTE         EQU     BUFF+1          ;0 FOR SINGLDE DEN., 10 FOR DBL.DEN
FC03 =          CTRBYTE         EQU     BUFF+3          ;RAM IMAGE OF RDSTAT OR WRCONT
FC04 =          TRACK           EQU     BUFF+4          ;TRACK NUMBER FOR CURRENT DRIVE
FC05 =          PRESDSK         EQU     BUFF+5          ;CURRENTLY LOGGED IN DRIVE NO.
FC0A =          SECTOR          EQU     BUFF+0AH
FC0D =          NXTDISK         EQU     BUFF+0DH        ;DRIVE NO. FOR NEXT I/O (READ/WRITE)
FC10 =          STEPTIME        EQU     BUFF+10H        ;STEP TIME IN MSEC
FC16 =          DENMAP          EQU     BUFF+16H        ;FOR EACH OF 4 DRIVES,
                                                        ;00 FOR SINGLE DENSITY
                                                        ;10H FOR <C> PROM
                                                        ;04H FOR <C1>PROM FOR DOUBLE DENSITY
FC20 =          TRY1            EQU     BUFF+20H        ;CBIOS ERROR CHECK
FC21 =          RETRYCOUNT      EQU     BUFF+21H        ;CBIOS TRK./SECTOR ERROR CHK. BYTE
```

```
FC22 =      CURRDRIVE     EQU     BUFF+22H        ;CBIOS DISK, LOGGED IN BEFORE WARMBOOT

                    ;*****************************************
                    ;*                                       *
                    ;*        LIST DEVICE EQUATES            *
                    ;*                                       *
                    ;*****************************************

            IF (MULTLIST OR CENTLIST)
            ;ASSEMBLED ONLY FOR PORTS ON MICROMATION MULTI I/O BOARD

;===============================================================================
;
; PORT EQUATES:CHANGE VALUE OF UART1C TO RE-MAP THE PORT DECODING OF I/O BOARD
;
;       NOTE:DECODING OF BOARD I/O SPACE OCCURS ALONG 32 PORT BOUNDRIES
;            (I.E.:A5-A7 ARE SIGNIFICANT ADDRESS BITS FOR I/O BLOCK)
;===============================================================================

;*******************************************************************************

UART1C  EQU     0               ;CONTROL PORT OF 8251 #1 (IC 4B)

;*******************************************************************************

BASE    EQU     UART1C          ;EQUATE BASE FOR REST OF TABLE
UART1D  EQU     BASE+1          ;DATA PORT OF UART #1

UART2C  EQU     BASE+4          ;UART #2 (IC 5B)
UART2D  EQU     BASE+5

UART3C  EQU     BASE+8          ;UART #3 (IC 7B)
UART3D  EQU     BASE+9

UART4C  EQU     BASE+0CH        ;UART #4 (IC 9B)
UART4D  EQU     BASE+0DH

P8255   EQU     BASE+10H        ;8255 CONTROL REG. (IC 1A)
PORTC   EQU     BASE+11H
PORTB   EQU     BASE+12H
PORTA   EQU     BASE+13H

RTC     EQU     BASE+14H        ;8253 TIMER/COUNTER CONTROL PORT (IC 3B)
TIMER2  EQU     BASE+15H        ;COUNTER #2
TIMER1  EQU     BASE+16H        ;COUNTER #1
TIMER0  EQU     BASE+17H        ;COUNTER #0

PORT0   EQU     BASE+18H        ;PARALLEL OUTPUT LATCH (IC 2A)
PORT1   EQU     BASE+19H        ;PARALLEL OUTPUT LATCH (IC 3A)

USERPRT EQU     BASE+1CH        ;BASE OF USER DEFINED I/O PORTS (PORTS 1C-1FH)

;*******************************************************************************
            ENDIF               ;STOP CONDITIONAL ASSEMBLY


            IF CENTLIST
```

```
                ;CONDITIONAL ASM FOR MM MULTI I/O BRD FOR USE WITH CENRONICS PRINTER

                COMMAND         EQU     P8255           ;COMMAND WORD REGISTER
                DATAPORT        EQU     PORTO           ;PARALLEL DATA OUTPUT LATCH TO PRINTER
                STATUS          EQU     PORTC           ;STATUS BITS:OBF/=BIT 7
                                                        ;            ACK/=BIT 6
                                                        ;            BUSY=BIT 5
                                ENDIF                   ;END OF PORT DEFINITIONS FOR CENTLIST


                        IF USERLST      ;AREA FOR USER LIST ROUTINE EQUATES

                        ENDIF

                ;       ********************************
                ;       *                              *
                ;       *   JUMP VECTORS TO ROUTINES IN CONTROLLER PROM    *
                ;       *                              *
                ;       ********************************
                ;
                ;
F803 =          CHOME           EQU     PROM+3          ;HOMES THE DISK
F806 =          CSELDSK         EQU     PROM+6          ;SELECTS DRIVE POINTED TO BY C REG
                                                        ;AND LOADS HEAD
F809 =          CSETTRK         EQU     PROM+9          ;STEPS DRIVE TO TRACK (C)
F80C =          CSETSEC         EQU     PROM+0CH        ;SET SECTOR NUMBER
F80F =          CSETDMA         EQU     PROM+0FH        ;SET DMA ADDRESS
F815 =          DISKWRITE       EQU     PROM+15H        ;WRITE SECTOR
F812 =          DISKREAD        EQU     PROM+12H        ;SECTOR READ
F81B =          SETDEN          EQU     PROM+1BH        ;TEST DENSITY OF CURRENT DRIVE
F81E =          WRITEPROTECT            EQU     PROM+1EH        ;CHECK FOR WRITE PROTECT

                ;
                ;       ********************************
                ;
DA00                            ORG     BIOS    ;ORIGIN OF THIS PROGRAM
                ;
                ;       ********************************
                ;
                ;
002C =          NSECTS          EQU     ($-CCP)/128     ;WARM START SECTOR COUNT
                ;
0000 =          BOOTBASE        EQU     0               ;BOTTOM OF USABLE MEMORY
0004 =          CPMDRIVE        EQU     BOOTBASE+4      ;LOC. 04 OF PAGE 00. CURRENT CPM DISK
0042 =.         TRACK1          EQU     BOOTBASE+42H    ;TEMPORARY STORAGE FOR TRACK NUMBER
0049 =          DENSIDE         EQU     BOOTBASE+49H    ;INFO ON DENSITY AND SIDE FOR UTILITIES
004D =          SIDE            EQU     BOOTBASE+4DH
                ;
                ;       ********************************
                ;       *                              *
                ;       *   JUMP VECTORS FOR INDIVIDUAL SUBROUTINES    *
                ;       *       USED BY ENTIRE SYSTEM                  *
                ;       *                              *
                ;       ********************************

DA00 C3CADA             JMP     BOOT            ;COLD START
```

```
DA03 C3D9DA    BWBOOT  JMP     WBOOT           ;WARM START
DA06 C3FFDC            JMP     CONST           ;CONSOLE STATUS
DA09 C308DD    CONIN1  JMP     CONIN           ;CONSOLE CHARACTER IN
DA0C C314DD    CONOUT1 JMP     CONOUT          ;CONSOLE CHARACTER OUT
DA0F C321DD            JMP     LIST            ;LIST CHARACTER OUT
DA12 C341DD            JMP     PUNCH           ;PUNCH CHARACTER OUT
DA15 C341DD            JMP     READER          ;READER CHARACTER OUT
DA18 C3C1DC            JMP     HOME            ;BIOS MOVE HEAD TO HOME POSITION
DA1B C362DC            JMP     SELDSK          ;BIOS SELECT DISK (DOESN'T ACTUALLY SELECTS)
DA1E C3C3DC            JMP     SETTRK          ;BIOS SET TRACK
DA21 C3DFDC            JMP     SETSEC          ;BIOS SET SECTOR
DA24 C30FF8            JMP     CSETDMA         ;CONTROLLER SET DMA ADDRESS
DA27 C3ABDB            JMP     READ            ;BIOS READ DISK
DA2A C38CDB            JMP     WRITE           ;BIOS WRITE DISK
DA2D C340DD            JMP     LISTST          ;RETURN LIST STATUS
DA30 C3C8DC            JMP     SECTRAN         ;SECTOR TRANSLATE
                ;
                ;
                ;       ********************************
                ;       *                              *
                ;       *   FIXED DISK PARAMETER BLOCK TABLES FOR FOUR     *
                ;       *              DRIVE SYSTEM                        *
                ;       *                              *
                ;       ********************************
                ;
                ;
DA33 =          DPBASE  EQU     $       ;BASE OF DISK PARAMETER BLOCKS
                ;
                ;       DISK PARAMETER BLOCK FOR DRIVE NO. 00
                ;
DA33 73DA0000   DPE0:   DW      TRANS,0000H     ;TRANSLATE TABLE
DA37 00000000           DW      0000H,0000H     ;SCRATCH AREA
DA3B 4EDD8DDA           DW      DIRBUF,DPB0     ;DIR BUFF, PARM BLOCK
DA3F EDDDCEDD           DW      CSV0,ALV0       ;CHECK, ALLOC VECTORS
                ;
                ;       DISK PARAMETER BLOCK FOR DRIVE NO. 01
                ;
DA43 73DA0000   DPE1:   DW      TRANS,0000H     ;TRANSLATE TABLE
DA47 00000000           DW      0000H,0000H     ;SCRATCH AREA
DA4B 4EDD8DDA           DW      DIRBUF,DPB0     ;DIR BUFF, PARM BLOCK
DA4F 2CDE0DDE           DW      CSV1,ALV1       ;CHECK, ALLOC VECTORS
                ;
                ;       DISK PARAMETER BLOCK FOR DRIVE NO. 02
                ;
DA53 73DA0000   DPE2:   DW      TRANS,0000H     ;TRANSLATE TABLE
DA57 00000000           DW      0000H,0000H     ;SCRATCH AREA
DA5B 4EDD8DDA           DW      DIRBUF,DPB0     ;DIR BUFF, PARM BLOCK
DA5F 6BDE4CDE           DW      CSV2,ALV2       ;CHECK, ALLOC VECTORS
                ;
                ;       DISK PARAMETER BLOCK FOR DRIVE NO. 03
                ;
DA63 73DA0000   DPE3:   DW      TRANS,0000H     ;TRANSLATE TABLE
DA67 00000000           DW      0000H,0000H     ;SCRATCH AREA
DA6B 4EDD8DDA           DW      DIRBUF,DPB0     ;DIR BUFF, PARM BLOCK
DA6F AADE8BDE           DW      CSV3,ALV3       ;CHECK, ALLOC VECTORS
                ;
                ;
```

```
                ;       SINGLE DENSITY SECTOR TRANSLATE TABLE
                ;
                ;
DA73 01070D13   TRANS:  DB      1,7,13,19       ;SECTORS 1,2,3,4
DA77 19050B11           DB      25,5,11,17      ;SECTORS 5,6,7,8
DA7B 1703090F           DB      23,3,9,15       ;SECTORS 9,10,11,12
DA7F 1502080E           DB      21,2,8,14       ;SECTORS 13,14,15,16
DA83 141A060C           DB      20,26,6,12      ;SECTORS 17,18,19,20
DA87 1218040A           DB      18,24,4,10      ;SECTORS 21,22,23,24
DA8B 1016               DB      16,22           ;SECTORS 25,26
                ;
                ;
                                ;DISK PARAMETER BLOCK FOR SINGLE DENSITY SINGLE SIDED DISKS
                ;                       WITH BLOCK SIZE BLKSZ = 1024 BYTES / BLOCK
                DPB0:
                ;
DA8D 1A00               DW      26              ;SECTORS PER TRACK
DA8F 03                 DB      3               ;BLOCK SHIFT FACTOR
DA90 07                 DB      7               ;BLOCK MASK
DA91 00                 DB      0               ;NULL MASK
DA92 F200               DW      242             ;DISK SIZE-1 (NO. OF BLOCKS/DISK-1)
DA94 3F00               DW      63              ;NO. OF DIRECTORY ENTRIES MAX.-1
DA96 C0                 DB      192             ;DIRECTORY ALOCATION SPACE MASK, 1 ST BYTE
DA97 00                 DB      0               ;SAME AS ABOVE, 2 ND BYTE
DA98 1000               DW      16              ;CHECK SIZE - (64 DIR ENTRIES DIV BY 4)
DA9A 0200               DW      2               ;NO. OF SYSTEM (NOT ACCESSABLE) TRACKS
                ;
                DPB1:           ;DISK PARAMETER BLOCK FOR DOUBLE DEN., SINGLE SIDED DISKS
                ;                       WITH BLOCK SIZE BLKSZ = 2048 BYTES / BLOCK
                ;
DA9C 3400               DW      52              ;SECTORS PER TRACK
DA9E 04                 DB      4               ;BLOCK SHIFT FACTOR
DA9F 0F                 DB      15              ;BLOCK MASK
DAA0 01                 DB      1               ;EXTENT MASK
DAA1 F200               DW      242             ;DISK SIZE-1 (NO. OF BLOCKS/DISK-1)
DAA3 7F00               DW      127             ;NO. OF DIRECTORY ENTRIES MAX.-1
DAA5 C0                 DB      192             ;DIRECTORY ALOCATION SPACE MASK, 1 ST BYTE
DAA6 00                 DB      0               ;SAME AS ABOVE, 2 ND BYTE
DAA7 2000               DW      32              ;CHECK SIZE
DAA9 0200               DW      2               ;NO. OF SYSTEM (NOT ACCESSABLE) TRACKS
                ;
                DPB2:           ;DISK PARAMETER BLOCK FOR SINGLE DEN., DOUBLE SIDED DISKS
                ;                       WITH BLOCK SIZE BLKSZ = 2048 BYTES / BLOCK
                ;
DAAB 3400               DW      52              ;SECTORS PER TRACK
DAAD 04                 DB      4               ;BLOCK SHIFT FACTOR
DAAE 0F                 DB      15              ;BLOCK MASK
DAAF 01                 DB      1               ;EXTENT MASK
DAB0 F200               DW      242             ;DISK SIZE-1 (NO. OF BLOCKS/DISK-1)
DAB2 7F00               DW      127             ;NO. OF DIRECTORY ENTRIES MAX.-1
DAB4 C0                 DB      192             ;DIRECTORY ALOCATION SPACE MASK, 1 ST BYTE
DAB5 00                 DB      0               ;SAME AS ABOVE, 2 ND BYTE
DAB6 2000               DW      32              ;CHECK SIZE
DAB8 0200               DW      2               ;NO. OF SYSTEM (NOT ACCESSABLE) TRACKS
                ;
                DPB3:           ;DISK PARAMETER BLOCK FOR DOUBLE DEN., DOUBLE SIDED DISKS
                ;                       WITH BLOCK SIZE BLKSZ = 4096 BYTES / BLOCK
```

```
DABA 6800           DW      104             ;SECTORS PER TRACK
DABC 05             DB      5               ;BLOCK SHIFT FACTOR
DABD 1F             DB      31              ;BLOCK MASK
DABE 03             DB      3               ;EXTENT MASK
DABF F200           DW      242             ;DISK SIZE-1 (NO. OF BLOCKS/DISK-1)
DAC1 7F00           DW      127             ;NO. OF DIRECTORY ENTRIES MAX.-1
DAC3 80             DB      128             ;DIRECTORY ALOCATION SPACE MASK, 1 ST BYTE
DAC4 00             DB      0               ;SAME AS ABOVE, 2 ND BYTE
DAC5 2000           DW      32              ;CHECK SIZE
DAC7 0200           DW      2               ;NO. OF SYSTEM (NOT ACCESSABLE) TRACKS
                    ;
                    ;
                    ;       ********************************
                    ;       *                              *
                    ;       *      END OF FIXED TABLES     *
                    ;       *                              *
                    ;       ********************************
                    ;
                    ;
                    PAGE


                    ;       ************************************************
                    ;       *      *      *      *      *      *      *      *
                    ;       ************************************************
                    ;
DAC9 FF             BOOTFLAG:       DB      0FFH    ;SET TO 00 ON COLDBOOT


                    BOOT:   ;SIMPLEST CASE IS TO JUST PERFORM PARAMETER INITIALIZATION

DACA 3EC1                   MVI     A,0C1H          ;SET IOBYTE TO: LST:=UL1:
                                                    ;               CON:=CRT:
DACC 320300                 STA     IOBYTE          ;INITIALIZE THE IOBYTE
                                                    ;(LST:=LPT:, CON:=CRT:)
DACF AF                     XRA     A               ;ZERO IN THE ACCUM
DAD0 320400                 STA     CPMDRIVE        ;LOC 0004 - CPM CURRENT DISK NO.
DAD3 32C9DA                 STA     BOOTFLAG

                    ;-------------------------------------------------------------------------
                    ;       INITIALIZATION ROUTINES FOR LIST DEVICES FOLLOW:
                    ;-------------------------------------------------------------------------
                    ;       IF GODBIO               ;START CONDITIONAL FOR GODBOUT I/O BOARD
                    ;       XRA     A               ;INITIALIZE THE I/O BOARD
                    ;       OUT     3               ;BY SENDING A 00 TO THE STATUS PORT
                    ;       ENDIF                   ;STOP ASSEMBLING FOR THE GODBOUT ONLY
                    ;ABOVE NOT NEEDED FOR CURRENT BOARD

                            IF DPIO                 ;UART INIT FOR DELTA PRODUCTS CPU BOARD
                            MVI     A,0EH           ;BRING UART TO COMAND INSTRUCTION MODE
                            OUT     01              ;PORT 1 IS UART A STATUS
                            OUT     03              ;PORT 3 IS UART B STATUS (ALSO TIME FILL)
                            MVI     A,40H           ;COMAND INSTRUCTION MODE - RESET UART
                            OUT     01
                            OUT     03
                            MVI     A,0EEH          ;MODE - BAUD RATE FACTOR = 16X
```

```
                OUT     01              ;DISABLE PARITY, 8 BIT CHARACTER LENGTH
                OUT     03              ;
                MVI     A,37H           ;COMAND INSTRUCTION MODE-TRANSM ENABLE,
                OUT     01              ;DATA TERM READY, RECEIVE ENABLE, ERROR
                OUT     03              ;RESET, REQUEST TO SEND (IF NEEDED)
                ENDIF                   ;STOP ASSEMBLING FOR THE DELTA PRODUCTS BRD.

                IF MULTLIST
                ;CONDITIONAL ASSEMBLY FOR MICROMATION MULTI I/O BOARD FOLLOWS

                ;8251 UART INITIALIATION:
                MVI     A,0EH           ;THESE FIRST TWO BYTES PUT 8251'S IN MODE SET
                OUT     UART1C          ;
                NOP                     ;WASTE TIME AT 4MHZ....
                NOP
                MVI     A,40H           ;
                OUT     UART1C          ;
                NOP                     ;WASTE TIME AT 4MHZ....
                NOP
                MVI     A,0EEH          ;MODE SET:ASYNC,8 DATA & 2 STOP BITS,NO PARITY
                OUT     UART1C          ;
                NOP                     ;WASTE TIME AT 4MHZ....
                NOP
                MVI     A,37H           ;COMMAND WORD:ENABLE RXD & TXD, RTS/ & DTR/
                                        ;FLAGS SET, ERROR FLAGS RESET...
                OUT     UART1C          ;

                ;8253 INITIALIZATION:
                MVI     A,0B6H          ;SET MODE OF 8253 COUNTER #2 (BAUD CLOCK)
                OUT     RTC             ;
                MVI     A,13            ;DIVIDE BY 13 OF MASTER 2MHZ. CLOCK
                OUT     TIMER2          ;
                XRA     A               ;
                OUT     TIMER2          ;
                ENDIF                   ;END OF CONDITIONAL ASSEMBLY


                IF CENTLIST
                ;8255 INITIALIZATION: SETS UP CENTRONICS TYPE PRINTER DRIVER

                MVI     A,0A8H          ;DEFINE MODE: PORTA= MODE1 (OUTPUT)
                                        ;PORTC= STATUS A AND MODE0 (INPUT)
                                        ;PORTB= DON'T CARE (SET TO MODE0 OUTPUT)
                OUT     COMMAND         ;
                MVI     A,0FFH          ;TURN OFF DATA STROBE
                OUT     DATAPORT        ;
                ENDIF                   ;STOP CONDITIONAL ASSEMBLY FOR CENTLIST


                IF USERLST    ;INSERT CUSTOM LIST DEVICE INITIALIZATION IN THIS AREA

                ENDIF


DAD6 C326DB     JMP     GOCPM           ;INITIALIZE AND GO TO CP/M
                ;
                WBOOT:  ;SIMPLEST CASE IS TO READ THE DISK UNTIL ALL SECTORS LOADED
```

```
DAD9 3EFF           MVI     A,0FFH
DADB 32C9DA         STA     BOOTFLAG        ;TELL THAT WE ARE WARM

DADE 3100C4         LXI     SP,CCP          ;SET UP STACK BELOW CP/M
DAE1 0E00           MVI     C,0             ;SELECT DISK 0
DAE3 CD06F8         CALL    CSELDSK
DAE6 CD03F8         CALL    CHOME           ;GO TO TRACK 00

DAE9 062C           MVI     B,NSECTS        ;B COUNTS # OF SECTORS TO LOAD
DAEB 0E00           MVI     C,0             ;C HAS THE CURRENT TRACK NUMBER
DAED 1602           MVI     D,2             ;D HAS THE NEXT SECTOR TO READ

                ;   NOTE THAT WE BEGIN BY READING TRACK 0, SECTOR 2 SINCE SECTOR 1
                ;   CONTAINS THE COLD START LOADER, WHICH IS SKIPPED IN A WARM START

DAEF 2100C4         LXI     H,CCP           ;BASE OF CP/M (INITIAL LOAD POINT)
            LOAD1:  ;LOAD ONE MORE SECTOR
DAF2 C5             PUSH    B               ;SAVE SECTOR COUNT, CURRENT TRACK
DAF3 D5             PUSH    D               ;SAVE NEXT SECTOR TO READ
DAF4 E5             PUSH    H               ;SAVE DMA ADDRESS
DAF5 4A             MOV     C,D             ;GET SECTOR ADDRESS TO REGISTER C
DAF6 CD0CF8         CALL    CSETSEC         ;CONTROLLER SET SECTOR ADDRESS FROM REGISTER C
DAF9 C1             POP     B               ;RECALL DMA ADDRESS TO B,C
DAFA C5             PUSH    B               ;REPLACE ON STACK FOR LATER RECALL
DAFB CD0FF8         CALL    CSETDMA         ;SET DMA ADDRESS FROM B,C
                ;
                ;   DRIVE SET TO 0, TRACK SET, SECTOR SET, DMA ADDRESS SET
DAFE CD12F8         CALL    DISKREAD        ;DIRECT DISK READ OF ONE SECTOR
DB01 B7             ORA     A               ;ANY ERRORS?
DB02 C2D9DA         JNZ     WBOOT           ;RETRY THE ENTIRE BOOT IF AN ERROR OCCURS
                ;
                ;   NO ERROR, MOVE TO NEXT SECTOR
DB05 E1             POP     H               ;RECALL DMA ADDRESS
DB06 118000         LXI     D,128           ;DMA=DMA+128
DB09 19             DAD     D               ;NEW DMA ADDRESS IS IN H,L
DB0A D1             POP     D               ;RECALL SECTOR ADDRESS
DB0B C1             POP     B               ;RECALL NUMBER OF SECTORS REMAINING
                                            ;AND CURRENT TRK
DB0C 05             DCR     B               ;SECTORS=SECTORS-1
DB0D CA26DB         JZ      GOCPM           ;TRANSFER TO CP/M IF ALL HAVE BEEN LOADED
                ;
                ;   MORE SECTORS REMAIN TO LOAD, CHECK FOR TRACK CHANGE
DB10 14             INR     D
DB11 7A             MOV     A,D             ;SECTOR=27?, IF SO, CHANGE TRACKS
DB12 FE1B           CPI     27
DB14 DAF2DA         JC      LOAD1           ;CARRY GENERATED IF SECTOR<27
                ;
                ;   END OF CURRENT TRACK, GO TO NEXT TRACK

            NXTTRK:
DB17 1601           MVI     D,1             ;BEGIN WITH FIRST SECTOR OF NEXT TRACK
DB19 0C             INR     C               ;TRACK=TRACK+1
                ;
                ;   SAVE REGISTER STATE, AND CHANGE TRACKS
DB1A C5             PUSH    B
DB1B D5             PUSH    D
```

```
DB1C E5          PUSH   H
DB1D CD09F8      CALL   CSETTRK ;TRACK ADDRESS SET FROM REGISTER C
DB20 E1          POP    H
DB21 D1          POP    D
DB22 C1          POP    B
DB23 C3F2DA      JMP    LOAD1          ;FOR ANOTHER SECTOR
                 ;
                 ;      *      *      *      *      *      *      *      *

                 ;      END OF LOAD OPERATION, SET PARAMETERS AND GO TO CP/M
                 GOCPM:
DB26 3EC3        MVI    A,0C3H         ;C3 IS A JMP INSTRUCTION
DB28 320000      STA    0              ;FOR JMP TO WBOOT
DB2B 2103DA      LXI    H,BWBOOT       ;CBIOS WBOOT ENTRY POINT
DB2E 220100      SHLD   1              ;SET ADDRESS FIELD FOR JMP AT 0
                 ;
DB31 320500      STA    5              ;FOR JMP TO BDOS
DB34 2106CC      LXI    H,BDOS         ;BDOS ENTRY POINT
DB37 220600      SHLD   6              ;ADDRESS FIELD OF JUMP AT 5 TO BDOS
                 ;
DB3A 018000      LXI    B,80H          ;DEFAULT DMA ADDRESS IS 80H
DB3D CD0FF8      CALL   CSETDMA

DB40 3AC9DA      LDA    BOOTFLAG
DB43 FE00        CPI    0              ;ARE WE COLD?
DB45 C285DB      JNZ    GOCCP          ;NO, WE ARE WARM, SKIP INIT CYCLE

                 ;**************************************************

                        ;INIT THE IOPROM IN THE SYSTEM
                        ;BUT FIRST WE MUST CHECK IF THERE IS AN I/O PROM PRESENT,
                        ;AND MOVE THE JUMP TABLE AS REQUIRED

                 PROMCHK:
DB48 2100F0              LXI    H,IOPROM
DB4B 7E                  MOV    A,M
DB4C FEC3                CPI    0C3H           ;IS THE BYTE A JMP?
DB4E C285DB              JNZ    GOCCP          ;NO!
DB51 2C                  INR    L              ;MAYBE
DB52 2C                  INR    L
DB53 2C                  INR    L
DB54 7E                  MOV    A,M
DB55 FEC3                CPI    0C3H
DB57 C285DB              JNZ    GOCCP          ;NO!
DB5A CD63DB              CALL   MOVECTORS      ;YES
DB5D CD00F0              CALL   IOPROM         ;MIGHT AS WELL SIGN ON WHILE WE ARE HERE
                                               ;(THIS IS TAKEN CARE OF ELSEWHERE IF
                                               ;THERE IS NO IOPROM)
DB60 C385DB              JMP    GOCCP

                 MOVECTORS:
                        **********************************************************
                        *
                        *  CHECKS PROM FOR JUMP VECTOR ADDRESS AND MOVES THEM
                        *  TO BIOS IF NON-ZERO
                        *
                        **********************************************************
```

```
DB63 1107F0          LXI     D,IOPROM + 7H   ;ADDR OF NEW TABLE
DB66 2107DA          LXI     H,BIOS + 7H     ;WHERE THIS BIOS WANTS IT
DB69 1A      AGAIN:  LDAX    D
DB6A FE00            CPI     0               ;IS THE HI BYTE 0? (IF IT IS, IT CANNOT
DB6C CA78DB          JZ      SKIPIT          ;YES                 BE A REAL ADDRESS)
DB6F 1A              LDAX    D               ;NO, MOVE 2 BYTES
DB70 77              MOV     M,A
DB71 13              INX     D
DB72 23              INX     H
DB73 1A              LDAX    D
DB74 77              MOV     M,A
DB75 C37ADB          JMP     SKIP2           ;MOVE POINTERS TO NEXT INSTRUCTION
DB78 23      SKIPIT: INX     H
DB79 13              INX     D
DB7A 23      SKIP2:  INX     H
DB7B 23              INX     H
DB7C 13              INX     D
DB7D 13              INX     D
DB7E 7B      NEXT:   MOV     A,E             ;WHERE ARE WE?
DB7F FE30            CPI     30H
DB81 DA69DB          JC      AGAIN           ;NOT DONE YET - LOOP AGAIN
DB84 C9              RET                     ;FINISHED - EXIT

             ;

             GOCCP:
             ;               EI              ;ENABLE THE INTERRUPT SYSTEM
DB85 3A0400          LDA     CPMDRIVE        ;CPM BOOT DRIVE
DB88 4F              MOV     C,A             ;SEND TO THE CCP
DB89 C300C4          JMP     CCP             ;GO TO CP/M FOR FURTHER PROCESSING
             ;
             ;       *       *       *       *       *       *       *       *
             ;
             ;       ERROR CHECKING READ AND WRITE RTNS FOR
             ;       MICROMATION CBIOS
             ;
             ;               AUGUST 1,1978
             ;
             ;
             WRITE
DB8C CD3EDC          CALL    SETUP           ;SELECT DISK, STEP TO CORRECT TRACK
DB8F CD1EF8          CALL    WRITEPROTECT    ;SETS Z IF DRIVE IS PROTECTED
DB92 C2A6DB          JNZ     WRTOK           ;IF ENABLED, CONTINUE
             ;
DB95 11E6DB          LXI     D,PROTMES       ;WRITE PROTECT MESAGE
DB98 CD42DD          CALL    PRINT           ;PRINT IT
             ;
DB9B CD09DA          CALL    CONIN1          ;WAIT FOR A CHAR & UNLOAD HEAD
DB9E FE03            CPI     3               ;CONTROL C?
DBA0 CAD9DA          JZ      WBOOT           ;REBOOT IF SO, OTHERWISE
DBA3 C38CDB          JMP     WRITE           ;LOOP UNTIL NOT PROTECTED

             WRTOK
DBA6 3E15            MVI     A,DISKWRITE AND OFFH    ;LOW ADDRESS OF WRITE ROUTINE IN PROM
DBA8 C3B0DB          JMP     READWRITE
```

```
                READ
DBAB CD3EDC             CALL    SETUP           ;SELECT DRIVE, STEP TO CORRECT TRACK
DBAE 3E12               MVI     A,DISKREAD AND 0FFH     ;LOW ADDRESS OF READ ROUTINE IN PROM

                READWRITE
DBB0 32BCDB             STA     RW+1            ;STORE ADDRESS IN CALL INSTRUCTION
DBB3 AF                 XRA     A
DBB4 3220FC             STA     TRY1            ;ZERO OUT ERROR COUNTERS
DBB7 3221FC             STA     RETRYCOUNT
                RETRY:
DBBA F3                 DI
DBBB CD12F8     RW      CALL    DISKREAD        ;READ OR WRITE SECTOR
                                                ;SETS Z ON SUCCESSFUL READ
                                                ;SETS C ON TRACK ERROR
DBBE FB                 EI
DBBF 3E00               MVI     A,0
DBC1 C8                 RZ                      ;SUCCESSFUL READ. RETURN

                                                ;READ OR WRITE ERROR HANDLERS
DBC2 DACFDB             JC      TRACKERROR      ;READ AND WRITE SET C ON TRACK ERROR
DBC5 2121FC             LXI     H,RETRYCOUNT
DBC8 34                 INR     M               ;INCREMENT RETRYCOUNT
DBC9 7E                 MOV     A,M
DBCA E60F               ANI     0FH             ;16 TRIES
DBCC C2BADB             JNZ     RETRY           ;IF MORE THAN 16 RETRIES, TRY RESEEKING

                TRACKERROR
DBCF 2120FC             LXI     H,TRY1
DBD2 34                 INR     M               ;INCREMENT NO OF TRACK ERRORS
DBD3 7E                 MOV     A,M
DBD4 FE0A               CPI     10              ;ALLOW ONLY 10 TRACK ERRORS
DBD6 3E01               MVI     A,1             ;CP/M CONVENTION FOR PERMANENT ERROR
DBD8 C8                 RZ                      ;IF >10, RETURN A FAILURE

DBD9 CD03F8             CALL    CHOME           ;ELSE, HOME THE HEAD
DBDC 3A4200             LDA     TRACK1
DBDF 4F                 MOV     C,A
DBE0 CD09F8             CALL    CSETTRK ;RESEEK TO CORRECT TRACK
DBE3 C3BADB             JMP     RETRY

                PROTMES:
DBE6 0D0A0A5052         DB      CR,LF,LF,'PROTECTED - INSERT'
DBFB 2041204E4F         DB      ' A NON-PROTECTED DISK'
DC10 20414E4420         DB      ' AND TYPE RETURN'
DC20 0D0A0A284F         DB      CR,LF,LF,'(OR TYPE CTL C TO ABORT)',CR,LF,'$'
                ;
                ;---------------------------------------------------------------
                SETUP:
                ;---------------------------------------------------------------
                ;SELECTS REQUESTED DRIVE IF IT IS NOT ALREADY SELECTED
                ;STEPS TO REQUESTED TRACK
                ;SETS UP CONTROLLER TO TRANSFER DATA TO/FROM PROPER SIDE OF DISK

DC3E 3A0DFC             LDA     NXTDISK,        ;GET NEXT DISK NO.
DC41 4F                 MOV     C,A
DC42 3A05FC             LDA     PRESDSK         ;GET PRES DSK.
DC45 B9                 CMP     C               ;EQUAL?
```

```
DC46 C406F8          CNZ     CSELDSK         ;IF NO, SELECT NEXT DISK
                     ;!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                     ;FIX FOR BUG IN C.1 PROM
                     ;IF YOUR DOUBLER HAS ANY PROM VERSION OTHER THAN C.1, YOU DON'T NEED THIS
                     ;       LDA     RDSTAT          ;GET DRIVE STATUS
                     ;       ANI     20H             ;HEAD LOADED?
                     ;       CNZ     CSELDSK         ;IF NO, CALL CSELDSK TO LOAD HEAD
                     ;                               ;(C REG ALREADY HAS PROPER DISK NUMBER)
                     ;END OF FIX FOR BUG IN C.1 PROM
                     ;!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

DC49 3A4200          LDA     TRACK1          ;GET REQUESTED TRACK NUMBER
DC4C 4F              MOV     C,A
DC4D 3A04FC          LDA     TRACK           ;GET TRACK # WE'RE ON
DC50 B9              CMP     C               ;ARE WE ON REQUESTED TRACK?
DC51 C409FB          CNZ     CSETTRK         ;IF NO, CALL CONTROLLER SETTRK ROUTINE

DC54 3A4D00          LDA     SIDE            ;SIDE = 00 FOR SIDE 0
DC57 0F              RRC                     ;SHIFT SIDE BIT INTO CARRY
DC58 3F              CMC                     ;COMPLEMENT CARRY
DC59 3A03FC          LDA     CTRBYTE
DC5C 17              RAL                     ;SHIFT CARRY INTO BIT 0, SET CARRY = BIT 7
DC5D 0F              RRC                     ;ROTATE IT BACK TO NORMAL
DC5E 3203FC          STA     CTRBYTE
DC61 C9              RET

                     ;       !       !       !       !       !       !       !       !

                     ;
                     ;       I/O DRIVERS FOR THE DISK FOLLOW
                     ;       FOR NOW, WE WILL SIMPLY STORE THE PARAMETERS AWAY FOR USE
                     ;       IN THE READ AND WRITE SUBROUTINES
                     ;

                     !---------------------------------------------------------------
                     SELDSK: ;SELECT DISK GIVEN BY REGISTER C
                     !---------------------------------------------------------------
                     ;ON ENTRY, BIT 0 OF REGISTER E CONTAINS THE DRIVE STATUS
                     ;BIT 0 IS 0 IF THE SELECTED DRIVE IS "OFF LINE" (NEVER BEEN SELECTED
                     ;               SINCE LAST WARM BOOT)
                     ;WE WANT TO TEST THE DENSITY OF THE DISK IF IT IS OFF LINE
DC62 210000          LXI     H,0000H         ;ERROR RETURN CODE
DC65 79              MOV     A,C
DC66 FE04            CPI     NDRIVES         ;VALID DRIVE NUMBER?
DC68 D0              RNC                     ;NO CARRY IF 4,5,...

DC69 320DFC          STA     NXTDISK         ;DOUBLER SCRATCH LOC. FOR NEXT DISKOP
                     ;
                     ;       DISK NUMBER IS IN THE PROPER RANGE
                     ;
DC6C 7B              MOV     A,E             ;GET DRIVE STATUS FOR EVALUATION
DC6D 1F              RAR                     ;GET BIT 0 - LOGIN STAT INTO CARRY
DC6E D480DC          CNC     SETPARM         ;CARRY IS CLEAR IF DRIVE IS JUST BEING
                                             ;SELECTED FOR THE FIRST TIME
                                             ;SELECT DESIRED DRIVE, TEST DENSITY
                     ;
                     ;       COMPUTE PROPER DISK PARAMETER HEADER ADDRESS
```

```
                ;
                CALCDPE
DC71 3A0DFC             LDA     NXTDISK         ;GET DISK, WHICH BACAME PRESENT AFTER SEL.
DC74 6F                 MOV     L,A             ;L=DISK NUMBER 0,1,2,3
DC75 2600               MVI     H,0             ;HIGH ORDER ZERO
DC77 29                 DAD     H               ;*2
DC78 29                 DAD     H               ;*4
DC79 29                 DAD     H               ;*8
DC7A 29                 DAD     H               ;*16 (SIZE OF EACH HEADER)
DC7B 1133DA             LXI     D,DPBASE
DC7E 19                 DAD     D               ;HL=.DPBASE(DISKNO*16)
DC7F C9                 RET

                ;---------------------------------------------------------------
                SETPARM:        ;TEST DISK WHICH IS BEING SELECTED FOR THE FIRST TIME
                ;---------------------------------------------------------------

                        ;ON ENTRY, C REG CONTAINS DISK NUMBER TO BE SELECTED

DC80 CD06F8             CALL    CSELDSK         ;SELECT DISK ROUTINE IN PROM
DC83 CD03F8             CALL    CHOME           ;HOME THE DRIVE
DC86 0E02               MVI     C,2             ;STEP TO TRACK 2
DC88 CD09F8             CALL    CSETTRK         ;WHEN THE PROM STEPS FROM TRACKS 0 OR 1
                                                ;TO TRACK 2, IT TESTS THE DENSITY
                                                ;DENSITY INFO IN DENBYTE IS NOW VALID
DC8B CD71DC             CALL    CALCDPE         ;CALCULATE DISK PARAMETER HEADER ADDRESS
DC8E 0600               MVI     B,0             ;FOR DAD
DC90 70                 MOV     M,B             ;ZERO OUT SECTOR TRANSLATE POINTER IN DPE
DC91 23                 INX     H               ;(IF DISK IS SINGLE DENSITY, WE'LL REPLACE IT)
DC92 70                 MOV     M,B
DC93 3A00FE             LDA     RDSTAT          ;CHECK FOR DOUBLE SIDED DISK
DC96 E640               ANI     40H             ;IF BIT 6 IS LOW, DISK IS DOUBLE SIDED
DC98 0E1E               MVI     C,DPB2-DPB0     ;OFFSET IN PARM TABLE FOR DOUBLE SIDED
DC9A CA9FDC             JZ      DBLSIDD

DC9D 0E00               MVI     C,0             ;NO OFFSET FOR SINGLE SIDED
                DBLSIDD
DC9F 3A01FC             LDA     DENBYTE         ;NOW FIND DENSITY
                                                ;FOR <C> PROM, DENBYTE = 0 FOR SD, 10H FOR DD
                                                ;FOR <C1> PROM, DENBYTE = 0 FOR SD, 4 FOR DD
DCA2 B7                 ORA     A               ;DENBYTE = 0?
DCA3 3E0F               MVI     A,DPB1-DPB0     ;OFFSET IN PARM TABLE FOR DOUBLE DENSITY
DCA5 C2AFDC             JNZ     DBLDEN

                                                ;SINGLE DENSITY.
DCA8 AF                 XRA     A               ;NO OFFSET FOR SINGLE DENSITY
DCA9 2B                 DCX     H               ;POINT BACK TO BEGINNING OF DP HEADER
DCAA 3673               MVI     M,TRANS AND 0FFH ;PUT IN LOW BYTE OF TRANSLATE TABLE POINTER
DCAC 23                 INX     H
DCAD 36DA               MVI     M,TRANS SHR 8   ;PUT IN HIGH BYTE OF TRANSLATE TABLE POINTER

DCAF B1         DBLDEN  ADD     C               ;ADD OFFSET FOR # SIDES
DCB0 324900             STA     DENSIDE         ;STORE IN SCRATCH FOR USE BY UTILITIES
DCB3 0E09               MVI     C,9             ;INDEX INTO PARM HEADER TO DPB FIELD
DCB5 09                 DAD     B               ;HL NOW CONTAINS ADDRESS OF DPB ENTRY OF HEADER
DCB6 EB                 XCHG                    ;SAVE IT IN DE
```

```
DCB7 4F             MOV     C,A             ;MOVE DPB OFFSET TO C FOR DAD
DCB8 218DDA         LXI     H,DPB0          ;POINT TO BEGINNING OF PARM BLOCKS
DCBB 09             DAD     B               ;NOW HL POINT TO CORRECT PARM BLOCK
DCBC EB             XCHG                    ;NOW DE POINT TO CORRECT PARM BLOCK
                                            ;RESTORE ADDRESS OF DPB ENTRY OF HEADER TO HL
DCBD 73             MOV     M,E             ;PUT LOW BYTE OF DPB ADDRESS INTO DP HEADER
DCBE 23             INX     H
DCBF 72             MOV     M,D             ;PUT HIGH BYTE OF DPB ADDRESS INTO DP HEADER
DCC0 C9             RET
                ;
                ;---------------------------------------------------------------
                HOME:   ;MOVE TO THE TRACK 00 POSITION OF CURRENT DRIVE
                ;       TRANSLATE THIS CALL INTO A SETTRK CALL WITH PARAMETER 00
                ;---------------------------------------------------------------

DCC1 0E00           MVI     C,0

                SETTRK: ;SET TRACK GIVEN BY REGISTER C
DCC3 79             MOV     A,C
DCC4 324200         STA     TRACK1          ;JUST STORE TRACK NUMBER
DCC7 C9             RET


                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                        ;       ;       ;       ;       ;       ;       ;       ;
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


                SECTRAN:

                        ;SECTOR TRANSLATE ROUTINE
                        ;THIS ROUTINE HANDLES ONLY SINGLE BYTE SECTOR VALUES
                        ;AND SINGLE BYTE TRANSLATE TABLES
                        ;ON ENTRY, B=0, C=LOGICAL SECTOR NUMBER, AND DE POINT TO
                        ;       THE TRANSLATE TABLE
                        ;ON EXIT, H=0 AND L CONTAINS THE PHYSICAL SECTOR NUMBER

                        ;IF DE=0000H, DO NOT TRANSLATE. JUST PUT MOVE BC INTO HL,
                        ;       INCREMENT (TO REFERENCE TO SECTOR 1, NOT 0), AND RETURN

                        ;IF DE <> 0000, WE ASSUME THE DISK IS SINGLE DENSITY
                        ;NORMALIZE C TO THE RANGE OF <0 TO 25> (BY SUBTRACTING 26 IF
                        ;       NECESSARY)
                        ;TRANSLATE WITH THE TABLE POINTED TO BY DE, AND, IF 26 WAS
                        ;       SUBTRACTED, ADD IT BACK IN
                        ;RESULT IS RETURNED IN HL (IT'S IN THE RANGE <1 TO 52>)

DCC8 69             MOV     L,C             ;MOVE BC (LOG SECT) TO HL (PHYS SECT)
DCC9 60             MOV     H,B             ;H IS NOW ZERO (SINCE B WAS)
DCCA 23             INX     H               ;INCREMENT FOR PHYSICAL SECTOR NUMBER
DCCB 7A             MOV     A,D
DCCC B3             ORA     E               ;DE=0000H?
DCCD C8             RZ                      ;IF YES, RETURN

DCCE 2B             DCX     H               ;DECR HL BACK TO LOGICAL SECT #
DCCF 79             MOV     A,C
DCD0 D61A           SUI     26              ;SECTOR NUMBER < 26?
```

```
DCD2 DAD8DC          JC      UNDER26

DCD5 061A            MVI     B,26            ;IF NO, PUT 26 IN B (TO ADD IN LATER)
DCD7 6F              MOV     L,A             ;PUT (C)-26 IN L REG

                     UNDER26                 ;NOW L IS IN RANGE OF 0-25
DCD8 19              DAD     D               ;INDEX INTO TRANSLATE TABLE
DCD9 7E              MOV     A,M             ;GET PHYSICAL SECTOR NUMBER FROM TABLE
DCDA 80              ADD     B               ;ADD OFFSET FOR SIDE 1
DCDB 2600            MVI     H,0
DCDD 6F              MOV     L,A             ;PUT BACK IN L REG
DCDE C9              RET


                     ;---------------------------------------------------------------
                     SETSEC:         ;SET SECTOR ROUTINE
                     ;---------------------------------------------------------------

                                     ;DECIDES IF DISK IS SINGLE OR DOUBLE SIDED
                                     ;NORMALIZES THE SECTOR NUMBER (1-26 IN SD, 1-52 IN DD)
                                     ;UPDATES *SIDE
                                     ;JUMPS TO CONTROLLER SET SECTOR ROUTINE
DCDF 2116FC          LXI     H,DENMAP        ;POINT TO DENSITY TABLE
DCE2 3A0DFC          LDA     NXTDISK         ;GET DISK NUMBER
DCE5 85              ADD     L               ;INDEX INTO TABLE
DCE6 6F              MOV     L,A
DCE7 7E              MOV     A,M             ;GET DENSITY BYTE
DCE8 214D00          LXI     H,SIDE
DCEB 3600            MVI     M,0             ;INITIALIZE SIDE TO 0
DCED B7              ORA     A               ;SINGLE DENSITY?
DCEE 79              MOV     A,C             ;MOVE SECTOR NUMBER TO ACCUMULATOR
DCEF CAF4DC          JZ      SDSECT
                                             ;DOUBLE DENSITY
DCF2 D61A            SUI     26              ;IF DOUBLE DENSITY, SECTOR # < 53?

DCF4 D61B    SDSECT  SUI     27              ;IF SINGLE DENSITY, SECTOR # < 27?
DCF6 FAFCDC          JM      SIDE0           ;IF YES, SIDE=0, USE SECTOR NUMBER IN C
                                             ;SIDE 1
DCF9 34              INR     M               ;SET SIDE=1
DCFA 3C              INR     A
DCFB 4F              MOV     C,A             ;USE SECTOR NUMBER IN ACCUMULATOR

DCFC C30CF8  SIDE0   JMP     CSETSEC         ;GO TO CONTROLLER SET SECTOR ROUTINE
                                             ;(SECTOR NUMBER IS IN C REG)

                     ;
                     ;       ************************************************************
                     ;       *                                                          *
                     ;       *   SIMPLE I/O HANDLERS (MUST BE FILLED IN BY USER)        *
                     ;       *                                                          *
                     ;       ************************************************************
                     ;
                     ;               HARDWARE UART CONSOLE ROUTINES
000D =               CR      EQU     0DH     ;ASCII <CR>
000A =               LF      EQU     0AH     ;<LF>
                     ;
```

```
                CONST:  ;CONSOLE STATUS, RETURN 0FFH IF CHARACTER READY, 00H IF NOT
                ;
DCFF 3A0AFE     CONSTAT LDA     UARTSTAT
DD02 E602               ANI     2
DD04 C8                 RZ
DD05 3EFF               MVI     A,0FFH
DD07 C9                 RET
                ;
                ;       *       *       *       *       *       *       *       *
                ;
                CONIN:  ;CONSOLE CHARACTER INTO REGISTER A
DD08 CDFFDC             CALL    CONSTAT
DD0B CA08DD             JZ      CONIN
DD0E 3A02FE             LDA     UARTDATA
DD11 E67F               ANI     7FH
DD13 C9                 RET
                ;
                ;       *       *       *       *       *       *       *       *
                ;
                CONOUT: ;CONSOLE CHARACTER OUTPUT FROM REGISTER C

DD14 3A0AFE             LDA     UARTSTAT
DD17 E601               ANI     01
DD19 CA14DD             JZ      CONOUT
DD1C 79                 MOV     A,C
DD1D 3202FE             STA     UARTDATA
DD20 C9                 RET
                ;
                PAGE


                ;       *       *       *       *       *       *       *       *
                ;
                LIST:   ;THIS ROUTINE IMPLEMENTS THE I/O BYTE FUNCTION FOR LST: DEVICES
                *===============================================================================
                *       I/O BYTE FOR LIST DEVICE IS IMPLEMENTED AS FOLLOWS:
                *               "TTY" = MULTLIST (MM MULTI I/O BOARD SERIAL PORT#1)
                *               "CRT" = CONOUT   (MM DOUBLER SERIAL PORT)
                *               "LPT" = CENTLIST (CENTRONICS 703/779 TYPE LIST DEVICE)
                *               "UL1" = OPTIONAL DRIVER SELECTED BY USER
                *===============================================================================

                        IF NOT (6ODBIO OR SSMIO OR DPIO OR USERLST OR MULTLIST OR CENTLIST)
                        RET                     ;THEN RETURN BECAUSE IT IS NOT SUPPORTED
                        ENDIF

                        IF (6ODBIO OR SSMIO OR DPIO OR USERLST OR MULTLIST OR CENTLIST)
DD21 3A0300             LDA     IOBYTE          ;GET THE CURRENT I/O STATUS
DD24 E6C0               ANI     0C0H
DD26 CA3EDD             JZ      MULTLST         ;MM MULTI I/O BRD SERIAL PORT AS LIST
DD29 17                 RAL
DD2A D214DD             JNC     CONOUT          ;DOUBLER CONSOLE AS LIST
DD2D 17                 RAL
DD2E D23FDD             JNC     CENTLST         ;CENTRONICS PARALLEL PRINTER DRIVER

                LIST1:
                        IF 6ODBIO               ;CONDITIONAL FOR THE 6ODBOUT I/O BOARD FOLLOWS:
```

```
DD31 DB21           IN      21H
DD33 E641           ANI     41H             ;BIT 0 IS TRANSMITTER BUFFER EMPTY
DD35 FE01           CPI     1               ;BIT 6 IS -DATA SET READY FROM PRINTER
DD37 C231DD         JNZ     LIST1           ;NOT READY YET
DD3A 79             MOV     A,C             ;LIST GETS CHARACTER IN C REG.
DD3B D320           OUT     20H             ;DATA PORT
                    ENDIF                   ;STOP ASSEMBLING FOR THE GODBOUT ONLY


                    IF SSMIO                ;LIST ROUTINE FOR SSM 2P+2S I/O BOARD
                    IN      2
                    ANI     81H             ;PRINTER READY AND TX READY
                    CPI     80H
                    JNZ     LIST1
                    MOV     A,C
                ;   CPI     0AH             ;FOR AN AUTO LF DEVICE
                ;   RZ
                    OUT     3
                    ENDIF

                    IF DPIO       ;LIST ROUTINE FOR DELTA PRODUCTS CPU BOARD WITH 8251 UART
                    IN      01              ;GET PRINTER STATUS
                    ANI     81H             ;PRINTER READY AND TX READY
                    CPI     81H             ;UART READY, DATA SET READY (BOTH)
                    JNZ     LIST1           ;NOT READY, WAIT TILL READY
                    MOV     A,C
                    OUT     00              ;PRINTER READY, OUTPUT DATA BYTE
                    ENDIF                   ;STOP CONDITIONAL ASSEMBLY

                    IF USERLST              ;AREA FOR USER DEFINED LIST ROUTINE
                    ENDIF

DD3D C9             RET                     ;HEY WATCH OUT..DON'T GET RID OF THIS <RET>

            MULTLST
                    IF MULTLIST   ;BEGIN CONDITIONAL ASSEMBLY FOR MM I/O BOARD SERIAL OUT

                    IN      UART1C          ;GET STATUS
                    ANI     81H             ;CHECK TO SEE IF TRANSMITTER BUFFER IS EMTPY--
                    CPI     81H             ;AND IF DSR IS ACTIVE
                    JNZ     MULTLST         ;WAIT TO SEND IT IF NOT...
                    MOV     A,C             ;GET CHARACTER TO OUTPUT
                    OUT     UART1D          ;SEND IT TO LIST DEVICE
                    ENDIF                   ;STOP ASSEMBLING FOR THE MICROMATION I/O BOARD
DD3E C9             RET

            CENTLST
                    IF CENTLIST             ;LIST ROUTINE FOR CENTRONICS 703/779 PRINTER
                    XRA     A               ;CLEAR CARRY FLAG
                    IN      STATUS          ;GET STATUS BYTE
                    RAL                     ;
                    RAL                     ;
                    RAL                     ;CHECK BIT 5:PRINTER BUSY???
                    JC      CENTLST         ;
            AKWAIT  IN      STATUS          ;
                    ANI     80H             ;CHECK BIT 7:/OBF??? (EMPTY=HIGH)
                    CPI     80H             ;
```

```
                JNZ     AKWAIT          ;
                MOV     A,C             ;GET CHARACTER FROM C REG.
                CPI     0AH             ;PRINTER IS AUTO LINEFEED
                RZ
                ORI     80H             ;RESET DATA STROBE BIT FIRST...
                OUT     DATAPORT        ;OUT TO PRINTER
                OUT     PORTA           ;THIS SETS THE /OBF FLAG OF STATUS PORT
                ANI     7FH             ;GENERATE DATA STROBE:BIT 7 OF PORT0
                OUT     DATAPORT        ;
                ORI     80H             ;
                OUT     DATAPORT        ;
                ENDIF                   ;END OF LIST ROUTINE FOR CENTRONICS 779
DD3F C9         RET

                ENDIF                   ;END OF LIST DRIVER BLOCK...(DO NOT REMOVE.)

            ;
            ;       *       *       *       *       *       *       *       *
            ;
            LISTST: ;RETURN LIST STATUS (0 IF NOT READY, 1 IF READY)
            ;
DD40 AF         XRA     A       ;0 IS ALWAYS OK TO RETURN
            PUNCH:
DD41 C9     READER: RET


            PRINT:
DD42 1A         LDAX    D               ;GET NEXT BYTE TO PRINT
DD43 FE24       CPI     '$'             ;END?
DD45 C8         RZ

DD46 4F         MOV     C,A
DD47 CD0CDA     CALL    CONOUT1         ;OUTPUT CHARACTER TO CONSOLE
DD4A 13         INX     D               ;POINT TO NEXT CHARACTER
DD4B C342DD     JMP     PRINT
            ;
            ;
            ;***************************************************************
            ;
            ;               WARNING
            ;
            ;       END OF CODE MUST BE LESS THAN
            ;
            ;          THE VALUE OF ENDCODE
            ;
            ;
DD80 =      ENDCODE EQU     BIOS + 380H

            ***************************************************************


            ;       *************************************************
            ;       *                                               *
            ;       *  THE REMAINDER OF THE CBIOS IS RESERVED UNINITIALIZED
            ;       *  DATA AREA, AND DOES NOT NEED TO BE A PART OF THE
            ;       *  SYSTEM MEMORY IMAGE (THE SPACE MUST BE AVAILABLE,
            ;       *  HOWEVER, BETWEEN "BEGDAT" AND "ENDDAT").
```

```
                ;       t                                                       t
                ;       ttttttttttttttttttttttttttttttttttttttttttttttttttttttttt
                ;
                ;
                ;       SCRATCH RAM AREA FOR BDOS USE
                ;
DD4E =          BEGDAT  EQU     $       ;BEGINING OF DATA AREA
DD4E            DIRBUF  DS      128     ;DIRECTORY ACCESS BUFFER

                ;
DDCE            ALV0            DS      31      ;ALLOCATION VECTOR 0
DDED            CSV0            DS      32      ;CHECK VECTOR 0
                ;
DE0D            ALV1            DS      31      ;ALLOCATION VECTOR 1
DE2C            CSV1            DS      32      ;CHECK VECTOR 1
                ;
DE4C            ALV2            DS      31      ;ALLOCATION VECTOR 2
DE6B            CSV2            DS      32      ;CHECK VECTOR 2
                ;
                ;
DE8B            ALV3            DS      31      ;ALLOCATION VECTOR 3
DEAA            CSV3            DS      32      ;CHECK VECTOR 3
                ;
DECA =          ENDDAT          EQU     $               ;END OF DATA AREA
017C =          DATSIZ          EQU     $-BEGDAT        ;SIZE OF DATA AREA
                ;
                t-----------------------------------
DECA                    END
DB69 AGAIN      DDCE ALV0       DE0D ALV1       DE4C ALV2       DE8B ALV3
CC06 BDOS       DD4E BEGDAT     9000 BIAS       DA00 BIOS       0000 BOOTBASE
DACA BOOT       DAC9 BOOTFLAG   FC00 BUFF       DA03 BWBOOT     DC71 CALCDPE
C400 CCP        0000 CENTLIST   DD3F CENTLST    F803 CHOME      DA09 CONIN1
DD08 CONIN      DA0C CONOUT1    DD14 CONOUT     DCFF CONST      DCFF CONSTAT
F800 CONTROLLER 0004 CPMDRIVE   000D CR         F806 CSELDSK    F80F CSETDMA
F80C CSETSEC    F809 CSETTRK    DDED CSV0       DE2C CSV1       DE6B CSV2
DEAA CSV3       FC03 CTRBYTE    FC22 CURRDRIVE  017C DATSIZ     DCAF DBLDEN
DC9F DBLSIDD    FC01 DENBYTE    FC16 DENMAP     0049 DENSIDE    DD4E DIRBUF
F812 DISKREAD   F815 DISKWRITE  DA8D DPB0       DA9C DPB1       DAAB DPB2
DABA DPB3       DA33 DPBASE     DA33 DPE0       DA43 DPE1       DA53 DPE2
DA63 DPE3       0000 DPIO       DD80 ENDCODE    DECA ENDDAT     0000 FALSE
DB85 GOCCP      DB26 GOCPM      FFFF GODBIO     DCC1 HOME       0003 IOBYTE
F000 IOPROM     000A LF         DD21 LIST       DD31 LIST1      DD40 LISTST
DAF2 LOAD1      DB63 MOVECTORS  0038 MSIZE      0000 MULTLIST   DD3E MULTLST
0004 NDRIVES    DB7E NEXT       0000 NONE       002C NSECTS     FC0D NXTDISK
DB17 NXTTRK     FC05 PRESDSK    DD42 PRINT      F800 PROM       DB48 PROMCHK
DBE6 PROTMES    DD41 PUNCH      FE05 RDMARK     FE00 RDSTAT     DBAB READ
DD41 READER     DBB0 READWRITE  FC21 RETRYCOUNT DBBA RETRY      DBBB RW
DCF4 SDSECT     FC0A SECTOR     DCC8 SECTRAN    DC62 SELDSK     F81B SETDEN
DC80 SETPARM    DCDF SETSEC     DCC3 SETTRK     DC3E SETUP      004D SIDE
DCFC SIDE0      DB7A SKIP2      DB78 SKIPIT     0000 SSMIO      FC10 STEPTIME
FC04 TRACK      0042 TRACK1     DBCF TRACKERROR DA73 TRANS      FFFF TRUE
FC20 TRY1       FE02 UARTDATA   FE0A UARTSTAT   DCD8 UNDER26    0000 USERLST
DAD9 WBOOT      FE01 WRCLK      FE00 WRCONT     DB8C WRITE
F81E WRITEPROTECT               DBA6 WRTOK
```

```
                ;SPECIAL PROM FOR SIO BOARD ON IMSAI

                ;
                ; SYSTEM RAM EQUATES
                ;
                ; SYSTEM DISK LAYOUT EQUATES
                ;
005E =          SIGNON  EQU     05EH            ;LOCATION OF BOOTSTRAP MESSAGE
0005 =          BDOS    EQU     5
                ;
0000 =          WBOOT   EQU     0000
0000 =          CRTS    EQU     00H             ;CRT STATUS PORT
0001 =          CRT     EQU     01H             ;CRT I/O PORT
0002 =          RDR     EQU     02H             ;PAPER TAPE READER INPUT PORT
0000 =          RDRS    EQU     00H             ;PAPER TAPE READER STATUS
0010 =          SIOBD   EQU     10H             ;BASE PORT # OF SIO BOARD
0013 =          SIOS1   EQU     SIOBD+3         ;STATUS PORT OF IMSAI SERIAL I/O BOARD TO INIT.
0015 =          SIOS2   EQU     SIOBD+5         ;STATUS PORT OF TOTHER
0018 =          SIOC    EQU     SIOBD+8         ;CONTROL PORT FOR SIO BOARD (BOTH CHANNELS)
0012 =          TTY     EQU     SIOBD+2         ;DIABLO PRINTER I/O PORT
0013 =          TTYS    EQU     SIOBD+3         ;DIABLO STATUS PORT
0014 =          MODEM   EQU     SIOBD+4         ;MODEM I/O PORT
0015 =          MODEMS  EQU     SIOBD+5         ;MODEM STATUS PORT
                ;
                ; I/O STATUS BITS
                ;
                ;
                ;
                ; EQUATES FOR ASCII CHARACTERS
                ;
0001 =          SOH     EQU     01H
0003 =          CTRLC   EQU     03H
0008 =          BS      EQU     08H
0009 =          TAB     EQU     09H
000A =          LF      EQU     0AH
000C =          FF      EQU     0CH
000D =          CR      EQU     0DH
001A =          CTRLZ   EQU     1AH
001B =          ESC     EQU     1BH
005F =          UNDERLINE EQU   5FH
007F =          RUBOUT  EQU     7FH
                ;

F000                    ORG     0F000H


                ; ENTRY POINT TABLE
                ;
F000 C366F0     ENTAB   JMP     INITIO          ;COLD START.
F003 C30000     WBOTE   JMP     WBOOT           ;COME HERE TO INITIATE REBOOT (VÍA LOC 0)
F006 000000             DB      00,00,00        ;JMP    CONSTAT
F009 000000             DB      00,00,00        ;JMP    CONIN
F00C 000000             DB      00,00,00        ;JMP    CONOUT
```

```
F00F C351F0            JMP     LIST
F012 C361F0            JMP     PUNCH
F015 C362F0            JMP     READER
F018 000000            DB      00,00,00        ;JMP    HOME
F01B 000000            DB      00,00,00        ;JMP    SELDSK
F01E 000000            DB      00,00,00        ;JMP    SETTRK
F021 000000            DB      00,00,00        ;JMP    SETSEC
F024 000000            DB      00,00,00        ;JMP    SETDMA
F027 000000            DB      00,00,00        ;JMP    READ
F02A 000000            DB      00,00,00        ;JMP    WRITE
F02D C349F0            JMP     LISTST
F030 000000            DB      00,00,00        ;JMP    SECTRAN
F033 C366F0            JMP     INITIO          ;CBIOS I/O EXTERNAL INITIALIZATION ENTRY
F036 C363F0            JMP     NXM             ;FOR RESTART 7: GIVE ERROR MESSAGE
F039 =         ENDTB:  EQU     $               ;END OF JUMP VECTOR
               ;
               ;
               ;
               ;
               ;########################################################
               ;
               ;       PHYSICAL DEVICE ROUTINES
               ;
               ;
               ; TELETYPE INPUT
               ;
               TTYIN:
F039 CD44F0            CALL    TTYSTAT
F03C CA39F0            JZ      TTYIN   ;WAIT FOR A CHAR TO BE AVAILABLE
F03F DB12              IN      TTY     ;INPUT IT
F041 E67F              ANI     7FH     ;REMOVE PARITY
F043 C9                RET
               ;
               TTYSTAT:                ;USED HERE AND IN CONSTAT ABOVE
F044 DB13              IN      TTYS    ;GET STATUS
F046 E602              ANI     02H     ;MASK BIT
F048 C9                RET             ;A IS NON-0 IF CHAR AVAILABLE
               ;
               ; TI 810 PRINTER DRIVER
               ;
F049 DB13      LISTST: IN      TTYS
F04B E680              ANI     80H
F04D C8                RZ              ;BUSY
F04E 3EFF              MVI     A,0FFH
F050 C9                RET             ;NOT BUSY
               LIST:
F051 CD49F0            CALL    LISTST
F054 CA51F0            JZ      LIST
F057 DB13              IN      TTYS    ;STATUS
F059 0F                RRC             ;TEST BIT 0
F05A D251F0            JNC     LIST    ;WAIT TILL READY TO ACCEPT CHARACTER
F05D 79                MOV     A,C
F05E D312              OUT     TTY     ;OUTPUT THE CHARACTER
F060 C9                RET
F061 C9        PUNCH:  RET
```

```
F062 C9        READER: RET

                ;
                ;
                ;
                ;
                ; ************************************************
                ;
                ;;      STARTUP & RESTART STUFF
                ;;
                ;;
                ;;
                ;;RESTART 7 ROUTINE. PRESUMABLY MEANS JMP TO NON-EXISTENT MEMORY
                ;;      TYPES "CRASH" AND TOP OF STACK (PRESUMED TO BE PC)
                ;;      AND BYTE TOP OF STACK POINTS TO
                ;;
F063 C30000     NXM:    JMP     WBOOT   ;POP    B               ;GET PC OF CRASH (OR MAYBE GARBAGE)
                ;       LXI     SP,100H ;SET UP STACK BELOW 100H
                ;       PUSH    B               ;SAVE THAT PC
                ;;TYPE "CRASH"
                ;       LXI     H,NXMMSG
                ;       CALL    CONOMSG
                ;;TYPE WHAT IS PROBABLY THE PC OF THE PROBLEM
                ;       POP     H               ;GET WHAT WAS ON STACK AT ENTRY TO NXM
                ;       MOV     A,H     ;HI ORDER BYTE
                ;       CALL    HOUT    ;HEX OUTPUT A
                ;       MOV     A,L     ;LO ORDER BYTE
                ;       CALL    HOUT
                ;;TYPE BYTE TOP OF STACK-1 POINTS TO: THIS MIGHT BE THE INSTRUCTION
                ; ;THAT CAUSED CRASH (RST-7, ETC)
                ;       MVI     C,' '
                ;       CALL    CONOUT  ;TYPE A SPACE
                ;       DCX     H               ;POINT ONE LESS
                ;       MOV     A,M             ;GET BYTE
                ;       CALL    HOUT    ;OUTPUT IT
                ;;REBOOT THE SYSTEM, SAME AS ANY WARM RESTART
                ;       JMP     WBOOT
                ;;
                ;;
                ;;OUT OF LINE STUFF FOR NXM
                ;;
                ;;TYPE MESSAGE HL POINTS TO ON CONSOLE.  TERMINATED BY 0 BYTE
                ;CONOMSG:
                ;       MOV     A,M     ;GET A CHAR OF MESSAGE
                ;       ORA     A               ;SET FLAGS
                ;       RZ              ;DONE IF 0 BYTE
                ;       MOV     C,A     ;TO C-REG FOR CONOUT
                ;       CALL    CONOUT  ;OUTPUT IT ON CONSOLE
                ;       INX     H               ;POINT NEXT CHARACTER
                ;       JMP     CONOMSG ;KEEP OUTPUTTING TO END
                ;
                ;;
                ;;HEX OUTPUT (A) TO CONSOLE
                ;HOUT:  PUSH    PSW
                ;       RRC
                ;       RRC
                ;       RRC
```

```
;       RRC
;       CALL    HOUTNIBL
;       POP     PSW
;HOUTNIBL:
;       ANI     0FH     ;MASK 4 BITS
;       CPI     10              ;IS IT A OR BIGGER
;       JM      HNBL1   ;IF NO
;       ADI     'A'-'0'-10      ;YES, ADD DIFFERENCE BETWEEN ASCII A AND 9+1
;HNBL1: ADI     '0'     ;CONVERT IT TO ASCII CHARACTER
;       MOV     C,A     ;TO C REGISTER FOR CONOUT
;       JMP     CONOUT  ;PRINT IT AND RETRURN
;;
;;
;;
;;      COLD BOOT INITIALIZATION
;;
;COLDBOOT:
;;
;; ;INITIALIZE IOBYT FROM SWITCHES
;;
;       IN      0FFH
;       CPI     0FFH
;       JNZ     COLDBT1
;       MVI     A,PORTFF        ;DEFAULT IOBYTE IF NO PORT FF
;COLDBT1:STA    IOBYT
;       MVI     D,0FFH          ;COLD START FLAG
;       JMP     GOCPM           ;GO COMPLETE INITIALIZATION
;;
;;
;GOCPM:
;;
;;"JMP NXM" FOR RESTART 7 (DDT WILL CHANGE IF USED).  THIS IS ACCESSED AFTER
;;  A JMP INTO NON-EXISTENT MEMORY, ALSO BY WRITE PROTECT VIOLATION ON RAM-4A
; ; IF WIRED DIRECT TO "INT" LINE BY USER. AN IMSAI EXTENSION OF BIOS FUNCTIONS.;
;       LXI     H,NXM
;       STA     038H
;       SHLD    038H+1
;;
; ;SET USER EXIT INSTRUCTION TO NOPS FOLLOWED BY A RET
;;
;       LXI     H,0     ;TWO NOPS
;       SHLD    USEREXIT
;       MVI     H,0C9H  ;A NOP AND A RET
;       SHLD    USEREXIT+2
;;
;       MOV     A,D
;       ORA     A
;       JZ      WBOOT9  ;IF WARM RESTART
;;
; ;COLD START ONLY:
;;
; ;   SIGN-ON MESSAGE
;       LXI     H,MESSAGE       ;MESSAGE TEXT IS IN BOOT RECORD
;MSLOOP:        MOV     C,M
;       CALL    CONOUT  ;USE CHAR TYPING ROUTINE IN BIOS
;       INX     H
;       MOV     A,M
```

```
                ;       ORA     A
                ;       JNZ     MSLOOP
                ;;
                ;;
                INITIO:                         ;SUBROUTINE TO INITIALIZE I/O PORTS
                ;

                ;MVI    A,PORTFF
                ;STA    IOBYT
                ;
                ;
                ; INITIALIZE BOTH CHANNELS OF IMSAI SERIAL INTERFACE BOARD
                ;
                ;IF SIO HAS JUST BEEN RESET, IT EXPECTS A "MODE" THEN A "COMMAND".
                ;BUT IF IT HASN'T BEEN RESET (WARM START), IT IS NOT EXPECTING A "MODE".
                ;SO WE SEND IT A DUMMY THAT LEAVES IT EXPECTING A COMMAND REGARDLESS,
                ;THEN A RESET COMMAND (40H), THEN DESIRED MODE AND COMMAND.
F066 2191F0             LXI     H,SIOSTRING
F069 7E                 MOV     A,M
F06A D313       SIOLUP: OUT     SIOS1
F06C D315               OUT     SIOS2
F06E 23                 INX     H
F06F 7E                 MOV     A,M
F070 B7                 ORA     A
F071 C26AF0             JNZ     SIOLUP
F074 AF                 XRA     A               ;TURN OFF INTERUPTS AND
F075 D318               OUT     SIOC            ;...CARRIER DETECT, BOTH CHANNELS
                ;
                ;
                ;SIGNON MESSAGE
F077 0E09               MVI     C,9
F079 115E00             LXI     D,SIGNON
F07C CD0500             CALL    BDOS
F07F 0E09               MVI     C,9
F081 1196F0             LXI     D,MESSAGE
F084 CD0500             CALL    BDOS

F087 C9                 RET

                ;
                ;
                ;
F088 0D0A435241NXMMSG: DB       0DH,0AH,'CRASH ',0      ;TEXT USED BY "NXM" ROUTINE
                ;
                ; BYTES TO SEND TO SIO STATUS PORTS
                ;
F091 AE405A1700SIOSTRING: DB    0AEH,40H,05AH,17H,0

                MESSAGE:
F096 0A0D285749         DB      0AH,0DH,'(WITH IOPROM IN AT '
F0AB 4630303029         DB      'F000)',0AH,0DH,24H
                ;
                ;
                ENDIOPROM:
                ;
                ;
F0B3                    END
```

```
0005 BDOS      0008 BS        000D CR        0000 CRTS      0001 CRT
0003 CTRLC     001A CTRLZ     F0B3 ENDIOPROM F039 ENDTB     F000 ENTAB
001B ESC       000C FF        F066 INITIO    000A LF        F051 LIST
F049 LISTST    F096 MESSAGE   0014 MODEM     0015 MODEMS    F063 NXM
F088 NXMMSG    F061 PUNCH     0002 RDR       0000 RDRS      F062 READER
007F RUBOUT    005E SIGNON    0010 SIOBD     0018 SIOC      F06A SIOLUP
0013 SIOS1     0015 SIOS2     F091 SIOSTRING 0001 SOH       0009 TAB
0012 TTY       F039 TTYIN     0013 TTYS      F044 TTYSTAT   005F UNDERLINE
0000 WBOOT     F003 WBOTE
```

```
                ;               BOOTSTRAP PROGRAM FOR MICROMATION
                ;
                ;               REVISED FOR CAVRO STD SYSTEM OCT 81
                ;               VERSION   2   REV  2  OCT 11, 1981
                ;
                ;       ****************************************************************
                ;
0000                    ORG     0000H
                ;
                ;       EQUATES FOR CP/M LOCATIONS
                ;       MSIZE DETERMINES THE LOCATION IN MEMORY WHERE THE BOOTSTRAP LOADS
                ;       THE CP/M OPERATING SYSTEM.
                ;
0038 =          MSIZE           EQU     56      ;SET FOR A 48K SYSTEM, CHANGE WHEN SYSTEM
                                                ;SIZE CHANGES.


F800 =          CONTROLLER      EQU     0F800H
FC00 =          BUFF            EQU     CONTROLLER+400H
FE02 =          UARTDATA        EQU     CONTROLLER+602H
FE0A =          UARTSTAT        EQU     CONTROLLER+60AH
                ;
                ;       ==>>> DO NOT SUBSTITUTE BOTH 20 (TWENTY) ON NEXT LINE !!!
9000 =          CBASE           EQU     (MSIZE-20)*1024 ;BIAS FOR SYSTEMS LARGER THAN 20K
CC06 =          BDOS            EQU     CBASE+3C06H
C400 =          CCP             EQU     CBASE+3400H     ;BASE OF CONSOLE COMMAND PROCESSOR, 16K
C380 =          CCPM            EQU     CCP-128         ;START OF LOAD TO SKIP BOOT
DA00 =          COLDBOOT        EQU     CBASE+4A00H     ;COLD START BOOT
                ;
                ;       EQUATES FOR RESERVED ROM LOCATIONS
F803 =          HOME            EQU     CONTROLLER+3
F809 =          SETTRK          EQU     CONTROLLER+9    ;SET TRACK
F80C =          SETSEC          EQU     CONTROLLER+0CH  ;SET SECTOR NUMBER
F80F =          SETDMA          EQU     CONTROLLER+0FH  ;SET DMA FUNCTION
F812 =          READ            EQU     CONTROLLER+12H  ;DISK READ
FC03 =          CONTROLBYTE     EQU     BUFF+3
FC04 =          TRACK           EQU     BUFF+4
FC0A =          SECTOR          EQU     BUFF+0AH
FC0B =          DMA             EQU     BUFF+0BH
FC0F =          TWOSIDE         EQU     BUFF+0FH
FC10 =          STEPTIME        EQU     BUFF+10H
000D =          CR              EQU     0DH
000A =          LF              EQU     0AH
                ;
                ;
0000 3180C3     BOOT:           LXI     SP,CCPM
0003 CD03F8                     CALL    HOME
0006 2100C3                     LXI     H,CCPM-80H
0009 220BFC                     SHLD    DMA
000C 0E00       RDTRK:          MVI     C,0
000E 79         RDSEC:          MOV     A,C
000F FE1A                       CPI     26
0011 CA2E00                     JZ      NXTTRK
0014 118000                     LXI     D,128
0017 2A0BFC                     LHLD    DMA
```

```
001A 19                 DAD     D
001B 0C                 INR     C
001C CD0CF8             CALL    SETSEC
001F C5                 PUSH    B
0020 220BFC             SHLD    DMA
0023 F3         RELP    DI
0024 CD12F8             CALL    READ
                ;       EI              ;COMENTED OUT
0027 C20000             JNZ     BOOT
002A C1                 POP     B
002B C30E00             JMP     RDSEC
002E 3A04FC     NXTTRK  LDA     TRACK
0031 B7                 ORA     A
0032 C23D00             JNZ     INIT
0035 0E01               MVI     C,1
0037 CD09F8             CALL    SETTRK
003A C30C00             JMP     RDTRK
003D 210AFE     INIT    LXI     H,UARTSTAT      ;HARDWARE UART INITIALIZATION ROUTINES
0040 360E               MVI     M,0EH           ;  USED FOR THE 8251 UART
0042 00                 NOP
0043 00                 NOP
0044 3640               MVI     M,40H           ;  THE NOP INSTRUCTIONS ARE NEEDED
0046 00                 NOP
0047 00                 NOP
0048 36EE               MVI     M,0EEH          ;  ONLY IF USING A 4 MHZ CPU THAT PUTS
004A 00                 NOP
004B 00                 NOP
004C 3605               MVI     M,5             ;  A 2MHZ CLOCK ON PIN 49 ON THE BUS
004E 3E05               MVI     A,5
0050 3210FC             STA     STEPTIME        ;STEPTIME FOR SHUGART DRIVES
0053 0E09               MVI     C,9
0055 115E00             LXI     D,SIGNON
0058 CD06CC             CALL    BDOS
005B C300DA             JMP     COLDBOOT


005E                    ORG     5EH


005E 0D0A       SIGNON  DB CR,LF
0060 35                 DB MSIZE/10+30H
0061 36                 DB MSIZE MOD 10 + 30H
0062 4B2043502F         DB 'K CP/M - MICROMATION VER 2.2$'
CC06 BDOS       0000 BOOT       FC00 BUFF       9000 CBASE      C400 CCP
C380 CCPM       DA00 COLDBOOT   FC03 CONTROLBYTE                 F800 CONTROLLER
000D CR         FC0B DMA        F803 HOME       003D INIT       000A LF
003B MSIZE      002E NXTTRK     000E RDSEC      000C RDTRK      F812 READ
0023 RELP       FC0A SECTOR     F80F SETDMA     F80C SETSEC     F809 SETTRK
005E SIGNON     FC10 STEPTIME   FC04 TRACK      FC0F TWOSIDE    FE02 UARTDATA
FE0A UARTSTAT
```

```
                ;
                ;
EE00            ORG     IOPROM-200H
                ;
                ;
                ;
                ;
EE00 5A32494F50DB       'Z2IOPROM  VER 1.7 REV 2 JER 10/11/81'
                ;
                ;
EE24 434156524FDB       'CAVRO CP/M BASIC I/O SYSTEM'
EE3F 434F505952DB       'COPYRIGHT (C) 1978,'
EE52 434156524FDB       'CAVRO SCIENTIFIC INSTRUMENTS'
EE6E 3132333420DB       '1234 ELKO DRIVE'
EE7D 53554E4E59DB       'SUNNYVALE, CA 94086'
                ;
                ; SYSTEM PARAMETER EQUATES
                ;
F000 =          IOPROM  EQU     0F000H          ;WHERE PROM BASIC I/O SYSTEM JUMP VECTOR STARTS
E800 =          VIOMEM  EQU     0E800H
F400 =          VIDEO   EQU     IOPROM+400H
005E =          SIGNON  EQU     05EH            ;LOCATION OF BOOTSTRAP MESSAGE
                                                ;PERSCI = 0 TO GENERATE CODE FOR SHUGART DRIVES
0001 =          TUARTIO EQU     1               ;TUARTIO = 1 TO GENERATE CODE FOR TUART BOARD
                                                ;TUARTIO = 0 TO GENERATE CODE FOR 2-SIO BOARD
00B1 =          PORTFF  EQU     0B1H            ;PORT FF INPUT DEFAULT IS VIO OUT AND
                                                ;TUARTA PARALLEL IN  CHANNEL ON TUART
F800 =          COLDBOOT  EQU   0F800H
                ;
                ;
                ;
                ;       EXTERNAL ENTRY POINTS
                ;
                ;
                ;
                ; LOWER RAM MEMORY ASSIGNMENTS
                ;
0000 =          WBOOT EQU       0               ;WHERE TO PUT JMP REBOOT
0003 =          IOBYT   EQU     3               ;INTEL COMPATIBLE I/O CONTROL BYTE
0004 =          LOGDISK  EQU    4               ;WHERE CP/M STORES LOGGEN ON DISK NO.
0005 =          ENTRYJMP EQU    5               ;WHERE TO PUT JMP ENTRYPOINT
0005 =          BDOS    EQU     5
                ;
                ; I/O PORT EQUATES
                ;
0000 =          CRTS    EQU     00H             ;CRT STATUS PORT
0001 =          CRT     EQU     01H             ;CRT I/O PORT
0002 =          RDR     EQU     02H             ;PAPER TAPE READER INPUT PORT
0000 =          RDRS    EQU     00H             ;PAPER TAPE READER STATUS
0020 =          TUART   EQU     20H             ;BASE PORT # OF TUART BOARD
0020 =          TUARTAS EQU     TUART           ;TUART PORT A STATUS
0021 =          TUARTA  EQU     TUARTAS+1       ;TUART PORT A SERIAL DATA
0022 =          TUARTAC EQU     TUARTA+1        ;TUART PORT A COMMAND
0023 =          TUARTAM EQU     TUARTAC+1       ;TUART PORT A INTERRUPT MASK
0030 =          TUARTBS EQU     TUART+10H       ;TUART PORT B STATUS
0031 =          TUARTB  EQU     TUARTBS+1       ;TUART PORT B SERIAL DATA
0032 =          TUARTBC EQU     TUARTB+1        ;TUART PORT B COMMAND
```

```
0033 =          TUARTBM EQU     TUARTBC+1       ;TUART PORT B INTERRUPT MASK
                ;
                ; I/O STATUS BITS
                ;
0080 =          TUARTBE EQU     80H             ;TUART BUFFER EMPTY
0040 =          TUARTDA EQU     40H             ;TUART DATA AVAILABLE
                ;
                ;
                ; DISK SCRATCH AREA FOR DIGITAL SYSTEMS CONTROLLER
                ;
0040 =          SCRAT   EQU     40H             ;START OF SCRATCH AREA
0040 =          TRACK   EQU     SCRAT           ;CURRENT TRACK ON DRIVE 0
0041 =          TRAK1   EQU     TRACK+1         ;CURRENT TRACK ON DRIVE 1
0042 =          SECTOR  EQU     SCRAT+2         ;CURRENTLY SELECTED SECTOR
0043 =          DMAAD   EQU     SCRAT+3         ;CURRENT DMA ADDRESS
0045 =          DISKND  EQU     SCRAT+5         ;CURRENT DISK NUMBER
0046 =          DUMMY   EQU     DISKNO+1        ;MUST BE ZERO FOR DOUBLE ADD
0047 =          ERRORS  EQU     DUMMY+1         ;ERROR COUNTER
0048 =          FLAG    EQU     ERRORS+1        ;COLD/WARM START FLAG
0049 =          LCHARI  EQU     FLAG+1          ;LAST CONSOLE INPUT CHAR
0049 =          LCHARO  EQU     LCHARI          ;LAST CONSOLE OUTPUT CHAR
004C =          USEREXIT EQU    4CH             ;USER EXIT INSTRUCTION FOR REALTIME I/O
                ;
                ; SIGN ON MESSAGE TYPED BY COLD START ROUTINE
                ;
                ;MESSAGE EQU    5CH             ;MESSAGE IN COLD BOOT RECORD
                ;
                ;
                ; EQUATES FOR ASCII CHARACTERS
                ;
0001 =          SOH     EQU     01H
0003 =          CTRLC   EQU     03H
0008 =          BS      EQU     08H
0009 =          TAB     EQU     09H
000A =          LF      EQU     0AH
000C =          FF      EQU     0CH
000D =          CR      EQU     0DH
001A =          CTRLZ   EQU     1AH
001B =          ESC     EQU     1BH
005F =          UNDERLINE EQU   5FH
007F =          RUBOUT  EQU     7FH
                ;
                ;



                ;LET US BEGIN
                ;
F000                    ORG     IOPROM          ;ORIGIN OF THIS PROGRAM
                ;
                ; ENTRY POINT TABLE
                ;
F000 C3AEF1     ENTAB   JMP     INITIO          ;COLD START.
F003 C30000     WBOTE   JMP     WBOOT           ;COME HERE TO INITIATE REBOOT (VIA LOC 0)
F006 C339F0             JMP     CONSTAT
F009 C351F0             JMP     CONIN
F00C C37EF0             JMP     CONOUT
```

```
F00F C38CF0          JMP     LIST
F012 C39CF0          JMP     PUNCH
F015 C3ADF0          JMP     READER
F018 000000          DB      00,00,00        ;       JMP     HOME
F01B 000000          DB      00,00,00        ;       JMP     SELDSK
F01E 000000          DB      00,00,00        ;       JMP     SETTRK
F021 000000          DB      00,00,00        ;       JMP     SETSEC
F024 000000          DB      00,00,00        ;       JMP     SETDMA
F027 000000          DB      00,00,00        ;       JMP     READ
F02A 000000          DB      00,00,00        ;       JMP     WRITE
F02D C31BF1          JMP     LISTST
F030 000000          DB      00,00,00        ;       JMP     SECTRAN
F033 C3AEF1          JMP     INITIO          ;CIOPROM I/O EXTERNAL INITIALIZATION ENTRY
F036 C36AF1          JMP     NXM             ;FOR RESTART 7: GIVE ERROR MESSAGE
F039 =       ENDTB:  EQU     $               ;END OF JUMP VECTOR
                     ;
                     ;##################################J################
                     ;
                     ;       LOGICAL DEVICE ROUTINES
                     ;
                     ;THESE ROUTINES USE VARIOUS PHYSICAL DEVICES
                     ; DEPENDING ON THE CONTENTS OF IOBYT
                     ;
                     ;
                     ; CONSOLE STATUS
                     ;
F039 CD41F0  CONSTAT: CALL    CONS            ;GETS STATUS OF SPECIFIC DEVICE
F03C B7              ORA     A
F03D C8              RZ                      ;IF NOT READY, RETURN 0 IN A
F03E 3EFF            MVI     A,0FFH          ;ELSE RETURN FF
F040 C9              RET
                     ;
F041 3A0300  CONS:   LDA     IOBYT   ;USE BITS 1-0 TO DETERMINE CONSOLE DEVICE
F044 CDBCF0          CALL    RLCDISPATCH
F047 CCF0            DW      TTYSTAT
F049 CEF0            DW      KEYBDS
F04B F7F0            DW      TUARTASTAT      ;2: TUART PORT A
F04D FCF0            DW      TUARTBSTAT      ;3: TUPART PORT B
                     ;
                     ;READER STATUS FOR BATCH MODE: NEVER A CHARACTER READY.
                     ;THIS IS CAUSE PRESENCE OF A CHARACTER FREQUENTLY MEANS
                     ;"ABORT WHAT YOU'RE DOING".
                     READERSTAT:
F04F AF              XRA     A
F050 C9              RET
                     ;
                     ; CONSOLE IN
                     ;
F051 3A0300  CONIN:  LDA     IOBYT
F054 CDBCF0          CALL    RLCDISPATCH
F057 CCF0            DW      TTYIN   ;0: TTY
F059 DBF0            DW      KEYBD   ;1: KEYBOARD TUART A PARALLEL
F05B 01F1            DW      TUARTAIN ;2: TUART A INPUT
F05D 0EF1            DW      TUARTBIN ;3: TUART B INPUT
                     ;
                     CONIXIT:                ;CONSOLE INPUT POST-PROCESSING
F05F F5              PUSH    PSW     ;SAVE LAST INPUT CHAR
```

```
F060 3A4900          LDA     LCHARI  ;GET PREVIOUS INPUT CHAR
F063 FE01            CPI     SOH     ;WAS IT A SOH?
F065 C279F0          JNZ     CONIX1  ;IF NOT, SKIP CHECKS
F068 F1              POP     PSW     ;RESTORE LAST INPUT CHAR
F069 FE43            CPI     'C'     ;WAS IT A C?
F06B CA00F8          JZ      0F800H  ;IF SO GO COLDBOOT CP/M
F06E FE38            CPI     '8'
F070 CA00A9          JZ      0A900H  ;RETURN TO 48K CP/M
F073 FE30            CPI     '0'
F075 CA0089          JZ      08900H  ;RETURN TO 40K CP/M
F078 F5              PUSH    PSW     ;OTHERWISE, STICK CHAR BACK ON STACK
F079 F1      CONIX1: POP     PSW     ;RESTORE LAST INPUT CHAR
F07A 324900          STA     LCHARI  ;SAVE IT UNTIL NEXT TIME
F07D C9              RET
                     ;
                     ; CONSOLE OUT
                     ;
                     ;MUST PRESERVE HL FOR NXM AND BOOTSTRAP
F07E 3A0300  CONOUT: LDA     IOBYT
F081 CDBCF0          CALL    RLCDISPATCH     ;GO TO ONE OF FOLLOWING ADDRESSES
F084 CDF0            DW      TTYOUT    ;BITS=0: USE TTY AS CONSOLE
F086 F3F0            DW      CRTOUT    ;1: CRT
F088 24F1            DW      TUARTAOUT ;2: TUART A OUTPUT
F08A 36F1            DW      TUARTBOUT ;3: TUART B OUTPUT
                     ;
                     ; LIST OUT
                     ;
F08C 3A0300  LIST:   LDA     IOBYT
F08F 07              RLC             ;BITS 7-6 TO 2-1
F090 07              RLC
F091 CDBCF0          CALL    RLCDISPATCH
F094 CDF0            DW      TTYOUT    ;0: TTY
F096 F3F0            DW      CRTOUT    ;1: CRT
F098 24F1            DW      TUARTAOUT ;2: LINE PRINTER
F09A 47F1            DW      PRINTER   ;3: TUARTAOUT PARALLEL
                     ;
                     ; PUNCH OUT
                     ;
F09C 3A0300  PUNCH:  LDA     IOBYT   ;BITS 4-5 TO 1-2
F09F 0F              RRC
F0A0 0F              RRC
F0A1 0F              RRC
F0A2 CDBDF0          CALL    DISPATCH
F0A5 CDF0            DW      TTYOUT   ;0: TTY
F0A7 F3F0            DW      PUNO     ;1: HIGH SPEED PUNCH
F0A9 69F1            DW      MODEMOUT         ;2: MODEM OUTPUT
F0AB F3F0            DW      NULLO    ;3: UNASSIGNED
                     ;
                     ; READER IN
                     ;
F0AD 3A0300  READER: LDA     IOBYT   ;BITS 3-2 TO 2-1
F0B0 0F              RRC
F0B1 CDBDF0          CALL    DISPATCH
F0B4 CCF0            DW      TTYIN    ;0: TTY
F0B6 59F1            DW      RDRIN    ;1: HIGH SPEED
F0B8 69F1            DW      MODEMIN  ;2: MODEM INPUT
F0BA 56F1            DW      NULLI    ;3: UNASSIGNED
```

```
                ;
                ;
                ;
                ;SUBROUTINE TO DISPATCH TO ONE OF 4 FOLLOWING ADDRESSES
                ;DEPENDING ON IOBYT BITS CALLER HAS POSITIONED IN
                ;BITS 2 AND 1 OF A.
                ;RETURNS TO SUBROUTINE CALL     PRIOR TO CALL   TO DISPATCH.
                ;
F0BC 07         RLCDISPATCH: RLC
F0BD E606       DISPATCH: ANI   06H     ;MASK BITS
F0BF E3                 XTHL            ;SAVE CALLER'S H, GET TABLE ADDRESS
F0C0 D5                 PUSH    D               ;##
F0C1 5F                 MOV     E,A
F0C2 1600               MVI     D,0     ;SET UP FOR DAD
F0C4 19                 DAD     D               ;INDEX INTO TABLE
F0C5 7E                 MOV     A,M
F0C6 23                 INX     H
F0C7 66                 MOV     H,M     ;TABLE WORD TO HL
F0C8 6F                 MOV     L,A     ;..
F0C9 D1                 POP     D               ;##
F0CA E3                 XTHL            ;PUT ADDRESS OF ROUTINE, GET CALLER'S H
F0CB C9                 RET             ;GO TO ROUTINE !
                ;
                ;
                ;##################################################
                ;
                ;       PHYSICAL DEVICE ROUTINES
                ;
                ;  ADDRESSED BY LOGICAL DEVICE ROUTINES ABOVE,
                ;  TTY AND CRT MAY ALSO HAVE EXTERNAL ENTRY POINTS
                ;
                ;
                ; TELETYPE INPUT
                ;
                TTYIN:
                        IF      NOT TUARTIO
                        CALL    TTYSTAT
                        JZ      TTYIN   ;WAIT FOR A CHAR TO BE AVAILABLE
                        IN      TTY     ;INPUT IT
                        ANI     7FH     ;REMOVE PARITY
                        JMP     CONIXIT ;POST-PROCESS CHAR
                        ENDIF
                ;
                TTYSTAT:                ;USED HERE AND IN CONSTAT ABOVE
                        IF      NOT TUARTIO
                        IN      TTYS    ;GET STATUS
                        ANI     02H     ;MASK BIT
                        ENDIF
F0CC C9                 RET             ;A IS NON-0 IF CHAR AVAILABLE
                ;
                ; TELETYPE OUTPUT
                ;
                ;CLOBBERS DE. BOOT DEPENDS ON PRESERVING HL
                ;MUST PRESERVE HL FOR NXM, BOOTSTRAP
                TTYOUT:
                        IF      NOT TUARTIO
                        IN      TTYS    ;STATUS
```

```
                        RRC             ;TEST BIT 0
                        JNC     TTYOUT  ;WAIT TILL READY TO ACCEPT CHARACTER
                        MOV     A,C
                        OUT     TTY     ;OUTPUT THE CHARACTER
                        CPI     CR
                        RNZ             ;DONE EXCEPT CR
                    ;DELAY 100 MSEC FOR CR, FOR SLOW-RETURNING TERMINALS
                        LXI     D,10500D
                        ENDIF
                    TTYWT1:
                        IF      NOT TUARTIO
                        DCX     D
                        ORA     D               ;DEPENDS ON A7=0 AT ENTRY TO ROUTINE
                        JP      TTYWT1  ;LOOP TAKES 9.5 USEC PER COUNT
                        ENDIF
F0CD C9                 RET
                    ;
                    ;
                    ; CRT INPUT
                    ;;
                    ;;MORE CONVENIENT CRT INPUT FOR LEAR-SIEGLER ADM-3
                    ;LSCRTIN:
                    ;       CALL    CRTIN   ;GET CHAR FROM REGULAR ROUTINE
                    ;;IGNORE BREAK KEY - ITS EASY TO FUMBLE AND HIT IT
                    ;       JZ      LSCRTIN
                    ;;CONVERT UNDERLINE (ARROW ON OLDER KEYBOARDS) TO RUBOUT
                    ;;SO IT ISN'T NECESSARY TO USE SHIFT KEY TO CORRECT ERRORS
                    ;;NOT DESIRABLE IF YOUR KEYBOARD HAS BACK ARROW.
                    ;       CPI     UNDERLINE
                    ;       JNZ     CONIXIT ;POST-PROCESS CHAR
                    ;       MVI     A,RUBOUT
                    ;       JMP     CONIXIT ;POST-PROCESS CHAR
                    ;;
                    ;; NOTE: IF TYPEING ^Z TO THE EDITOR ERASES THE SCREEN ON YOUR ADM-3,
                    ;; OPEN IT UP AND SET THE 'CLEAR SCREEN' SWITCH TO 'DISABLE'.
                    ;;
                    ;;
                    ;; CRT OUTPUT
                    ;;
                    ;;MUST PRESERVE HL FOR BOOT, NXM
                    ;;CLOBBERS DE
                    ;CRT INPUT    TUARTAIN PARALLEL
                    ;
                    KEYBDS:
F0CE F3                 DI
F0CF 3E04               MVI     A,04
F0D1 D323               OUT     23H
F0D3 DB20               IN      20H
F0D5 E620               ANI     20H
F0D7 C9                 RET
F0D8 00                 NOP
F0D9 00                 NOP
F0DA 00                 NOP
F0DB F3             KEYBD:  DI
F0DC 3E04           ST:     MVI     A,04
F0DE D323               OUT     23H
F0E0 DB20               IN      20H
```

```
F0E2 E620           ANI     20H
F0E4 CADCF0         JZ      ST
F0E7 DB23           IN      23H
F0E9 DB24           IN      24H
F0EB E67F           ANI     7FH
F0ED C35FF0         JMP     CONIXIT
F0F0 00             NOP
F0F1 00             NOP
F0F2 00             NOP
                ;;
                ;;
                ;
                CRTOUT:
                ;       MOV     A,C
                ;       CPI     0DH     ;CR?
                ;       JZ      NEWLINE ;YES
F0F3 CD00F4         CALL    VIDEO
F0F6 C9             RET
                ;
                ;
                ;
                ;       ADD SPACE TO MOVE DISPLAY OVER
                ;
                ;NEWLINE:PUSH   PSW
                ;       CALL    VIDEO
                ;       MVI     A,20H   ;LOAD BLANKS
                ;       CALL    VIDEO
                ;       CALL    VIDEO
                ;       CALL    VIDEO
                ;       CALL    VIDEO
                ;       POP     PSW
                ;       RET
                ;
                ; TUART I/O
                ;
                TUARTASTAT:                     ;TUART PORT A STATUS ROUTINE
                        IF      TUARTIO
F0F7 DB20               IN      TUARTAS         ;GET STATUS
F0F9 E640               ANI     TUARTDA         ;DATA AVAILABLE?
F0FB C9                 RET
                        ENDIF
                ;
                TUARTBSTAT:                     ;TUART PORT B STATUS ROUTINE
                        IF      TUARTIO
F0FC DB30               IN      TUARTBS         ;GET STATUS
F0FE E640               ANI     TUARTDA         ;DATA AVAILABLE?
F100 C9                 RET
                        ENDIF
                ;
                TUARTAIN:                       ;TUART PORT A INPUT ROUTINE
                        IF      TUARTIO
F101 CDF7F0             CALL    TUARTASTAT      ;CHECK STATUS
F104 CA01F1             JZ      TUARTAIN        ;LOOP TIL CHAR READY
F107 DB21               IN      TUARTA          ;READ DATA CHAR
F109 E67F               ANI     7FH             ;STRIP PARITY
F10B C35FF0             JMP     CONIXIT         ;POST PROCESS CHAR
                        ENDIF
```

```
                ;
                TUARTBIN:                       ;TUART PORT B INPUT ROUTINE
                        IF      TUARTIO
F10E CDFCF0             CALL    TUARTBSTAT      ;CHECK STATUS
F111 CA0EF1             JZ      TUARTBIN        ;LOOP TIL CHAR READY
F114 DB31               IN      TUARTB          ;READ DATA CHAR
F116 E67F               ANI     7FH             ;STRIP PARITY BIT
F118 C35FF0             JMP     CONIXIT         ;POST PROCESS CHAR
                        ENDIF
                ;
                        ;TUART PORT A OUTPUT ROUTINE

                        IF      TUARTIO
                LISTST:
                TUARTAST:
F11B DB20               IN      TUARTAS         ;CHECK STATUS
F11D F6FB               ORI     0FBH            ;IS PRINTER BUSY?
F11F 3C                 INR     A
F120 C8                 RZ                      ;YES, GO TELL'EM
F121 3EFF               MVI     A,0FFH
F123 C9                 RET                     ;NO
                TUARTAOUT:
F124 CD1BF1             CALL    TUARTAST
F127 CA24F1             JZ      TUARTAOUT       ;LOOP UNTIL NOT BUSY
F12A DB20       BUFCHK: IN      TUARTAS
F12C F67F               ORI     07FH            ;CHECK IF BUFFER IS EMPTY
F12E 3C                 INR     A
F12F C22AF1             JNZ     BUFCHK
F132 79                 MOV     A,C             ;FETCH OUTPUT CHAR
F133 D321               OUT     TUARTA          ;WRITE CHAR
F135 C9                 RET                     ;EXIT
                        ENDIF
                ;
                TUARTBOUT:                      ;TUART PORT B OUTPUT
                        IF      TUARTIO
F136 DB30               IN      TUARTBS         ;CHECK STATUS
F138 E680               ANI     TUARTBE         ;BUFFER EMPTY?
F13A CA36F1             JZ      TUARTBOUT       ;LOOP IF NOT
F13D 79                 MOV     A,C             ;FETCH OUTPUT CHAR
F13E B7                 ORA     A               ;TEST PARITY
F13F E244F1             JPO     TUARTBO1        ;SKIP IF PARITY ALREADY ODD
F142 F680               ORI     80H             ;INSERT PARITY BIT
                TUARTBO1:
F144 D331               OUT     TUARTB          ;WRITE CHAR
                        ENDIF
F146 C9                 RET                     ;EXIT
                ;
                ;
                ;
                ;               PRINTER
                ;
                ;
                ;
F147 0000000000PRINTER:         DB      00,00,00,00,00,00,00,00,00,00,00,00,00,00
F155 C9                 RET
                ;
                ;
```

```
                ; NULL DEVICE, FOR UNDEFINED DEVICES.
                ;
                ;FOR UNASSIGNED AND AND UNIMPLEMENTED INPUT DEVICES,
                ;HERE IS AN INFINITE SOURCE OF EOF'S:
F156 3E1A       NULLI:  MVI     A,CTRLZ
F158 C9                 RET
F156 =          NULLSTAT EQU NULLI      ;CHARACTER ALWAYS READY
                ;
                ;DON'T USE CRT FOR UNASS INPUT DEVICES CAUSE IF THERE
                ; IS NO CRT ON SYSTEM BUT INTERFACE BOARD IS PRESENT,
                ; SYSTEM WILL HANG.
                ;
                ;FOR UNUASS AND UNIMP OUTPUT DEVICES, USE CRT.
                ;  IF NO CRT IS PRESENT, THIS IS AN INFINITE DATA SINK.
F0F3 =          NULLO   EQU  CRTOUT
                ;
                ; HIGH SPEED READER DRIVER
                ;
F159 CD64F1     RDRIN   CALL    RDRSTAT
F15C CA59F1             JZ      RDRIN   ;WAIT FOR AVAILABLE CHARACTER
F15F DB02               IN      RDR     ;GET IT
F161 E67F               ANI     7FH     ;STRIP PARITY
F163 C9                 RET
                ;
F164 DB00       RDRSTAT IN      RDRS    ;GET STATUS
F166 E620               ANI     20H     ;MASK BIT FOR RDR
F168 C9                 RET
                ;
                ; HERE IS WHERE TO PUT HIGH SPEED PUNCH DRIVER
                ;
F0F3 =          PUND    EQU  NULLO      ;MEANWHILE, USE NULL DEVICE
                ;
                ; MODEM INPUT DRIVER
                ;
                MODEMIN:
                        IF      NOT TUARTIO
                        CALL    MODEMST
                        JZ      MODEMIN ;WAIT FOR A CHAR TO BE AVAILABLE
                        IN      MODEM   ;GET CHAR
                        ANI     7FH     ;REMOVE PARITY
                        RET
                        ENDIF
                ;
                MODEMST:
                        IF      NOT TUARTIO
                        IN      MODEMS  ;GET STATUS
                        ANI     02H     ;MASK BIT
                        RET             ;A IS NON-ZERO IF CHAR AVAILABLE
                        ENDIF
                ;
                ; MODEM OUTPUT DRIVER
                ;
                MODEMOUT:
                        IF      NOT TUARTIO
                        IN      MODEMS          ;STATUS
                        RRC                     ;TEST BIT 0
                        JNC     MODEMOUT        ;WAIT TIL READY
```

```
                        MOV     A,C
                        OUT     MODEM           ;PUT THE CHAR
                        ENDIF
F169 C9                 RET
                ;
                ;
                ; ************************************************
                ;
                ;       STARTUP & RESTART STUFF
                ;
                ;
                ;
                ; RESTART 7 ROUTINE. PRESUMABLY MEANS JMP TO NON-EXISTENT MEMORY
                ;       TYPES "CRASH" AND TOP OF STACK (PRESUMED TO BE PC)
                ;       AND BYTE TOP OF STACK POINTS TO
                ;
F16A C1         NXM:    POP     B               ;GET PC OF CRASH (OR MAYBE GARBAGE)
F16B 310001             LXI     SP,100H ;SET UP STACK BELOW 100H
F16E C5                 PUSH    B               ;SAVE THAT PC
                ;TYPE "CRASH"
F16F 21FDF1             LXI     H,NXMMSG
F172 CD8BF1             CALL    CONOMSG
                ;TYPE WHAT IS PROBABLY THE PC OF THE PROBLEM
F175 E1                 POP     H               ;GET WHAT WAS ON STACK AT ENTRY TO NXM
F176 7C                 MOV     A,H     ;HI ORDER BYTE
F177 CD96F1             CALL    HOUT    ;HEX OUTPUT A
F17A 7D                 MOV     A,L     ;LO ORDER BYTE
F17B CD96F1             CALL    HOUT
                ;TYPE BYTE TOP OF STACK-1 POINTS TO: THIS MIGHT BE THE INSTRUCTION
                ; THAT CAUSED CRASH (RST-7, ETC)
F17E 0E20               MVI     C,' '
F180 CD7EF0             CALL    CONOUT  ;TYPE A SPACE
F183 2B                 DCX     H               ;POINT ONE LESS
F184 7E                 MOV     A,M             ;GET BYTE
F185 CD96F1             CALL    HOUT    ;OUTPUT IT
                ;REBOOT THE SYSTEM, SAME AS ANY WARM RESTART
F188 C30000             JMP     WBOOT
                ;
                ;
                ;OUT OF LINE STUFF FOR NXM
                ;
                ;TYPE MESSAGE HL POINTS TO ON CONSOLE.  TERMINATED BY 0 BYTE
                CONOMSG:
F18B 7E                 MOV     A,M     ;GET A CHAR OF MESSAGE
F18C B7                 ORA     A               ;SET FLAGS
F18D C8                 RZ              ;DONE IF 0 BYTE
F18E 4F                 MOV     C,A     ;TO C-REG FOR CONOUT
F18F CD7EF0             CALL    CONOUT  ;OUTPUT IT ON CONSOLE
F192 23                 INX     H               ;POINT NEXT CHARACTER
F193 C38BF1             JMP     CONOMSG ;KEEP OUTPUTTING TO END

                ;
                ;HEX OUTPUT (A) TO CONSOLE
F196 F5         HOUT:   PUSH    PSW
F197 0F                 RRC
F198 0F                 RRC
F199 0F                 RRC
```

```
F19A 0F             RRC
F19B CD9FF1         CALL    HOUTNIBL
F19E F1             POP     PSW
                HOUTNIBL:
F19F E60F           ANI     0FH     ;MASK 4 BITS
F1A1 FE0A           CPI     10              ;IS IT A OR BIGGER
F1A3 FAA8F1         JM      HNBL1   ;IF NO
F1A6 C607           ADI     'A'-'0'-10      ;YES, ADD DIFFERENCE BETWEEN ASCII A AND 9+1
F1A8 C630       HNBL1:  ADI     '0'     ;CONVERT IT TO ASCII CHARACTER
F1AA 4F             MOV     C,A     ;TO C REGISTER FOR CONOUT
F1AB C37EF0         JMP     CONOUT  ;PRINT IT AND RETRURN
                ;
                ;       INDIVIDUAL SUBROUTINES TO PERFORM EACH FUNCTION
                ;
                ;
                ; PUT THE VARIOUS JUMPS IN LOWER RAM
                ;
                ;       MVI     A,0C3H  ;"JMP" OP CODE
                ;;"JMP REBOOT" AT 0
                ;       LXI     H,MEMT-200H+3
                ;       STA     WBOOT
                ;       SHLD    WBOOT+1
                ;;"JMP ENTRYPOINT" AT 5 FOR SYSTEM CALLS
                ;       LXI     H,ENTRYPOINT
                ;       STA     ENTRYJMP
                ;       SHLD    ENTRYJMP+1
                ;
                ;
                ;
                ;
                INITIO:

                ;       DO A DUMMY SET UP TO KILL TIME -- I DON'T KNOW WHY

                WAIT:
                ;       LXI     SP,400H
F1AE 060A           MVI     B,10D   ;10 MILLISECONDS
F1B0 2E1F       DELAY:  MVI     L,31D
F1B2 3A06FE     DELAY1: LDA     0FE06H  ;MICROMATION RDATA PORT
F1B5 2D             DCR     L
F1B6 C2B2F1         JNZ     DELAY1
F1B9 05             DCR     B
F1BA C2B0F1         JNZ     DELAY
                ;
                ; INITIALIZE BOTH CHANNELS OF CROMEMCO TUART BOARD
                ;
F1BD AF             XRA     A               ;MASK INTERRUPTS FOR BOTH PORTS
F1BE D323           OUT     TUARTAM
F1C0 D333           OUT     TUARTBM
F1C2 3E01           MVI     A,01            ;RESET BOTH PORTS
F1C4 D322           OUT     TUARTAC
F1C6 D332           OUT     TUARTBC
F1C8 3EC0           MVI     A,0C0H          ;PORTA := 9600 BAUD
                                            ;FOR 300 BUAD, USE 84
                                            ;(FOR 1200 BAUD, USE 88)
F1CA D320           OUT     TUARTAS
F1CC 3EA0           MVI     A,0A0H          ;PORT B := 4800 BUAD
```

```
F1CE D330           OUT     TUARTBS
                    ;
                    ;
                    ;
                    ;INIT Z2 IOBYTE
                    ;
F1D0 3E81           MVI A,PORTFF
F1D2 320300         STA IOBYT
                    ;
                    ;
                    ;
                    ;INITIALIZE VIO BOARD TO 80 BY 24
                    ;AND CLEAR VIDEO PARAMETER AREA
F1D5 21F0EF         LXI     H,VIOMEM+07F0H  ;CLEAR PARAMETER AREA
F1D8 0E0F           MVI     C,0FH
F1DA 3E00           MVI     A,0
F1DC 77         ZERO:   MOV     M,A
F1DD 23             INX     H
F1DE 0D             DCR     C
F1DF C2DCF1         JNZ     ZERO
F1E2 21FFEF         LXI     H,VIOMEM+7FFH   ;INITIALIZE
F1E5 3608           MVI     M,008H          ;80X24
F1E7 0E0B           MVI     C,0BH           ;CLEAR SCREEN
F1E9 CD00F4         CALL    VIDEO
                    ;
                    ;
                    ;
                    ;    SIGN-ON MESSAGE
F1EC 0E09           MVI     C,9
F1EE 115E00         LXI     D,SIGNON
F1F1 CD0500         CALL    BDOS
F1F4 0E09           MVI     C,9
F1F6 1106F2         LXI     D,MESSAGE
F1F9 CD0500         CALL    BDOS

F1FC C9             RET
                    ;
                    ;
F1FD 0D0A435241NXMMSG: DB       0DH,0AH,'CRASH ',00H    ;TEXT USED BY "NXM" ROUTINE
                    ;
                    ;
                    MESSAGE:
F206 0A0D285749     DB      0AH,0DH,'(WITH IOPROM IN AT '
F21B 46303030       DB      'F000'   ;IOPROM/1000+30H
F21F 290A0D24       DB      ')',0AH,0DH,24H

                    ;
                    ;
                    ENDIOPROM:
                    ;
                    ;
F223                END
0005 BDOS       0008 BS         F12A BUFCHK     F800 COLDBOOT   F051 CONIN
F079 CONIX1     F05F CONIXIT    F18B CONOMSG    F07E CONOUT     F041 CONS
F039 CONSTAT    000D CR         F0F3 CRTOUT     0000 CRTS       0001 CRT
0003 CTRLC      001A CTRLZ      F1B0 DELAY      F1B2 DELAY1     0045 DISKNO
F0BD DISPATCH   0043 DMAAD      0046 DUMMY      F223 ENDIOPROM  F039 ENDTB
```

```
F000 ENTAB      0005 ENTRYJMP   0047 ERRORS     001B ESC        000C FF
0048 FLAG       F1AB HNBL1      F196 HOUT       F19F HOUTNIBL   F1AE INITIO
0003 IOBYT      F000 IOPROM     F0DB KEYBD      F0CE KEYBDS     0049 LCHARI
0049 LCHARO     000A LF         F08C LIST       F11B LISTST     0004 LOGDISK
F206 MESSAGE    F169 MODEMIN    F169 MODEMOUT   F169 MODEMST    F156 NULLI
F0F3 NULLO      F156 NULLSTAT   F16A NXM        F1FD NXMMSG     0081 PORTFF
F147 PRINTER    F09C PUNCH      F0F3 PUNO       0002 RDR        F159 RDRIN
0000 RDRS       F164 RDRSTAT    F0AD READER     F04F READERSTAT
F0BC RLCDISPATCH                007F RUBOUT     0040 SCRAT      0042 SECTOR
005E SIGNON     0001 SOH        F0DC ST         0009 TAB        0040 TRACK
0041 TRAK1      F0CC TTYIN      F0CD TTYOUT     F0CC TTYSTAT    F0CD TTYWT1
0022 TUARTAC    F101 TUARTAIN   0023 TUARTAM    F124 TUARTAOUT  0020 TUARTAS
0021 TUARTA     F0F7 TUARTASTAT F11B TUARTAST   0032 TUARTBC   0080 TUARTBE
F10E TUARTBIN   0033 TUARTBM    F144 TUARTBD1   F136 TUARTBOUT  0030 TUARTBS
0031 TUARTB     F0FC TUARTBSTAT 0040 TUARTDA    0001 TUARTIO    0020 TUART
005F UNDERLINE  004C USEREXIT   F400 VIDEO      E800 VIOMEM     F1AE WAIT
0000 WBOOT      F003 WBOTE      F1DC ZERO
```

```
                ;
                ; ******** VIO VIDEO DRIVER ********
                ;
                ; VIODVR.ASM    V1.00   ACW     8/19/79
                ;               V1.10   JER     11/21/79
                ;               V1.11   JER     12/6/79 MOVED START OF LINE
                ;                                       AND FIXED ESC SEQ TO USE ASCII
                ;               V1.12   JER     7/12/81 MOVED LINE START BACK
                ;                                       AND MOVED TO IOPROM OFFSET
                ; BASED ON:
                ;
                ;       VDM DRIVER FROM:
                ;       PROCESSOR TECHNOLOGY CORP.'S
                ;       SOLOS VERSION   1.3     -  RELEASE   3/27/77
                ;       FROM CP/M USERS GROUP VOLUME 15
                ;
                ; MODIFIED BY:
                ;
                ;       ART WILLIS
                ;       267 CASITAS BULEVAR
                ;       LOS GATOS, CA 95030
                ;
                ;
                ; THIS DRIVER IS DESIGNED FOR A 24 X 80 DISPLAY, BUT
                ; CAN BE RECONFIGURED BY CHANGING N$LINE AND L$LINE.
                ;
0018 =          N$LINE  EQU     24      ;NUMBER OF LINES ON SCREEN
0050 =          L$LINE  EQU     80      ;LENGTH OF EACH LINE
                ;
                ;       SYSTEM EQUATES
                ;
E800 =          VIOMEM  EQU     0E800H
F000 =          IOPROM  EQU     0F000H
F400 =          LOCATE  EQU     IOPROM+0400H    ;OFFSET FOR 7000=9200
                ;
                ;       VIDEO DISPLAY ROUTINES
                ;
                ;
                ;   THESE ROUTINES ALLOW FOR STANDARD VIDEO TERMINAL
                ; OPERATIONS.  ON ENTRY, THE CHARACTER FOR OUTPUT IS IN
                ; REGISTER C AND ALL REGISTERS EXCEPT "A" AND FLAGS ARE
                ; UNALTERED ON RETURN.
                ;
                ;
                ;*************************************************************;
                ;       ORG     LOCATE-100H
S               :
                ; THIS IS DEBUGGING CODE TO RELOCATE THE VIO DRIVER FOR TEST USE.
                ;*************************************************************
                ;
                ;BIOS   EQU     0BE00H  ;RAM ADDRESS OF BIOS JUMP VECTOR
                ;
                ;
                ;
                ;START: LXI     D,200H
                ;       LXI     H,VIOUT
                ;       LXI     B,800H
```

```
                ;LOOP:  LDAX    D
                ;       MOV     M,A
                ;       INX     D
                ;       INX     H
                ;       DCX     B
                ;       MOV     A,C
                ;       ORA     B
                ;       JNZ     LOOP-START+100H
                ;;      LXI     H,CONIN
                ;;      SHLD    BIOS+0AH
                ;       LXI     H,VIOUT
                ;       SHLD    BIOS+0DH
                ;       CALL    PERSE
                ;       MVI     A,08H   ;SELECT MODE 2
                ;       STA     VIOMEM+7FFH
                ;;      MVI     A,081H
                ;;      STA     3       ;SET PRINTER ON SERIAL PORT
                ;;      MVI     A,088H
                ;;      OUT     20H     ;SET BAUD RATE TO 1200 (DEFAULT WAS 300)
                ;;      JMP     00H
                ;;
F400                    ORG     LOCATE
                ;
                ;CONIN: CALL    0C01FH  ;CHECK KEYBOARD STATUS
                ;       JZ      CONIN   ;LOOP TIL CHAR AVAILABLE
                ;       IN      KDATA   ;READ KEYBOARD
                ;       CPI     80H     ;IS IT A CONTROL KEY?
                ;       RC              ;NO: EXIT
                ;       ANI     2FH     ;YES: COMPRESS CODES TO 0-1F RANGE
                ;       CPI     20H     ;STILL TOO BIG?
                ;       JC      CONIN1  ;NO: SKIP ADJUSTMENT
                ;       SUI     10H     ;YES: ADJUST TO WITHIN RANGE
                ;CONIN1:        PUSH    H       ;SAVE WORKING REGISTER
                ;       LXI     H,ITAB  ;ADDRESS OF INPUT TRANSLATION TABLE
                ;       ADD     L       ;INDEX BY COMPRESSED INPUT CHAR
                ;       MOV     L,A
                ;       JNC     CONIN2  ;JIF H IS O.K.
                ;       INR     H       ;OVRFLO: MUST BUMP H
                ;CONIN2:        MOV     A,M     ;A CONTAINS XLATED CHAR
                ;       POP     H       ;RESTORE H/L
                ;       RET
                ;
F400 E5         VIOUT:  PUSH    H       ;SAVE MOST REGISTERS
F401 D5                 PUSH    D
F402 C5                 PUSH    B
F403 79                 MOV     A,C
F404 47                 MOV     B,A     ;SAVE CHAR IN B
                ;
                ; TEST IF ESC SEQUENCE HAS BEEN STARTED
                ;
F405 3AF3EF             LDA     ESCFL   ;GET ESCAPE FLAG
F408 B7                 ORA     A
F409 C23BF5             JNZ     ESCS    ;IF NON-ZERO GO PROCESS THE REST
                ;
                ;
F40C B0         CHPCK:  ORA     B       ;IS IT A NULL
F40D CA27F4             JZ      GOBK    ;DO A QUICK EXIT IF A NULL
```

```
F410 2194F5          LXI     H,TBL   ;POINT TO SPECIAL SHARACTER TABLE
F413 CD31F4          CALL    TSRCH   ;GO PROCESS
                     ;
F416 CDEEF4    GOBACK: CALL    VDADD   ;GET SCREEN ADDRESS
F419 7E              MOV     A,M     ;GET PRESENT CURSOR CHARACTER
F41A F680            ORI     80H
F41C 77              MOV     M,A     ;CURSOR IS BACK ON
F41D 2AF1EF          LHLD    SPEED-1 ;GET DELAY SPEED
F420 23              INX     H       ;MAKE SURE IT IS NON-ZERO
F421 AF              XRA     A       ;DELAY WILL END WHEN H=0
F422 2B        TIMER:  DCX     H       ;TIMER DELAYS HERE
F423 BC              CMP     H       ;DONE WITH DELAY YET
F424 C222F4          JNZ     TIMER   ;KEEP DELAYING
F427 C1        GOBK:   POP     B
F428 D1              POP     D       ;RESTORE REGISTERS
F429 E1              POP     H
F42A 79              MOV     A,C     ;SAVE CHAR
F42B 32F5EF          STA     LOCHAR
F42E C9              RET             ;EXIT FROM VDMOT
                     ;
F42F 23        NEXT:   INX     H
F430 23              INX     H
                     ;
                     ;
                     ;   THIS ROUTINE SEARCHES THROUGH A SINGLE CHARACTER
                     ; TABLE FOR A MATCH TO THE CHARACTER IN "B".  IF FOUND
                     ; A DISPATCH IS MADE TO THE ADDRESS FOLLOWING THE MATCHED
                     ; CHARACTER.  IF NOT FOUND THE CHARACTER IS DISPLAYED ON
                     ; THE MONITOR.
                     ;
F431 7E        TSRCH:  MOV     A,M     ;GET CHR FROM TABLE
F432 B7              ORA     A
F433 CA47F4          JZ      CHAR    ;ZERO IS THE LAST
F436 B8              CMP     B       ;TEST THE CHR
F437 23              INX     H       ;POINT FORWARD
F438 C22FF4          JNZ     NEXT
F43B E5              PUSH    H       ;FOUND ONE...SAVE ADDRESS
F43C CD0BF5          CALL    CREM    ;REMOVE CURSOR
F43F E3              XTHL            ;GET DISPATCH ADDRESS TO HL
                     ;
                     ; THIS IS THE DISPATCH ROUTINE
                     ; HL POINTS TO RETURN ADDRESS, BUT HL WILL BE
                     ; RESTORED FROM STACK BEFORE DISPATCH
                     ;
F440 7E        DISPT:  MOV     A,M     ;LOW ADDRESS BYTE
F441 23              INX     H
F442 66              MOV     H,M     ;HIGH ADDRESS BYTE
F443 6F              MOV     L,A     ;HL NOW COMPLETE
F444 E3        DISP1:  XTHL            ;XCHG HL WITH HL ON STACK
F445 7D              MOV     A,L     ;ALSO COPY HERE FOR SETS
F446 C9              RET             ;AND GO OFF TO THE ROUTINE
                     ;
                     ;
                     ;       PUT CHARACTER TO SCREEN
                     ;
F447 78        CHAR:   MOV     A,B     ;GET CHARACTER
F448 FE7F            CPI     7FH     ;IS IT A DEL?
```

```
F44A C8             RZ              ;GO BACK IF SO
                    ;
                    ;
                    ;
F44B =      OCHAR:  EQU     $       ;ACTUALLY PUT CHAR TO SCREEN NOW
F44B CDEEF4         CALL    VDADD   ;GET SCREEN ADDRESS
F44E 70             MOV     M,B     ;PUT CHR ON SCREEN
                    ;
F44F 3AF0EF         LDA     NCHAR   ;GET CHARACTER POSITION
F452 FE4F           CPI     L$LINE-1        ;END OF LINE?
F454 DA79F4         JC      OK
F457 3AF1EF         LDA     LINE
F45A FE17           CPI     N$LINE-1        ;END OF SCREEN?
F45C C279F4         JNZ     OK
                    ;
                    ; END OF SCREEN...ROLL UP ONE LINE
                    ;
F45F AF     SCROLL: XRA     A
F460 32F0EF         STA     NCHAR   ;BACK TO FIRST CHAR POSITION
F463 1150E8 SROL:   LXI     D,VIOMEM+L$LINE ;ADDRESS OF SECOND LINE
F466 2100E8         LXI     H,VIOMEM        ;ADDRESS OF FIRST LINE
F469 C5             PUSH    B               ;SAVE DATA CHAR
F46A 018007         LXI     B,N$LINE*L$LINE ;NUMBER OF CHARS TO MOVE
F46D 1A     SROL1:  LDAX    D               ;MOV A CHAR
F46E 77             MOV     M,A
F46F 13             INX     D               ;BUMP POINTERS
F470 23             INX     H
F471 0B             DCX     B
F472 79             MOV     A,C
F473 B0             ORA     B
F474 C26DF4         JNZ     SROL1
F477 C1             POP     B
F478 C9             RET
                    ;
                    ; INCREMENT LINE COUNTER IF NECESSARY
                    ;
F479 3AF0EF OK:     LDA     NCHAR   ;GET CHR POSITION
F47C 3C             INR     A
F47D FE50           CPI     L$LINE  ;CHECK FOR END OF LINE
F47F 32F0EF         STA     NCHAR
F482 C0             RNZ             ;DIDN'T HIT END OF LINE, OK
F483 AF             XRA     A       ;RESET CHAR POINTER
F484 32F0EF         STA     NCHAR
F487 =      PDOWN:  EQU     $       ;CURSOR DOWN ONE LINE HERE
F487 3AF1EF         LDA     LINE    ;GET THE LINE COUNT
F48A 3C             INR     A
F48B FE18   CURSC:  CPI     N$LINE  ;STORE THE NEW
F48D 32F1EF CUR:    STA     LINE    ;STORE THE NEW
F490 C0             RNZ             ;EXIT, UNLESS SCROLL NEEDED
F491 3D             DCR     A       ;LEAVE LINE COUNTER AS WAS
F492 32F1EF         STA     LINE
F495 C9             RET             ;<<< MAY NEED TO CALL SCROLL. >>>
                    ;
                    ; ERASE SCREEN
                    ;
F496 2100E8 PERSE:  LXI     H,VIOMEM        ;POINT TO SCREEN
F499 36A0           MVI     M,80H+' '       ;THIS IS THE CURSOR
```

```
F49B 23             INX   H       ;BUMP   1ST
F49C C5             PUSH  B       ;SAVE CHAR
F49D 01CF07         LXI   B,N$LINE$L$LINE+L$LINE-1        ;SCREEN LENGTH - CURSOR
F4A0 =        ERAS1:EQU   $       ;LOOPS HERE TO ERASE SCREEN
F4A0 3620           MVI   M,' '   ;BLANK IT OUT
F4A2 23             INX   H       ;NEXT
F4A3 0B             DCX   B       ;DECREMENT COUNT
F4A4 79             MOV   A,C     ;END OF SCREEN?
F4A5 B0             ORA   B
F4A6 C2A0F4         JNZ   ERAS1   ;NO: KEEP BLANKING
F4A9 C1             POP   B       ;RESTORE DATA CHAR
                    ;
F4AA 3E00     PHOME:MVI   A,0     ;RESET CURSOR--CARRY=ERASE, ELSE HOME
F4AC 32F1EF         STA   LINE    ;ZERO LINE
F4AF 32F0EF         STA   NCHAR   ;LEFT SIDE OF SCREEN
F4B2 C9       ERAS3:RET
                    ;
                    ;
F4B3 CDEEF4   CLINE:CALL  VDADD   ;GET CURRENT SCREEN ADDRESS
F4B6 3AF0EF         LDA   NCHAR   ;CURRENT CURSOR POSITION
F4B9 FE50     CLIN1:CPI   L$LINE  ;NO MORE THAN L$LINE-1
F4BB D0             RNC           ;ALL DONE
F4BC 3620           MVI   M,' '   ;ALL SPACED OUT
F4BE 23             INX   H
F4BF 3C             INR   A
F4C0 C3B9F4         JMP   CLIN1   ;LOOP TO END OF LINE
                    ;
                    ;
                    ;  ROUTINE TO MOVE THE CURSOR UP ONE LINE
                    ;
F4C3 3AF1EF   PUP:  LDA   LINE    ;GET LINE COUNT
F4C6 3D             DCR   A
F4C7 F2CCF4         JP    PUP1    ;MERGE TO HANDLE CURSOR
F4CA 3E17           MVI   A,N$LINE-1
F4CC 32F1EF   PUP1: STA   LINE
F4CF C9             RET
                    ;
                    ;  MOVE CURSOR LEFT ONE POSITION
                    ;
F4D0 3AF0EF   PLEFT:LDA   NCHAR
F4D3 3D             DCR   A
F4D4 =        PCUR: EQU   $       ;CURSOR ON SAME LINE
F4D4 32F0EF         STA   NCHAR   ;UPDATED CURSOR
F4D7 F2DCF4         JP    PLEFT1          ;EXIT UNLESS WRAP
F4DA 3E4F           MVI   A,L$LINE-1      ;LET IT WRAP
F4DC 32F0EF   PLEFT1:STA  NCHAR
F4DF C9             RET
                    ;
                    ;  CURSOR RIGHT ONE POSITION
                    ;
F4E0 3AF0EF   PRIT: LDA   NCHAR
F4E3 3C             INR   A
F4E4 FE50           CPI   L$LINE
F4E6 C2EAF4         JNZ   PRIT2
F4E9 AF       PRIT1:XRA   A
F4EA 32F0EF   PRIT2:STA   NCHAR
F4ED C9             RET
```

```
                        ;
                        ; ROUTINE TO CALCULATE SCREEN ADDRESS
                        ;
                        ;       ENTRY AT:       RETURNS:
                        ;
                        ;       VDADD   CURRENT SCREEN ADDRESS
                        ;       VDAD2   ADDRESS OF CURRENT LINE, CHAR "C"
                        ;       VDAD    LINE "A", CHARACTER POSITION 'C'
                        ;
F4EE 3AF0EF     VDADD:  LDA     NCHAR   ;GET CHARACTER POSITION
F4F1 4F                 MOV     C,A     ;'C' KEEPS IT
F4F2 3AF1EF     VDAD2:  LDA     LINE    ;LINE POSITION
F4F5 6F         VDAD:   MOV     L,A     ;INTO 'L'
F4F6 87                 ADD     A       ;TIMES TWO
F4F7 87                 ADD     A       ;TIMES FOUR
F4F8 85                 ADD     L       ;TIMES FIVE
F4F9 0F                 RRC             ;SWAP NIBBLES
F4FA 0F                 RRC
F4FB 0F                 RRC
F4FC 0F                 RRC
F4FD 6F                 MOV     L,A     ;L HAS IT
F4FE E60F               ANI     0FH     ;LOW ORDER DIGIT OF H
F500 C6E8               ADI     VIOMEM SHR 8    ;HIGH SCREEN OFFSET
F502 67                 MOV     H,A     ;NOW H IS DONE
F503 7D                 MOV     A,L     ;TWIST L'S ARM
F504 E6F0               ANI     0F0H
F506 81                 ADD     C
F507 6F                 MOV     L,A
F508 D0                 RNC
F509 24                 INR     H
F50A C9                 RET             ;H & L ARE NOW PERVERTED
                        ;
                        ;  ROUTINE TO REMOVE CURSOR
                        ;
F50B CDEEF4     CREM:   CALL    VDADD   ;GET CURRENT SCREEN ADDRESS
F50E 7E                 MOV     A,M
F50F E67F               ANI     7FH     ;STRIP OFF THE CURSOR
F511 77                 MOV     M,A
F512 C9                 RET
                        ;
                        ;  ROUTINE TO BACKSPACE
                        ;
F513 CD0DF4     PBACK:  CALL    PLEFT
F516 CDEEF4             CALL    VDADD   ;GET SCREEN ADDRESS
F519 3620               MVI     M,' '   ;PUT A BLANK THERE
F51B C9                 RET
                        ;
                        ;  ROUTINE TO PROCESS A CARRIAGE RETURN
                        ;
F51C 3AF5EF     PCR:    LDA     LOCHAR
F51F B8                 CMP     B
F520 C8                 RZ
                ;       CALL    CLINE
F521 CDE9F4             CALL    PRIT1   ;AND STORE THE NEW VALUE
                ;       CALL    PRIT    ;MOVE START OF LINE OVER FOR DUMB VIDEO
                ;       CALL    PRIT
                ;       CALL    PRIT
```

```
                ;       CALL    PRIT
F524 C9                 RET
                ;
                ;  ROUTINE TO PROCESS A LINEFEED
                ;
F525 3AF1EF     PLF:    LDA     LINE    ;GET LINE COUNT
F528 3C                 INR     A
F529 FE18               CPI     N$LINE  ;END OF SCREEN?
F52B C28DF4             JNZ     CUR     ;NO--NO NEED TO SCROLL
F52E C363F4             JMP     SROL    ;YES--THEN SCROLL
                ;
                ;  ROUTINE TO PROCESS ERASE TO END OF LINE
                ;
F531 CDB3F4     PEEOL:  CALL    CLINE   ;ERASE TO END OF LINE
F534 C9                 RET             ;... AND EXIT
                ;
                ;  SET ESCAPE PROCESS FLAG
                ;
F535 3EFF       PESC:   MVI     A,(-1) AND OFFH
F537 32F3EF             STA     ESCFL   ;SET FLAG
F53A C9                 RET
                ;
                ;  PROCESS ESCAPE SEQUENCE
                ;
F53B CD0BF5     ESCS:   CALL    CREM    ;REMOVE CURSOR
F53E CD44F5             CALL    ESCSP   ;PROCESS THE NEXT PART OF SEQUENCE
F541 C316F4             JMP     GOBACK
                ;
F544 3AF3EF     ESCSP:  LDA     ESCFL   ;GET ESCAPE FLAG
F547 FEFF               CPI     (-1) AND OFFH   ;TEST FLAG
F549 CA75F5             JZ      SECOND
                ;
                ;  PROCESS THIRD CHR OF ESC SEQUENCE
                ;
F54C 21F3EF             LXI     H,ESCFL
F54F 3600               MVI     M,0     ;NO MORE PARTS TO THE SEQUENCE
F551 FE32               CPI     '2'
F553 DA64F5             JC      SETX    ;SET X IF IS ONE
F556 CA6AF5             JZ      SETY    ;SET Y IF IS TWO
F559 FE38               CPI     '8'
F55B CA70F5             JZ      STSPD   ;SET NEW DISPLAY SPEED IF "8"
F55E FE39               CPI     '9'
F560 DA4BF4             JC      OCHAR   ;PUT IT ON THE SCREEN
F563 C0                 RNZ
                ;
                ;  TAB ABSOLUTE TO VALUE IN REG B
                ;
F564 78         SETX:   MOV     A,B     ;GET CHARACTER
F565 D620               SUI     32
F567 C3DCF4             JMP     PLEFT1
                ;
                ;  SET CURSOR TO LINE "B"
                ;
F56A 78         SETY:   MOV     A,B
F56B D620               SUI     32
F56D C3CCF4             JMP     PUP1
                ;
```

```
                ;  SET DISPLAY SPEED
                ;
F570 78         STSPD:  MOV     A,B     ;HERE TO SET DISPLAY SPEED
F571 32F2EF     DISPD:  STA     SPEED   ;SET DISPLAY SPEED
F574 C9                 RET
                ;
                ;
                ;  PROCESS SECOND CHR OF ESC SEQUENCE
                ;
F575 78         SECOND: MOV     A,B     ;GET WHICH (RECOVER CHR?)
F576 FE33               CPI     '3'
F578 CA8BF5             JZ      CURET   ;RETURN CURSOR PARAMETERS
F57B FE34               CPI     '4'
F57D C287F5             JNZ     ARET2
                ;
                ;  ESC <4>  RETURN ABSOLUTE SCREEN ADDRESS
                ;
F580 44         ARET:   MOV     B,H
F581 4D                 MOV     C,L     ;PRESENT SCREEN ADDRESS TO BC FOR RETURN
                ;
F582 E1         ARET1:  POP     H       ;RETURN ADDRESS
F583 D1                 POP     D       ;OLD B
F584 C5                 PUSH    B
F585 E5                 PUSH    H
F586 AF                 XRA     A
F587 32F3EF     ARET2:  STA     ESCFL
F58A C9                 RET
                ;
                ;
                ; ESC <3> RETURN PRESENT SCREEN PARAMETERS IN "BC"
                ;
F58B 21F0EF     CURET:  LXI     H,NCHAR
F58E 46                 MOV     B,M     ;CHARACTER POSITION
F58F 23                 INX     H
F590 4E                 MOV     C,M     ;LINE POSITION
F591 C382F5             JMP     ARET1
                ;
                ;
                ;       DISPLAY DRIVER COMMAND TABLE
                ;
                ;     THIS TABLE DEFINES THE CHARACTERS FOR SPECIAL
                ; PROCESSING.  IF THE CHARACTER IS NOT IN THE TABLE IT
                ; GOES TO THE SCREEN.
                ;
F594 0B.        TBL:    DB      CLEAR-80H       ;CLEAR SCREEN
F595 96F4               DW      PERSE
F597 17                 DB      UP-80H          ;UP CURSOR
F598 C3F4               DW      PUP
F59A 1A                 DB      DOWN-80H        ;DOWN CURSOR
F59B 87F4               DW      PDOWN
F59D 01                 DB      LEFT-80H        ;LEFT CURSOR
F59E D0F4               DW      PLEFT
F5A0 13                 DB      RIGHT-80H       ;RIGHT CURSOR
F5A1 E0F4               DW      PRIT
F5A3 0E                 DB      HOME-80H        ;HOME CURSOR
F5A4 AAF4               DW      PHOME
F5A6 0D                 DB      CR              ;CARRIAGE RETURN
```

```
F5A7 1CF5           DW      PCR
F5A9 0A             DB      LF              ;LINE FEED
F5AA 25F5           DW      PLF
F5AC 06             DB      EEOL            ;ERASE TO END OF LINE
F5AD 31F5           DW      PEEOL
F5AF 5F             DB      BACKS           ;BACKSPACE
F5B0 13F5           DW      PBACK
F5B2 1B             DB      ESC             ;ESCAPE KEY
F5B3 35F5           DW      PESC
F5B5 00             DB      0               ;END OF TABLE
                ;
                ;
                ;       VIO EQUATES
                ;
                ;
                ;       VIO PARAMETERS
                ;
                ;       KEYBOARD SPECIAL KEY ASSIGNMENTS
                ;
009A =          DOWN    EQU     9AH
0097 =          UP      EQU     97H
0081 =          LEFT    EQU     81H
0093 =          RIGHT   EQU     93H
0080 =          MODE    EQU     80H
008B =          CLEAR   EQU     8BH
008E =          HOME    EQU     08EH
005F =          BACKS   EQU     5FH     ;BACKSPACE
0006 =          EEOL    EQU     06
000A =          LF      EQU     10
000D =          CR      EQU     13
0020 =          BLANK   EQU     ' '
0020 =          SPACE   EQU     BLANK
0018 =          CX      EQU     'X'-40H
001B =          ESC     EQU     1BH
                ;
                ;       PORT ASSIGNMENTS
                ;
00FA =          STAPT   EQU     0FAH    ;STATUS PORT GENERAL
00F8 =          SERST   EQU     0F8H    ;SERIAL STATUS PORT
00F9 =          SDATA   EQU     0F9H    ;SERIAL DATA
00FA =          TAPPT   EQU     0FAH    ;TAPE STATUS PORT
00FB =          TDATA   EQU     0FBH    ;TAPE DATA
00FC =          KDATA   EQU     0FCH    ;KEYBOARD DATA
00FD =          PDATA   EQU     0FDH    ;PARALLEL DATA
00FE =          DSTAT   EQU     0FEH    ;VDM DISPLAY PARAMETER PORT
00FF =          SENSE   EQU     0FFH    ;SENSE SWITCHES
                ;
                ;
                ;       BIT ASSIGNMENT MASKS
                ;
0001 =          SCD     EQU     1       ;SERIAL CARRIER DETECT
0002 =          SDSR    EQU     2       ;SERIAL DATA SET READY
0004 =          SPE     EQU     4       ;SERIAL PARITY ERROR
0008 =          SFE     EQU     8       ;SERIAL FRAMING ERROR
0010 =          SOE     EQU     16      ;SERIAL OVERRUN ERROR
0020 =          SCTS    EQU     32      ;SERIAL CLEAR TO SEND
0040 =          SDR     EQU     64      ;SERIAL DATA READY
```

CHR 1 - FIND ESC IN TABLE
- STO -1 IN ESCFL

CHR 2 - GET ESCFL
- ORA
- NZ ⇒ ESCS (ELSE CHK TBL)
- REMOVE CURSOR
- TEST FLAG
- CPI -1 ⇒ SECOND

SECOND - RECOVER CHR &
CPI 3 - RETURN CUR. PARM
CPI 4 - RET ABS SCR. ADDR
ELSE STORE CHR
IN ESCFL & RET

CHR 3 - IF # 2 WAS NZ ESCFL
WILL TEST NZ &
GO TO ESCS
- REM CUR
- CHECK IF -1 IN ESCFL
NO
∴ MUST BE THIRD CHR

CPI 2 C ⇒ SET X (B)
Z ⇒ SET Y (B)

CPI 8 SET SPEED (B)

CPI 9 PUT CHR ON SCREEN

```
0080 =          STBE    EQU     128     ;SERIAL TRANSMITTER BUFFER EMPTY
                ;
0001 =          KDR     EQU     1       ;KEYBOARD DATA READY
0002 =          PDR     EQU     2       ;PARALLEL DATA READY
0004 =          PXDR    EQU     4       ;PARALLEL DEVICE READY
0008 =          TFE     EQU     8       ;TAPE FRAMING ERROR
0010 =          TOE     EQU     16      ;TAPE OVERRUN ERROR
0040 =          TDR     EQU     64      ;TAPE DATA READY
0080 =          TTBE    EQU     128     ;TAPE TRANSMITTER BUFFER EMPTY
                ;
                ;
                ;
                ;
                ;               VIO GLOBAL AREA
                ;
                ;
EFF0                    ORG     VIOMEM+07F0H
                ;
                ;
                ;       ***** PARAMETERS STORED IN RAM *****
                ;
EFF0 01         NCHAR   DB      1       ;CURRENT CHARACTER POSITION
EFF1 01         LINE    DB      1       ;CURRENT LINE POSITION
EFF2 01         SPEED   DB      1       ;SPEED CONTROL BYTE
EFF3 01         ESCFL   DB      1       ;ESCAPE FLAG CONTROL BYTE
EFF4 01         NUCNT   DB      1       ;NUMBER OF NULLS AFTER CRLF
EFF5 01         LOCHAR  DB      1       ;LAST OUTPUT CHAR
                ;
                ;;      INPUT TRANSLATION TABLE
                ;;
                ;ITAB:  DB      00H,13H,1AH,10H,1AH,1AH,1AH,03H
                ;       DB      1AH,1AH,1AH,0BH,0CH,1AH,0EH,1AH
                ;       DB      13H,05H,18H,1AH,03H,12H,06H,19H
                ;       DB      0CH,15H,16H,17H,1AH,07H,04H,0EH
                ;;
                ;;
EFF6                    END
F582 ARET1      F587 ARET2      F580 ARET       005F BACKS      0020 BLANK
F447 CHAR       F40C CHPCK      008B CLEAR      F4B9 CLIN1      F4B3 CLINE
F50B CREM       000D CR         F58B CURET      F48B CURSC      F48D CUR
0018 CX         F444 DISP1      F571 DISPD      F440 DISPT      009A DOWN
00FE DSTAT      0006 EEOL       F4A0 ERAS1      F4B2 ERAS3      EFF3 ESCFL
001B ESC        F53B ESCS       F544 ESCSP      F416 GOBACK     F427 GOBK
008E HOME       F000 IOPROM     00FC KDATA      0001 KDR        0081 LEFT
000A LF         EFF1 LINE       0050 LLINE      F400 LOCATE     EFF5 LOCHAR
0080 MODE       EFF0 NCHAR      F42F NEXT       0018 NLINE      EFF4 NUCNT
F44B OCHAR      F479 OK         F513 PBACK      F51C PCR        F4D4 PCUR
00FD PDATA      F487 PDOWN      0002 PDR        F531 PEEOL      F496 PERSE
F535 PESC       F4AA PHOME      F4D0 PLEFT      F4DC PLEFT1     F525 PLF
F4E0 PRIT       F4E9 PRIT1      F4EA PRIT2      F4C3 PUP        F4CC PUP1
0004 PXDR       0093 RIGHT      0001 SCD        F45F SCROLL     0020 SCTS
00F9 SDATA      0040 SDR        0002 SDSR       F575 SECOND     00FF SENSE
00F8 SERST      F564 SETX       F56A SETY       0008 SFE        0010 SOE
0020 SPACE      EFF2 SPEED      0004 SPE        F463 SROL       F46D SROL1
00FA STAPT      0080 STBE       F570 STSPD      00FA TAPPT      F594 TBL
00FB TDATA      0040 TDR        0008 TFE        F422 TIMER      0010 TOE
F431 TSRCH      0080 TTBE       0097 UP         F4F2 VDAD2      F4F5 VDAD
```

F4EE VDADD E800 VIOMEM F400 VIOUT

```
                ;
                ;PROM ROUTINES FOR MICROMATION DOUBLER, VERSION C.2



                ;THE C.1 VERSION HAS NOPS IN SYNC ROUTINE TO ALLOW MORE FREQUENT REFRESH
                ;OF DYNAMIC RAMS
                ;IT ALSO SETS UP THE SIDE BIT EARLIER TO MEET SETUP TIME FOR Y-E DATA DRIVES

                ;
                ;THIS VERSION HAS THE FOLLOWING CHANGES FROM C.1:
                ;HAS FIX FOR C.1 BUG IN SETTING UP DENSITY
                ;DISABLES INTERRUPTS AFTER FINDING CORRECT SECTOR
                ;HAS SLOWER STEP AND SETTLE TIMES
                ;

                ;               FEB 11, 1980



F800            BASE            ORG     0F800H
FC00 =          BUFF            EQU     BASE+400H       ;SCRATCHPAD RAM
                        ;
                        ;START OF HARDWARE PORT DEFINITIONS
                        ;
FE00 =          WRCONT          EQU     BASE+600H
FE01 =          WRCLK           EQU     WRCONT+1
FE02 =          WRUART          EQU     WRCONT+2
FE04 =          WRMRKCRC        EQU     WRCONT+4
FE05 =          WRMRK           EQU     WRCONT+5
FE06 =          WRDATA          EQU     WRCONT+6
FE07 =          WRCRC           EQU     WRCONT+7
FE00 =          RDSTAT          EQU     WRCONT
FE02 =          RDUART          EQU     WRCONT+2
FE04 =          RDMRKCRC        EQU     WRCONT+4
FE05 =          RDMARK          EQU     WRCONT+5
FE06 =          RDDATA          EQU     WRCONT+6
FE07 =          SYNCPORT        EQU     WRCONT+7
                        ;
                        ;START OF RAM VARIABLE DEFINITIONS
                        ;
FC00 =          ERRORBYTE       EQU     BUFF    ;NO. OF ERRORS DURING RETRIES
FC01 =          DENBYTE         EQU     BUFF+1  ;0 FOR SINGLE DENSITY
                                                ;4 FOR DOUBLE DENSITY
FC02 =          READWRITE       EQU     BUFF+2  ;0  FOR READ
                                                ;10H FOR WRITE
FC03 =          CONTROLBYTE     EQU     BUFF+3  ;RAM IMAGE OF RDSTAT OR WRCONT
FC04 =          TRACK           EQU     BUFF+4
FC05 =          PRESDISK        EQU     BUFF+5
FC06 =          LOGINTAB        EQU     BUFF+6  ;FOR EACH DRIVE
                                                ;0 IF DRIVE HAS NOT BEEN LOGGED IN
```

```
                                        ;55H IF DRIVE HAS BEEN LOGGED IN
FC0A =          SECTOR          EQU     BUFF+0AH
FC0B =          DMA             EQU     BUFF+0BH ;DMA ADDRESS
FC0D =          DISK            EQU     BUFF+0DH
FC0E =          TESTNEXT        EQU     BUFF+0EH ;55H IF WANT TO TEST DENSITY
                                        ; OF NEXT TRACK
FC0F =          TWOSIDE         EQU     BUFF+0FH
FC10 =          STEPTIME        EQU     BUFF+10H
FC11 =          ABOVE43         EQU     BUFF+11H ;10H IF (TRACK)<44D
                                        ; 50H OTHERWISE
FC12 =          TRACKTAB        EQU     BUFF+12H
FC16 =          DENMAP          EQU     BUFF+16H ;SAME CONVENTION AS DENBYTE
FC20 =          TRY1            EQU     BUFF+20H
FC21 =          RETRYCOUNT      EQU     BUFF+21H
FC22 =          CURRDRIVE       EQU     BUFF+22H
FC23 =          TESTMAX         EQU     BUFF+23H ;NO. RETRIES FOR DENSITY TEST

000F =          STEPSETTLE      EQU     15
0028 =          HEADSETTLE      EQU     40
FC40 =          STACK           EQU     BUFF+64D

                ;BEGIN WITH JUMP TABLE

F800 C3D3FB             JMP     COLDBOOT
F803 C397FB             JMP     HOME
F806 C31EFB             JMP     SELDSK
F809 C3AEFA             JMP     SETTRK
F80C C3A9FA             JMP     SETSEC
F80F C3A3FA             JMP     SETDMA
F812 C329F8             JMP     READ
F815 C32DF8             JMP     WRITE
F818 C369FA             JMP     SKEW
F81B C303FB             JMP     SETDEN

                PAGE


                WRITEPROTECT:

F81E CDBEFB             CALL    DISKREADY1      ;LOADS HEAD
                                                ;WAITS TILL DISK READY
                                                ;RETURNS (RDSTAT) IN B
F821 78                 MOV     A,B
F822 E604               ANI     04      ;WRITEPRT BIT FROM DRIVE
F824 C0                 RNZ
F825 3A05FE             LDA     RDMARK  ;RESETS HEAD LOAD COUNTER
F828 C9                 RET


                READ:                           ;ENTRY POINT FOR READ ROUTINE

F829 AF                 XRA     A               ;(READWRITE)= 00 FOR READ
F82A C32FF8             JMP     GO

                WRITE:                          ;ENTRY POINT FOR WRITE ROUTINE

F82D 3E10               MVI     A,10H           ;(READWRITE)=10H FOR WRITE
```

```
F82F 3202FC   GO:     STA     READWRITE
F832 2A01FC           LHLD    DENBYTE         ;(L)=(DENBYTE)
F835 3A03FC           LDA     CONTROLBYTE
F838 2F               CMA
F839 E6FB             ANI     0FBH            ;MASK OUT BIT 2 (SD/-DD = 0)
F83B B5               ORA     L
F83C 2F               CMA
F83D 3200FE           STA     WRCONT
F840 CDBEFB           CALL    DISKREADY1
F843 3A0AFC           LDA     SECTOR
F846 4F               MOV     C,A             ;(C)=(SECTOR)
F847 3A04FC           LDA     TRACK
F84A 47               MOV     B,A             ;(B)=(TRACK)
F84B AF               XRA     A
F84C 3200FC           STA     ERRORBYTE       ;(ERRORBYTE)= 0
F84F 7D               MOV     A,L
F850 B7               ORA     A               ;TEST FOR SINGLE DENSITY
F851 CA70F9           JZ      SD

                ;             DOUBLE DENSITY READ OR WRITE

                READDD:

F854 CD5FF9   BLOOP:  CALL    SYNC            ;SYNC ON HEADER
                                              ;FOUND HEADER
F857 360A             MVI     M,0AH           ;FIND 0A CLOCK FOR ID MARK
F859 1A               LDAX    D               ;SYNC WITH -EOW
F85A 3A04FE           LDA     RDMRKCRC
F85D FEA1             CPI     0A1H            ;DATA FOR ID MARK
F85F C254F8           JNZ     BLOOP
                                              ;FOUND ID ADDRESS MARK
                                              ;
F862 1A               LDAX    D               ;BYTE AFTER ID MARK SHOULD BE FE
F863 FEFE             CPI     0FEH
F865 C254F8           JNZ     BLOOP
                                              ;FOUND FE BYTE
                                              ;
F868 1A               LDAX    D               ;TRACK BYTE FROM DISK
F869 B8               CMP     B               ;(B)=(TRACK)
F86A C2F2F8           JNZ     TERROR1         ;TRACK ERROR

F86D 1A               LDAX    D               ;SECTOR BYTE FROM DISK
F86E B9               CMP     C               ;(C)=(SECTOR)
F86F C254F8           JNZ     BLOOP           ;WRONG SECTOR.  TRY AGAIN

F872 1A               LDAX    D
F873 F3               DI                      ;DISABLE INTERRUPTS BEFORE CHECKING ID CRC
F874 1A               LDAX    D
F875 1A               LDAX    D               ;READ 1 BYTE PAST ID CRC
F876 3A00FE           LDA     RDSTAT
F879 1F               RAR                     ;CHECK ID CRC
F87A 1A               LDAX    D
F87B DADDF8           JC      ERROR           ;ID CRC ERROR

F87E 1A               LDAX    D
F87F 3A11FC           LDA     ABOVE43
```

```
F882 47             MOV     B,A
F883 1A             LDAX    D
F884 70             MOV     M,B             ;(WRCLK)=(ABOVE43)
F885 1A             LDAX    D               ;NOW 5 BYTES INTO GAP
F886 0609           MVI     B,9
F888 1A       GLOOP: LDAX   D
F889 05             DCR     B
F88A C288F8         JNZ     GLOOP

F88D 1A             LDAX    D               ;NOW 15 BYTES INTO GAP
F88E 3A02FC         LDA     READWRITE
F891 B7             ORA     A               ;CHECK FOR WRITE
F892 1A             LDAX    D               ;16 BYTES INTO GAP
F893 C203F9         JNZ     WRITEDD
                                            ;
                                            ;DOUBLE DENSITY READ
                                            ;
F896 1A             LDAX    D
F897 1A             LDAX    D
F898 36FF           MVI     M,0FFH
F89A 1A             LDAX    D
F89B 1A             LDAX    D
F89C 1A             LDAX    D               ;21 BYTES INTO GAP
F89D 13             INX     D               ;(D)=SYNCPORT
F89E 1A             LDAX    D               ;SYNC ON FF CLOCK PATTERN
F89F 1B             DCX     D               ;(D)=RDDATA
F8A0 360A           MVI     M,0AH           ;(WRCLK)=0A
                                            ;CLOCK PATTERN FOR DATA MARK
F8A2 2A0BFC         LHLD    DMA
F8A5 1A             LDAX    D               ;SYNC WITH -EOW
F8A6 3A04FE         LDA     RDMRKCRC        ;GET DATA PATTERN FOR DATA MARK
F8A9 FEA1           CPI     0A1H
F8AB C2DDF8         JNZ     ERROR           ;MISSING DATA MARK
                                            ;
                                            ;FOUND DATA MARK
                                            ;START TRANSFERRING DATA
                                            ;
F8AE 1A       RXFER: LDAX   D
F8AF 77             MOV     M,A
F8B0 23             INX     H
F8B1 42             MOV     B,D
F8B2 1A             LDAX    D
F8B3 77             MOV     M,A
F8B4 23             INX     H
F8B5 4B             MOV     C,E
F8B6 0A             LDAX    B
F8B7 77             MOV     M,A
F8B8 23             INX     H
F8B9 1EE1           MVI     E,0E1H
F8BB 0A             LDAX    B
F8BC 77             MOV     M,A             ;4 BYTES OF DATA
F8BD 23             INX     H
F8BE 0A             LDAX    B

F8BF 77       RLOOP: MOV    M,A
F8C0 0A             LDAX    B
F8C1 1C             INR     E
```

```
F8C2 23          INX   H
F8C3 77          MOV   M,A
F8C4 0A          LDAX  B
F8C5 23          INX   H
F8C6 77          MOV   M,A
F8C7 0A          LDAX  B
F8C8 23          INX   H
F8C9 77          MOV   M,A
F8CA 23          INX   H
F8CB 0A          LDAX  B
F8CC C2BFF8      JNZ   RLOOP        ;HAVE TRANSFERRED 128 BYTES
                                    ;AND HAVE READ 129TH BYTE

F8CF 0A          LDAX  B
F8D0 0A          LDAX  B            ;READ 1 BYTE PAST CRC
F8D1 3A00FE      LDA   RDSTAT
F8D4 1F          RAR                ;CHECK DATA CRC
F8D5 DADDF8      JC    ERROR        ;DATA CRC ERROR
                 ;
                 ;SUCCESSFUL SECTOR READ
                 ;
F8D8 AF          XRA   A            ;RETURN 00 IN ACCUMULATOR
F8D9 3201FE      STA   WRCLK
F8DC C9          RET

         ERROR:

         ;ARRIVE HERE ON ANY OF FOLLOWING CONDITIONS
         ;       30H TRACK ERRORS
         ;       ID CRC ERROR
         ;       MISSING DATA MARK
         ;       DATA CRC ERROR

F8DD 3EEF        MVI   A,0EFH       ;RETURN EFH IN ACC
F8DF B7          ORA   A            ;       (UNSUCCESSFUL READ)
F8E0 3201FE      STA   WRCLK
F8E3 C9          RET

         TERROR:

         ;ARRIVE HERE ON TRACK ERROR IN SINGLE DENSITY

F8E4 CDFBF8      CALL  ERRORCOUNT   ;INCREMENT ERRORBYTE
F8E7 C270F9      JNZ   ALOOP        ;TRY AGAIN IF LESS THAN 30H

F8EA 3EEF    NO  MVI   A,0EFH       ;30H TRACK ERRORS
F8EC B7          ORA   A            ;RETURN EFH IN ACC
F8ED 37          STC                ;(UNSUCESSFUL DISK OPERATION)
F8EE 3201FE      STA   WRCLK
F8F1 C9          RET

         TERROR1:

         ;ARRIVE HERE ON TRACK ERROR IN DOUBLE DENSITY

F8F2 CDFBF8      CALL  ERRORCOUNT   ;INCREMENT ERRORBYTE
F8F5 C254F8      JNZ   BLOOP        ;TRY AGAIN IF LESS THAN 30H
```

```
F8F8 C3EAF8          JMP     NO

F8FB 2100FC    ERRORCOUNT    LXI     H,ERRORBYTE
F8FE 34                      INR     M               ;INCREMENT ERRORBYTE
F8FF 7E                      MOV     A,M
F900 FE30                    CPI     30H
F902 C9                      RET

               WRITEDD:

               ;DOUBLE DENSITY WRITE
               ;ARRIVE HERE 16 BYTES AFTER ID FIELD

F903 3E4E            MVI     A,4EH
F905 12              STAX    D               ;WRITE 4 BYTES OF 4E
F906 12              STAX    D
F907 12              STAX    D
F908 12              STAX    D
F909 AF              XRA     A
F90A 12              STAX    D               ;WRITE 6 BYTES OF 00
F90B 12              STAX    D
F90C 2A0BFC          LHLD    DMA
F90F 12              STAX    D
F910 12              STAX    D
F911 0104FE          LXI     B,WRMRKCRC
F914 12              STAX    D
F915 12              STAX    D
F916 3EA1            MVI     A,0A1H
F918 02              STAX    B               ;WRITE DATA MARK (A1)
F919 0EE1            MVI     C,0E1H

               ;START WRITING DATA TO DISK FROM MEMORY

F91B 7E        WXFER: MOV    A,M
F91C 12        WLOOP: STAX   D
F91D 23              INX     H
F91E 0C              INR     C
F91F 7E              MOV     A,M
F920 12              STAX    D
F921 23              INX     H
F922 7E              MOV     A,M
F923 12              STAX    D
F924 23              INX     H
F925 7E              MOV     A,M
F926 23              INX     H
F927 12              STAX    D
F928 7E              MOV     A,M
F929 C21CF9          JNZ     WLOOP

                     ;WHEN WE ARRIVE HERE WE'VE WRITTEN
                     ; 31#4=124 BYTES TO DISK

F92C 12              STAX    D
F92D 23              INX     H
F92E 7E              MOV     A,M
F92F 12              STAX    D
F930 23              INX     H
```

```
F931 7E             MOV     A,M
F932 23             INX     H
F933 12             STAX    D
F934 7E             MOV     A,M
F935 12             STAX    D               ;128TH BYTE TO DISK
F936 3EFF           MVI     A,0FFH
F938 3207FE         STA     WRCRC           ;WRITE 2 BYTES OF DATA CRC
F93B 3207FE         STA     WRCRC
F93E 12             STAX    D               ;WRITE 3 BYTES OF FF
F93F 12             STAX    D
F940 12             STAX    D
F941 AF             XRA     A               ;RETURN 00 IN ACC
F942 3201FE         STA     WRCLK           ;(SUCCESSFUL WRITE)
F945 C9             RET

                    ;SINGLE DENSITY ROUTINES
                    ;ENTRY POINT IS SD  (BELOW)

            WRITESD:        ;ARRIVE HERE 6 BYTES PAST ID FIELD
F946 3EFF           MVI     A,0FFH
F948 12             STAX    D               ;WRITE 3 BYTES FF (BYTES 7,8,9)
F949 12             STAX    D
F94A 12             STAX    D
F94B AF             XRA     A
F94C 12             STAX    D               ;WRITE 6 BYTES 00 (BYTES 10-15)
F94D 12             STAX    D
F94E 2A0BFC         LHLD    DMA
F951 12             STAX    D
F952 12             STAX    D
F953 12             STAX    D
F954 12             STAX    D               ;BYTE 15 OF GAP
F955 3EFB           MVI     A,0FBH          ;WRITE DATA MARK FOR SINGLE DEN
F957 3204FE         STA     WRMRKCRC
F95A 0EE1           MVI     C,0E1H
F95C C31BF9         JMP     WXFER           ;JUMP TO COMMON WRITE ROUTINE

            SYNC:

            ;ROUTINE TO SYNC ON HEADER

F95F 2101FE         LXI     H,WRCLK
F962 36FF           MVI     M,0FFH
F964 1107FE         LXI     D,SYNCPORT
F967 1A     CLOOP:  LDAX    D               ;SYNC ON FF CLOCK IN HEADER
F968 B7             ORA     A               ;SHOULD HAVE 00 DATA
                                            ;FOUND SYNC PATTERN
F969 00             NOP
F96A 00             NOP
F96B 1B             DCX     D               ;(D)=WRDATA=READDATA
F96C C8             RZ
F96D C35FF9         JMP     SYNC


                    ;SINGLE DENSITY ENTRY POINT

            SD:
F970 CD5FF9 ALOOP:  CALL    SYNC
```

```
                                        ;FOUND HEADER
F973 36C7      MLOOP:  MVI     M,0C7H   ;CLOCK PATTERN FOR ID MARK
F975 3A04FE    LLOOP:  LDA     RDMRKCRC
F978 B7                ORA     A
F979 CA75F9            JZ      LLOOP
F97C FEFE              CPI     0FEH
F97E CA8DF9            JZ      NLOOP
F981 36FF              MVI     M,0FFH
F983 3A07FE            LDA     SYNCPORT
F986 B7                ORA     A
F987 CA73F9            JZ      MLOOP
F98A C370F9            JMP     ALOOP
               NLOOP:                   ;FOUND DATA MARK
F98D 1A                LDAX    D        ;TRACK BYTE FROM DISK
F98E B8                CMP     B
F98F C2E4F8            JNZ     TERROR   ;TRACK ERROR
F992 1A                LDAX    D        ;SIDE BYTE FROM DISK (IGNORE)
F993 1A                LDAX    D        ;SECTOR BYTE FROM DISK
F994 B9                CMP     C
F995 C270F9            JNZ     ALOOP    ;WRONG SECTOR.  TRY AGAIN

                                        ;FOUND CORRECT TRACK AND SECTOR
F998 F3                DI               ;DISABLE INTERRUPTS BEFORE CHECKING ID CRC
F999 1A                LDAX    D
F99A 1A                LDAX    D        ;CRC BYTE
F99B 1A                LDAX    D        ;CRC BYTE
F99C 1A                LDAX    D        ;GAP BYTE 1
F99D 3A00FE            LDA     RDSTAT   ;CHECK ID CRC
F9A0 1F                RAR
F9A1 1A                LDAX    D        ;GAP BYTE 2
F9A2 1A                LDAX    D        ;GAP BYTE 3
F9A3 DADDF8            JC      ERROR    ;ID CRC ERROR

F9A6 1A                LDAX    D        ;GAP BYTE 4
F9A7 3A11FC            LDA     ABOVE43
F9AA 77                MOV     M,A
F9AB 1A                LDAX    D        ;GAP BYTE 5
F9AC 3A02FC            LDA     READWRITE
F9AF B7                ORA     A        ;CHECK FOR WRITE
F9B0 1A                LDAX    D        ;GAP BYTE 6
F9B1 C246F9            JNZ     WRITESD

               ;SINGLE DENSITY READ

F9B4 1A                LDAX    D        ;READ 6 BYTES OF GAP
F9B5 1A                LDAX    D
F9B6 1A                LDAX    D
F9B7 1A                LDAX    D
F9B8 1A                LDAX    D
F9B9 1A                LDAX    D
F9BA 36FF              MVI     M,0FFH   ;(WRCLK)=FF
F9BC 0106FE            LXI     B,RDDATA
F9BF 1A                LDAX    D        ;GAP BYTE 14
F9C0 13                INX     D        ;(D)=SYNCPORT
F9C1 1A                LDAX    D
F9C2 36C7              MVI     M,0C7H   ;CLOCK PATTERN FOR DATA MARK
F9C4 1E04              MVI     E,04     ;(D)=RDMRKCRC
```

```
F9C6 0A             LDAX    B           ;GAP BYTE 16
F9C7 1A             LDAX    D           ;READ DATA MARK
F9C8 E6FC           ANI     0FCH
F9CA FEF8           CPI     0F8H        ;DATA PATTERN FOR DATA MARK
F9CC C2DDF8         JNZ     ERROR       ;MISSING DATA MARK

                ;FOUND SINGLE DENSITY DATA MARK

F9CF 1EE0           MVI     E,0E0H      ;32*4=128 BYTE TRANSFER
F9D1 0A             LDAX    B
F9D2 2A0BFC         LHLD    DMA
F9D5 C3BFF8         JMP     RLOOP       ;JUMP TO MAIN READ ROUTINE


                TEST:

                ;TESTS DENSITY OF DISKETTE IN LOGGED-IN DRIVE
                ;RETURNS 00 IN ACC IF DOUBLE DENSITY
                ;RETURNS 0F IN ACC IF SINGLE DENSITY
                ;RETURNS 0A IN ACC IF TEST FAILS

F9D8 AF             XRA     A
F9D9 3223FC         STA     TESTMAX     ;(TESTMAX)=0
F9DC AF         TEST1:  XRA     A
F9DD 3200FC         STA     ERRORBYTE   ;(ERRORBYTE)=0
F9E0 CDC5FB         CALL    DISKREADY   ;LOAD HEAD
F9E3 0100FE         LXI     B,WRCONT
F9E6 3A03FC         LDA     CONTROLBYTE
F9E9 F680           ORI     80H         ;SET CONTROLLER FOR SIDE 0
F9EB E6FB           ANI     0FBH        ;TRY DOUBLE DENSITY
F9ED 02             STAX    B

                LOOP6:      ;DOUBLE DENSITY TEST
F9EE 2101FE         LXI     H,WRCLK
F9F1 36FF           MVI     M,0FFH
F9F3 1107FE         LXI     D,SYNCPORT  ;SYNC ON FF CLOCK IN HEADER
F9F6 1A         LOOP7:  LDAX    D           ;READ DATA PATTERN
F9F7 2C             INR     L           ;ABORT AFTER 256 TRIES
F9F8 CA1AFA         JZ      RETRY
F9FB B7             ORA     A           ;DATA SHOULD BE 00
F9FC C2F6F9         JNZ     LOOP7
                                        ;FOUND HEADER
F9FF 1B             DCX     D           ;(D)=READDATA
FA00 2E01           MVI     L,01        ;(H)=WRCLK
FA02 360A           MVI     M,0AH
FA04 1A             LDAX    D           ;SYNC WITH -EOW
FA05 3A04FE         LDA     RDMRKCRC    ;LOOK FOR ID MARK
FA08 FEA1           CPI     0A1H
FA0A C21AFA         JNZ     RETRY
                                        ;FOUND ID MARK
FA0D 1A             LDAX    D           ;FE BYTE
FA0E 1A             LDAX    D           ;TRACK BYTE
FA0F 1A             LDAX    D           ;SECTOR BYTE
FA10 1A             LDAX    D           ;CRC BYTE
FA11 1A             LDAX    D           ;CRC BYTE
FA12 1A             LDAX    D           ;GAP BYTE 1
FA13 0A             LDAX    B
```

```
FA14 1F            RAR                 ;CHECK ID CRC
FA15 DA1AFA        JC      RETRY
                                       ;ID CRC OK
FA18 AF            XRA     A           ;RETURN 00
FA19 C9            RET

FA1A CDFBF8  RETRY:  CALL    ERRORCOUNT
FA1D C2EEF9        JNZ     LOOP6

             ;SINGLE DENSITY TEST
             ;ARRIVE HERE AFTER 30H TRIES AT DOUBLE DENSITY

FA20 AF      SDTEST: XRA     A
FA21 3200FC        STA     ERRORBYTE   ;(ERRORBYTE)=0
FA24 3A03FC        LDA     CONTROLBYTE
FA27 F684          ORI     84H         ;SET UP SIDE 0, SINGLE DENSITY
FA29 02            STAX    B           ;TO WRCONT
             SDLOOP1:
FA2A 1E07          MVI     E,07        ;(D)=SYNCPORT
FA2C 2101FE        LXI     H,WRCLK
FA2F 36FF          MVI     M,0FFH      ;SYNC ON FF CLOCK PATTERN
             SDLOOP2:
FA31 1A            LDAX    D           ;GET CORRESPONDING DATA
FA32 2C            INR     L           ;ABORT AFTER 256 TRIES
FA33 CA57FA        JZ      RETRY1
FA36 B7            ORA     A           ;DATA SHOULD BE 00
FA37 C231FA        JNZ     SDLOOP2
                                       ;FOUND HEADER
FA3A 1B            DCX     D           ;(D)=READDATA
FA3B 2E01          MVI     L,01        ;(H)=WRCLK
FA3D 36C7          MVI     M,0C7H      ;LOOK FOR C7 CLOCK
FA3F 1A            LDAX    D           ;SYNC WITH -EOW
FA40 3A04FE        LDA     RDMRKCRC
FA43 FEFE          CPI     0FEH        ;DATA FOR ID MARK
FA45 C257FA        JNZ     RETRY1
                                       ;FOUND ID MARK
FA4B 1A            LDAX    D           ;TRACK BYTE
FA49 1A            LDAX    D           ;SIDE
FA4A 1A            LDAX    D           ;SECTOR
FA4B 1A            LDAX    D
FA4C 1A            LDAX    D           ;CRC BYTE
FA4D 1A            LDAX    D           ;CRC BYTE
FA4E 1A            LDAX    D
FA4F 0A            LDAX    B           ;GET RDSTAT
FA50 1F            RAR                 ;CHECK ID CRC
FA51 DA57FA        JC      RETRY1
                                       ;ID CRC OK
FA54 F6FF          ORI     0FFH        ;RETURN FF
FA56 C9            RET

FA57 CDFBF8  RETRY1: CALL    ERRORCOUNT
FA5A C22AFA        JNZ     SDLOOP1

                   ;FAILED BOTH DOUBLE AND SINGLE DENSITY
                   ; TESTS 30H TIMES

FA5D 2123FC        LXI     H,TESTMAX
```

```
FA60 34            INR   M             ;INCREMENT TESTMAX
FA61 7E            MOV   A,M
FA62 FE0A          CPI   10
FA64 C2DCF9        JNZ   TEST1
                   ;FAILED TEST 10 TIMES
FA67 B7            ORA   A             ;RETURN 0A
FA68 C9            RET


           SKEW:
                   ;COMPUTES PHYSICAL SECTOR FROM LOGICAL SECTOR
                   ;SKEW FACTOR IS 8
                   ;INPUT AND OUTPUT ARE IN C REG
                   ;OUTPUT=(((INPUT) MOD 52)*8 - 7) MOD 52
                   ;IF INPUT>52, SELECTS SIDE 1

FA69 210000        LXI   H,0
FA6C E5            PUSH  H
FA6D 3A03FC        LDA   CONTROLBYTE
FA70 E67F          ANI   7FH           ;SIDE 1
FA72 5F            MOV   E,A
FA73 79            MOV   A,C
FA74 D634          SUI   52
FA76 47            MOV   B,A           ;(B)=(C)-52
FA77 7B            MOV   A,E           ;(A)=(CONTROLBYTE)^7F
FA78 F27EFA        JP    SKIPY
                                       ;INPUT WAS LESS THAN 52
FA7B F680          ORI   80H           ;CHOOSE SIDE 0
FA7D 41            MOV   B,C
FA7E 320FFC  SKIPY: STA   TWOSIDE
FA81 78            MOV   A,B           ;(B)=(INPUT) MOD 52
FA82 68            MOV   L,B
FA83 C1            POP   B
FA84 0C      LOOP10: INR   C
FA85 D60D          SUI   13
FA87 F284FA        JP    LOOP10
FA8A 29            DAD   H
FA8B 29            DAD   H
FA8C 29            DAD   H
FA8D 7C            MOV   A,H
FA8E B7            ORA   A
FA8F 7D            MOV   A,L
FA90 C4A0FA        CNZ   HIGHE
FA93 FE34    LOOP11: CPI   52
FA95 DA9DFA        JC    SKIP12
FA98 C6CC          ADI   204
FA9A C393FA        JMP   LOOP11
FA9D 81      SKIP12: ADD   C
FA9E 4F            MOV   C,A
FA9F C9            RET
FAA0 C630    HIGHE:  ADI   48
FAA2 C9            RET


FAA3 60      SETDMA: MOV   H,B
FAA4 69              MOV   L,C
FAA5 220BFC          SHLD  DMA           ;STORE DMA ADDRESS
```

```
FAA8 C9             RET


FAA9 79       SETSEC: MOV     A,C
FAAA 320AFC           STA     SECTOR          ;STORE SECTOR NUMBER
FAAD C9               RET


              SETTRK:         ;STEPS DRIVE TO TRACK (C)
FAAE 79               MOV     A,C
FAAF FE2C             CPI     44D             ;IF (C)<44
FAB1 3E10             MVI     A,10H           ;  THEN (ABOVE43)=10H
FAB3 DAB8FA           JC      SKIP3
FAB6 3E50             MVI     A,50H           ;  ELSE (ABOVE43)=50H
FAB8 3211FC   SKIP3:  STA     ABOVE43
FABB CDC5FB           CALL    DISKREADY

              STEPLOOP:
FABE 2104FC           LXI     H,TRACK
FAC1 7E               MOV     A,M             ;GET (TRACK)
FAC2 B9               CMP     C               ;DONE?
FAC3 CAEAFA           JZ      DONESTEP
FAC6 CDCCFA           CALL    STEPHEAD        ;NO, STEP HEAD
FAC9 C3BEFA           JMP     STEPLOOP        ;REPEAT

              STEPHEAD:
FACC DAD8FA            JC     STEPIN          ;IF (TRACK)<(C) THEN STEP IN
              STEPOUT :
FACF 3A03FC           LDA     CONTROLBYTE     ;ELSE STEP OUT
FAD2 35               DCR     M               ;(TRACK)=(TRACK)-1
FAD3 F602             ORI     02H             ;DIR=OUT
FAD5 C3DEFA           JMP     DOSTEP

FAD8 3A03FC   STEPIN: LDA     CONTROLBYTE
FADB 34               INR     M               ;(TRACK)=(TRACK)+1
FADC E6FD             ANI     0FDH            ;DIR=IN

FADE 12       DOSTEP: STAX    D               ;STORE DIRECTION IN WRCONT
FADF 3D               DCR     A               ;-STEP=0
FAE0 12               STAX    D
FAE1 3C               INR     A               ;-STEP=1
FAE2 12               STAX    D
FAE3 3A10FC           LDA     STEPTIME
FAE6 47               MOV     B,A             ;WAIT 8 MS FOR NEXT STEP
FAE7 C37DFB           JMP     DELAY           ;DELAY EXECUTES A RETURN

              DONESTEP:
FAEA 060F             MVI     B,STEPSETTLE
FAEC CD7DFB           CALL    DELAY           ;WAIT 8 MS FOR STEP SETTLE
FAEF 79               MOV     A,C
FAF0 FE02             CPI     2               ;IF (TRACK)<2 THEN SET TESTNEXT
FAF2 DA69FB           JC      SETTN
FAF5 3A0EFC           LDA     TESTNEXT
FAF8 B7               ORA     A
FAF9 3E00             MVI     A,0
FAFB 320EFC           STA     TESTNEXT
FAFE 37               STC
```

```
FAFF C203FB          JNZ     SETDEN          ;IF TESTNEXT=55 TEST DENSITY
FB02 C9              RET

             SETDEN:

                     ;TESTS DENSITY
                     ;UPDATES DENBYTE AND DENMAP

FB03 CDD8F9          CALL    TEST            ;TEST DENSITY
FB06 3E04            MVI     A,4             ;IF Z IS SET (DOUBLE DENSITY)
FB08 CA0DFB          JZ      SKIP            ;  THEN (DENBYTE)=4
FB0B 3E00            MVI     A,0             ;  ELSE (DENBYTE)=0
FB0D 3201FC    SKIP: STA     DENBYTE
FB10 2116FC          LXI     H,DENMAP
FB13 F5              PUSH    PSW
FB14 3A05FC          LDA     PRESDISK
FB17 4F              MOV     C,A
FB18 0600            MVI     B,0
FB1A 09              DAD     B
FB1B F1              POP     PSW             ;SAVE FLAGS
FB1C 77              MOV     M,A             ;(DENMAP(PRESDISK))=(DENBYTE)
FB1D C9              RET

             SELDSK:

                     ;SELECTS DRIVE POINTED TO BY C REG
                     ;LOADS HEAD OF SELECTED DRIVE

FB1E 21F9FB          LXI     H,MASKTABLE
FB21 0600            MVI     B,0
FB23 09              DAD     B               ;C CONTAINS DRIVE NUMBER
FB24 7E              MOV     A,M             ;MASKTABLE CONTAINS 0 FOR
             SELDSK1:
FB25 3200FE          STA     WRCONT          ; SELECTED DRIVE, 1'S ELSEWHERE
FB28 320FFC          STA     TWOSIDE
FB2B 3203FC          STA     CONTROLBYTE
FB2E 2112FC          LXI     H,TRACKTAB
FB31 3A05FC          LDA     PRESDISK
FB34 5F              MOV     E,A
FB35 50              MOV     D,B
FB36 19              DAD     D
FB37 3A04FC          LDA     TRACK
FB3A 77              MOV     M,A             ;(TRACKTAB(PRESDISK))=(TRACK)
FB3B 79              MOV     A,C
FB3C 3205FC          STA     PRESDISK        ;(PRESDISK)=(C)
FB3F 320DFC          STA     DISK            ;(DISK)=(C)
FB42 2112FC          LXI     H,TRACKTAB
FB45 09              DAD     B
FB46 7E              MOV     A,M
FB47 3204FC          STA     TRACK           ;(TRACK)=(TRACKTAB(C))
FB4A 2106FC          LXI     H,LOGINTAB
FB4D 09              DAD     B
FB4E 7E              MOV     A,M
FB4F B7              ORA     A               ;HAS DRIVE BEEN LOGGED IN?
FB50 C259FB          JNZ     INOK
FB53 3E55            MVI     A,55H           ;NO. MARK AS LOGGED IN
FB55 77              MOV     M,A
```

```
FB56 CD97FB          CALL    HOME            ;  AND HOME THE HEAD
FB59 CDC5FB  INOK:   CALL    DISKREADY       ;LOAD HEAD
FB5C 0628            MVI     B,HEADSETTLE
FB5E CD7DFB          CALL    DELAY           ;WAIT FOR HEAD SETTLING
FB61 3A04FC          LDA     TRACK
FB64 FE02            CPI     02
FB66 D203FB          JNC     SETDEN
FB69 3E55    SETTN:  MVI     A,55H           ;ON TRACKS 0 AND 1, WE WANT
FB6B 320EFC          STA     TESTNEXT        ; TO TEST DENSITY OF NEXT TRACK
FB6E C303FB          JMP     SETDEN          ;TEST DENSITY OF THIS TRACK


             HEADLOAD:
FB71 1A               LDAX   D               ;ASSUMES (D)=RDSTAT
FB72 E620            ANI     20H             ;HEAD ALREADY LOADED?
FB74 3A05FE          LDA     RDMARK          ;RESET HEAD LOAD COUNTER
FB77 0628            MVI     B,HEADSETTLE
FB79 C47DFB          CNZ     DELAY           ;IF HEAD WASN'T LOADED
FB7C C9              RET

             DELAY:

                     ;DELAYS (B) MILLISECONDS

FB7D E5              PUSH    H               ;SAVE HL
FB7E 3A03FC  DELAY2: LDA     CONTROLBYTE
FB81 E604            ANI     4               ;IF SINGLE DENSITY,
FB83 2E1F            MVI     L,31            ;31 BYTES * 32 USEC = 1 MS
FB85 C28AFB          JNZ     DELAY1
FB88 2E3F            MVI     L,63            ;IN DD, 63 BYTES * 16 USEC = 1 MS
FB8A 3A06FE  DELAY1: LDA     RDDATA
FB8D 2D              DCR     L
FB8E C28AFB          JNZ     DELAY1
FB91 05              DCR     B               ;END 1 MS LOOP
FB92 C27EFB          JNZ     DELAY2
FB95 E1              POP     H               ;RESTORE HL
FB96 C9              RET


FB97 CDC5FB  HOME:   CALL    DISKREADY
FB9A 2104FC          LXI     H,TRACK         ;FOR STEPIN AND STEPOUT
FB9D CDD8FA  ATHOME: CALL    STEPIN          ;STEP TOWARD 76
FBA0 1A              LDAX    D
FBA1 E602            ANI     02              ;  UNTIL -TRK0 IS INACTIVE
FBA3 C2A9FB          JZ      ATHOME
FBA6 CDCFFA  GOHOME: CALL    STEPOUT         ;THEN STEP TOWARD 00
FBA9 1A              LDAX    D
FBAA E602            ANI     02              ;  UNTIL -TRK0 IS ACTIVE
FBAC C2A6FB          JNZ     GOHOME
FBAF 3E10            MVI     A,10H
FBB1 3211FC          STA     ABOVE43         ;(ABOVE43)=10H
FBB4 320EFC          STA     TESTNEXT        ;(TESTNEXT)=10H
FBB7 AF              XRA     A
FBB8 3204FC          STA     TRACK           ;(TRACK)=00
FBBB C303FB          JMP     SETDEN          ;TEST DENSITY
```

```
                DISKREADY1:
 FBBE 3A00FE            LDA     RDSTAT
 FBC1 47                MOV     B,A
 FBC2 E6A0              ANI     0A0H            ;IF DRIVE READY AND HEAD LOADED
 FBC4 C8                RZ                      ;       THEN RETURN
                DISKREADY:
 FBC5 C5                PUSH    B
 FBC6 1100FE            LXI     D,WRCONT        ;(D)=WRCONT=RDSTAT
 FBC9 CD71FB            CALL    HEADLOAD        ;LOAD HEAD
 FBCC C1                POP     B
 FBCD 1A                LDAX    D
 FBCE 07                RLC
 FBCF DAC5FB            JC      DISKREADY       ;LOOP UNTIL DRIVE READY
 FBD2 C9                RET


                COLDBOOT:
 FBD3 3140FC            LXI     SP,STACK
 FBD6 AF                XRA     A
 FBD7 0100FC            LXI     B,BUFF
 FBDA 02        CBUFF:  STAX    B               ;ZERO OUT RAM BUFFER
 FBDB 0C                INR     C
 FBDC C2DAFB            JNZ     CBUFF

 FBDF 3E0A              MVI     A,10
 FBE1 3210FC            STA     STEPTIME        ;SET STEPTIME LONGER THAN IT NEEDS TO BE
                                                ;TO BE SAFE, SINCE COLD BOOT LOADER RESETS IT
 FBE4 010000            LXI     B,0
 FBE7 CDA3FA            CALL    SETDMA          ;SETDMA DOES NOT CHANGE C REG, SO...
 FBEA CD1EFB            CALL    SELDSK          ;SELECT DRIVE A
 FBED 0E01              MVI     C,01            ;LOAD BOOTSTRAP LOADER
 FBEF CDA9FA            CALL    SETSEC          ;  FROM TRACK 0 SECTOR 1
 FBF2 CD29FB            CALL    READ
 FBF5 C2D3FB            JNZ     COLDBOOT        ;ON READ FAILURE, TRY AGAIN
 FBF8 C7                RST     0               ;EXECUTE BOOTSTRAP LOADER
                                                ;(SAVES 2 BYTES OVER JMP 0000)

 FBF9 BFDFEFF7  MASKTABLE       DB      0BFH,0DFH,0EFH,0F7H
FC11 ABOVE43    F970 ALOOP      FB9D ATHOME     F800 BASE       F854 BLOOP
FC00 BUFF       FBDA CBUFF      F967 CLOOP      FBD3 COLDBOOT
FC03 CONTROLBYTE                FC22 CURRDRIVE  FB7D DELAY      FB8A DELAY1
FB7E DELAY2     FC01 DENBYTE    FC16 DENMAP     FC0D DISK       FBBE DISKREADY1
FBC5 DISKREADY  FC0B DMA        FAEA DONESTEP   FADE DOSTEP     FC00 ERRORBYTE
F8DD ERROR      F8FB ERRORCOUNT F888 GLOOP      F82F GO         FBA6 GOHOME
FB71 HEADLOAD   0028 HEADSETTLE FAA0 HIGHE      FB97 HOME       FB59 INOK
F975 LLOOP      FC06 LOGINTAB   FA84 LOOP10     FA93 LOOP11     F9EE LOOP6
F9F6 LOOP7      FBF9 MASKTABLE  F973 MLOOP      F98D NLOOP      F8EA NO
FC05 PRESDISK   FE06 RDDATA     FE05 RDMARK     FE04 RDMRKCRC   FE00 RDSTAT
FE02 RDUART     F829 READ       F854 READDD     FC02 READWRITE  FA57 RETRY1
FC21 RETRYCOUNT FA1A RETRY      F8BF RLOOP      F8AE RXFER      F970 SD
FA2A SDLOOP1    FA31 SDLOOP2    FA20 SDTEST     FC0A SECTOR     FB1E SELDSK
FB25 SELDSK1    FB03 SETDEN     FAA3 SETDMA     FAA9 SETSEC     FB69 SETTN
FAAE SETTRK     FA69 SKEW       FA9D SKIP12     FAB8 SKIP3      FB0D SKIP
FA7E SKIPY      FC40 STACK      FACC STEPHEAD   FAD8 STEPIN     FABE STEPLOOP
FACF STEPOUT    000F STEPSETTLE FC10 STEPTIME   F95F SYNC       FE07 SYNCPORT
F8F2 TERROR1    F8E4 TERROR     F9DC TEST1      FC23 TESTMAX    FC0E TESTNEXT
F9D8 TEST       FC04 TRACK      FC12 TRACKTAB   FC20 TRY1       FC0F TWOSIDE
```

F91C WLOOP FE01 WRCLK FE00 WRCONT FE07 WRCRC FE06 WRDATA
F82D WRITE F903 WRITEDD F81E WRITEPROTECT F946 WRITESD
FE04 WRMRKCRC FE05 WRMRK FE02 WRUART F91B WXFER