Tarbell - 2ASM.COM

Disk file explanation

This is the explanation of the files which are on this disk.

1. 2xBIOS24.ASM
This is your bios file for your operating system. This contains all the disk I/O, and console I/O routines.

2. 2xBOOT24.ASM
This is the secondary bootstrap for your operating system.

3. 2ZBIOS24.ASM
This is a special modified file of the 2xBIOS24.ASM with Z-80 instructions. This file is smaller in size then the standard bios file used in your system. To assemble it, a special assembler called ZASM.COM is included on the disk. You must have a Z-80 cpu in order to use the 2ZBIOS24 in your system.

4. ASM.COM
This is the Digital Research assembler. It does not assemble Z-80 instructions.

5. BASIC.COM
This is a public domain basic written by Gorden Ubanks for his thesis. It is an unsupported peice of software.

6. COPY.ASM and COPY.COM
This is a disk copy routine. It will copy a disk from drive A to drive B.
Commands are:

COPY ALL<cr>	copys entire disk
COPY DATA<cr>	copys data tracks only
COPY SYSTEM<cr>	copys system tracks only

7. CPM.COM
This is your imiage file of your operating system. DO NOT DESTROY!!

8. DDT.COM
This is Digital Researches Dynamic Debugger Tool as described in their manuals

9. DEBLOCK.ASM
This is a special blocking/deblocking file which must be merged into your bios file to allow operating with sector sizes other than 128 byte. Instructions are in your manuals from Digital Research.

10.DISKDEF.LIB
This file is used to automatically create disk definitions in the bios file. You must have Digital Researches "MAC" macro assembler to use it.

11. DISKTEST.ASM
This is a disk test file. It will report hard and soft disk errors. this program is read only.

12. DUMP.ASM and DUMP.COM
This is a disk dump file from Digital Research.
To use it, type DUMP <filename.ext><cr>

13. ED.COM
This is Digital Researches Context Editor.

14. EXTRACT.COM
This program allows you to list out to the console (or list device if you type ^p), any portion of a PRN file between two (2) labels.
Command syntax:
EXTRACT <filename> (1st label) (2nd label)<cr>
This program will list starting in the middle of the file to the end of the file by using a dummy lable for the 2nd lable name (one which does not appear in the file).

15. FORMAT.ASM
This is a disk format program. It will only format in the A drive. Besure to remove your system disk from drive A and replace it with the disk you wish to format, as this program will destroy any data on the disk.

16.GUESS.COM
Fun game of guessing a number that the computer knows. This program

will run only on a Z-80 system. (oh well!).

17.INV.BAS
Unsupported inventory program. Will run with the basic on this disk.

18.LOAD.COM
Used to turn a HEX file into a COM file for execution by CPM.

19.PIP.COM
Program to transfer files from disk to disk, disk to console, etc. as explained in your CPM manuals.

20.PRINT.COM
Utility program for printing file name heading and page numbers on list device.
Command syntax:
PRINT <filename.ext><cr>

21.RUN.COM
Run time package for BASIC.COM

22.STAT.COM
Status program as explained in manuals.

23. STRIP.COM
This program will allow you to make an ASM file from a PRN file in case you should loose your original ASM file. It will also allow you to name the ASM file to something else if you wish.
Command syntax:
STRIP <filename> (<new filename> or <old filename>)<cr>
This program looks automatically for the PRN file and automatically creats the new file with the extenstion of ASM.

24.SUBMIT.COM
this is a batch processor as described in the manuals.

25.SYSGEN.COM
File used to put a new system on the first 2 tracks of a disk.

26.XDIR.COM
Wide disk directory view program. Only works with drive A or B.

27. XSUB.COM
This is the extended batch processor file as described in the manual.

28. ZASM.COM
This is a unsupported Z-80 assembler. This file will run on either 8080 or Z-80. The opcodes recognized by this assembler is included in your package, along with a cross reference from ZASM to ZILOG mnemonics. For the most part the command syntax is the same as the Digital researches ASM.COM file with one exception.
Command syntax:
ZASM <filename.ABC><cr>
where ABC must be provided or the assembler will not work right. The extension of .ABC are the options that may be used during assembly time.

```
                                ;THIS IS AN EXAMPLE
                                ;OF THE OPCODES FOR
                                ;THE ZASM ASSEMBLER.
                                ;
                                        TITLE   'ZASM OPCODES'  ;TITLE EXAMPLE
                                ;
0100                            LABLE:  ORG     100H            ;ORIGIN STATEMENT
0100                            STORE   EQU     100H            ;EXAMPLE OF AN EQU STATEMENT
0100  00 01                             DW      STORE           ;EXAMPLE OF A DW STATEMENT
0102  5A 41 53 4D 2E                    DB      'ZASM.COM'      ;EXAMPLE OF A DB STRING
0107  43 4F 4D
FFFF                            TRUE:   EQU     0FFFFH          ;DEFINE TRUE
0000                            FALSE:  EQU     NOT TRUE        ;DEFINE FALSE
                                ;
FFFF                            TEST    EQU     TRUE            ;CONDITIONAL STATEMENT
                                ;
                                        IF      TEST            ;ASSEMBLER EVALUATES THIS
010A  54 45 53 54 49                    DB      'TESTING'       ;STRING
010F  4E 47
                                        ELSE                    ;EXAMPLE OF ELSE
                                        DB      'RUNNING'       ;STRING
                                        ENDIF                   ;MUST END AN "IF" STATEMENT
0111  50 52 4F 47 52                    DB      'PROGRAM'       ;STRING EXAMPLE
0116  41 4D
                                ;
0118                                    DS      80              ;DECLARE STORAGE AREA
                                ;
0168  49                                DB      01001001B       ;BINARY EXAMPLE
0169  41                                DB      011100101Q      ;OCTAL EXAMPLE
016A  23                                DB      23H             ;HEX EXAMPLE
016B  32                                DB      50              ;DECIMAL EXAMPLE
                                ;
0035                            TYPE:   EQU     35H
0046                            NEW     EQU     46H
                                ;
007B                            SAME:   EQU     TYPE+NEW        ;ADDITION EXAMPLE
FFEF                            OLD:    EQU     TYPE-NEW        ;SUBTRACTION EXAMPLE
0E7E                            AGAIN:  EQU     TYPE*NEW        ;MULTIPLICATION EXAMPLE
0000                            TOP:    EQU     TYPE/NEW        ;DIVISION EXAMPLE
001A                            BOT:    EQU     TYPE SHR 1      ;SHIFT RIGHT EXAMPLE
00D4                            MID:    EQU     TYPE SHL 2      ;SHIFT LEFT EXAMPLE
FFCA                            LID:    EQU     NOT TYPE        ;'NOT' EXAMPLE
0004                            DISK:   EQU     TYPE AND NEW    ;'AND' EXAMPLE
0077                            BOOK:   EQU     TYPE OR NEW     ;'OR' EXAMPLE
0073                            LIST:   EQU     TYPE XOR NEW    ;'XOR' EXAMPLE
016C  33                        SIZE:   DB      TYPE MOD 10+'0' ;'MOD' EXAMPLE
                                ;
                                ;
016D  CE 00                             ACI     0
016F  8F                                ADC     A
0170  88                                ADC     B
0171  89                                ADC     C
0172  8A                                ADC     D
0173  8B                                ADC     E
```

```
 52.  0174  8C                    ADC     H
 53.  0175  8D                    ADC     L
 54.  0176  8E                    ADC     M
 55.  0177  DD 8E 46              ADC     [X+NEW]
 56.  017A  FD 8E 46              ADC     [Y+NEW]
 57.                         ;
 58.  017D  87                    ADD     A
 59.  017E  80                    ADD     B
 60.  017F  81                    ADD     C
 61.  0180  82                    ADD     D
 62.  0181  83                    ADD     E
 63.  0182  84                    ADD     H
 64.  0183  85                    ADD     L
 65.  0184  86                    ADD     M
 66.  0185  DD 86 46              ADD     [X+NEW]
 67.  0188  FD 86 46              ADD     [Y+NEW]
 68.                         ;
 69.  018B  C6 00                 ADI     0
 70.                         ;
 71.  018D  A7                    ANA     A
 72.  018E  A0                    ANA     B
 73.  018F  A1                    ANA     C
 74.  0190  A2                    ANA     D
 75.  0191  A3                    ANA     E
 76.  0192  A4                    ANA     H
 77.  0193  A5                    ANA     L
 78.  0194  A6                    ANA     M
 79.  0195  DD A6 46              ANA     [X+NEW]
 80.  0198  FD A6 46              ANA     [Y+NEW]
 81.                         ;
 82.  019B  E6 00                 ANI     0
 83.                         ;
 84.  019D  CB 47                 BIT     0,A
 85.  019F  CB 40                 BIT     0,B
 86.  01A1  CB 41                 BIT     0,C
 87.  01A3  CB 42                 BIT     0,D
 88.  01A5  CB 43                 BIT     0,E
 89.  01A7  CB 44                 BIT     0,H
 90.  01A9  CB 45                 BIT     0,L
 91.  01AB  CB 46                 BIT     0,M
 92.  01AD  DD CB 46 46           BIT     0,[X+NEW]
 93.  01B1  FD CB 46 46           BIT     0,[Y+NEW]
 94.                         ;
 95.  01B5  CB 4F                 BIT     1,A
 96.  01B7  CB 48                 BIT     1,B
 97.  01B9  CB 49                 BIT     1,C
 98.  01BB  CB 4A                 BIT     1,D
 99.  01BD  CB 4B                 BIT     1,E
100.  01BF  CB 4C                 BIT     1,H
101.  01C1  CB 4D                 BIT     1,L
102.  01C3  CB 4E                 BIT     1,M
103.  01C5  DD CB 46 4E           BIT     1,[X+NEW]
104.  01C9  FD CB 46 4E           BIT     1,[Y+NEW]
105.                        ;
```

```
106.  01CD  CB 57              BIT   2,A
107.  01CF  CB 50              BIT   2,B
108.  01D1  CB 51              BIT   2,C
109.  01D3  CB 52              BIT   2,D
110.  01D5  CB 53              BIT   2,E
111.  01D7  CB 54              BIT   2,H
112.  01D9  CB 55              BIT   2,L
113.  01DB  CB 56              BIT   2,M
114.  01DD  DD CB 46 56        BIT   2,[X+NEW]
115.  01E1  FD CB 46 56        BIT   2,[Y+NEW]
116.                           ;
117.  01E5  CB 5F              BIT   3,A
118.  01E7  CB 58              BIT   3,B
119.  01E9  CB 59              BIT   3,C
120.  01EB  CB 5A              BIT   3,D
121.  01ED  CB 5B              BIT   3,E
122.  01EF  CB 5C              BIT   3,H
123.  01F1  CB 5D              BIT   3,L
124.  01F3  CB 5E              BIT   3,M
125.  01F5  DD CB 46 5E        BIT   3,[X+NEW]
126.  01F9  FD CB 46 5E        BIT   3,[Y+NEW]
127.                           ;
128.  01FD  CB 67              BIT   4,A
129.  01FF  CB 60              BIT   4,B
130.  0201  CB 61              BIT   4,C
131.  0203  CB 62              BIT   4,D
132.  0205  CB 63              BIT   4,E
133.  0207  CB 64              BIT   4,H
134.  0209  CB 65              BIT   4,L
135.  020B  CB 66              BIT   4,M
136.  020D  DD CB 46 66        BIT   4,[X+NEW]
137.  0211  FD CB 46 66        BIT   4,[Y+NEW]
138.                           ;
139.  0215  CB 6F              BIT   5,A
140.  0217  CB 68              BIT   5,B
141.  0219  CB 69              BIT   5,C
142.  021B  CB 6A              BIT   5,D
143.  021D  CB 6B              BIT   5,E
144.  021F  CB 6C              BIT   5,H
145.  0221  CB 6D              BIT   5,L
146.  0223  CB 6E              BIT   5,M
147.  0225  DD CB 46 6E        BIT   5,[X+NEW]
148.  0229  FD CB 46 6E        BIT   5,[Y+NEW]
149.                           ;
150.  022D  CB 77              BIT   6,A
151.  022F  CB 70              BIT   6,B
152.  0231  CB 71              BIT   6,C
153.  0233  CB 72              BIT   6,D
154.  0235  CB 73              BIT   6,E
155.  0237  CB 74              BIT   6,H
156.  0239  CB 75              BIT   6,L
157.  023B  CB 76              BIT   6,M
158.  023D  DD CB 46 76        BIT   6,[X+NEW]
159.  0241  FD CB 46 76        BIT   6,[Y+NEW]
```

```
160.                                    ;
161.  0245  CB 7F                       BIT     7,A
162.  0247  CB 78                       BIT     7,B
163.  0249  CB 79                       BIT     7,C
164.  024B  CB 7A                       BIT     7,D
165.  024D  CB 7B                       BIT     7,E
166.  024F  CB 7C                       BIT     7,H
167.  0251  CB 7D                       BIT     7,L
168.  0253  CB 7E                       BIT     7,M
169.  0255  DD CB 46 7E                 BIT     7,[X+NEW]
170.  0259  FD CB 46 7E                 BIT     7,[Y+NEW]
171.                                    ;
172.  025D  CD 00 01                    CALL    STORE
173.  0260  DC 00 01                    CC      STORE
174.  0263  ED A9                       CCD
175.  0265  ED B9                       CCDR
176.  0267  ED A1                       CCI
177.  0269  ED B1                       CCIR
178.  026B  FC 00 01                    CM      STORE
179.  026E  2F                          CMA
180.  026F  3F                          CMC
181.                                    ;
182.  0270  BF                          CMP     A
183.  0271  B8                          CMP     B
184.  0272  B9                          CMP     C
185.  0273  BA                          CMP     D
186.  0274  BB                          CMP     E
187.  0275  BC                          CMP     H
188.  0276  BD                          CMP     L
189.  0277  BE                          CMP     M
190.  0278  DD BE 46                    CMP     [X+NEW]
191.  027B  FD BE 46                    CMP     [Y+NEW]
192.                                    ;
193.  027E  D4 00 01                    CNC     STORE
194.  0281  C4 00 01                    CNZ     STORE
195.  0284  F4 00 01                    CP      STORE
196.  0287  EC 00 01                    CPE     STORE
197.  028A  FE 00                       CPI     0
198.  028C  E4 00 01                    CPO     STORE
199.  028F  CC 00 01                    CZ      STORE
200.  0292  27                          DAA
201.  0293  09                          DAD     B
202.  0294  19                          DAD     D
203.  0295  29                          DAD     H
204.  0296  39                          DAD     SP
205.  0297  DD 09                       DADX    B
206.  0299  DD 19                       DADX    D
207.  029B  DD 29                       DADX    X
208.  029D  DD 39                       DADX    SP
209.  029F  ED 4A                       DADC    B
210.  02A1  ED 5A                       DADC    D
211.  02A3  ED 6A                       DADC    H
212.  02A5  ED 7A                       DADC    SP
213.  02A7  FD 09                       DADY    B
```

```
  214.  02A9  FD 19              DADY    D
  215.  02AB  FD 29              DADY    Y
  216.  02AD  FD 39              DADY    SP
  217.                          ;
  218.  02AF  3D                DCR     A
  219.  02B0  05                DCR     B
  220.  02B1  0D                DCR     C
  221.  02B2  15                DCR     D
  222.  02B3  1D                DCR     E
  223.  02B4  25                DCR     H
  224.  02B5  2D                DCR     L
  225.  02B6  35                DCR     M
  226.  02B7  DD 35 46          DCR     [X+NEW]
  227.  02BA  FD 35 46          DCR     [Y+NEW]
  228.                          ;
  229.  02BD  0B                DCX     B
  230.  02BE  1B                DCX     D
  231.  02BF  2B                DCX     H
  232.  02C0  3B                DCX     SP
  233.  02C1  DD 2B             DCX     X
  234.  02C3  FD 2B             DCX     Y
  235.                          ;
  236.  02C5  F3                DI
  237.  02C6  10 FE             DJNZ    $
  238.  02C8  ED 42             DSBB    B
  239.  02CA  ED 52             DSBB    D
  240.  02CC  ED 62             DSBB    H
  241.  02CE  ED 72             DSBB    SP
  242.  02D0  FB                EI
  243.  02D1  08                EXAF
  244.  02D2  D9                EXX
  245.  02D3  76                HLT
  246.  02D4  ED 46             IM0
  247.  02D6  ED 56             IM1
  248.  02D8  ED 5E             IM2
  249.                          ;
  250.  02DA  DB 00             IN      0
  251.  02DC  ED AA             IND
  252.  02DE  ED BA             INDR
  253.  02E0  ED A2             INI
  254.  02E2  ED B2             INIR
  255.                          ;
  256.  02E4  ED 78             INP     A
  257.  02E6  ED 40             INP     B
  258.  02E8  ED 48             INP     C
  259.  02EA  ED 50             INP     D
  260.  02EC  ED 58             INP     E
  261.  02EE  ED 60             INP     H
  262.  02F0  ED 68             INP     L
  263.                          ;
  264.  02F2  3C                INR     A
  265.  02F3  04                INR     B
  266.  02F4  0C                INR     C
  267.  02F5  14                INR     D
```

```
 268.  02F6  1C                    INR     E
 269.  02F7  24                    INR     H
 270.  02F8  2C                    INR     L
 271.  02F9  34                    INR     M
 272.  02FA  DD 34 46              INR     [X+NEW]
 273.  02FD  FD 34 46              INR     [Y+NEW]
 274.                          ;
 275.  0300  03                    INX     B
 276.  0301  13                    INX     D
 277.  0302  23                    INX     H
 278.  0303  33                    INX     SP
 279.  0304  DD 23                 INX     X
 280.  0306  FD 23                 INX     Y
 281.                          ;
 282.  0308  DA 00 01              JC      STORE
 283.  030B  FA 00 01              JM      STORE
 284.  030E  C3 00 01              JMP     STORE
 285.  0311  18 FE                 JMPR    $
 286.  0313  D2 00 01              JNC     STORE
 287.  0316  C2 00 01              JNZ     STORE
 288.  0319  F2 00 01              JP      STORE
 289.  031C  EA 00 01              JPE     STORE
 290.  031F  E2 00 01              JPO     STORE
 291.  0322  38 FE                 JRC     $
 292.  0324  28 FE                 JRZ     $
 293.  0326  30 FE                 JRNC    $
 294.  0328  20 FE                 JRNZ    $
 295.  032A  CA 00 01              JZ      STORE
 296.                          ;
 297.  032D  ED 4B 00 01           LBCD    STORE
 298.  0331  0A                    LDAX    B
 299.  0332  1A                    LDAX    D
 300.  0333  3A 00 01              LDA     STORE
 301.  0336  ED 57                 LDAI
 302.  0338  ED A8                 LDD
 303.  033A  ED B8                 LDDR
 304.  033C  ED 5B 00 01           LDED    STORE
 305.  0340  ED A0                 LDI
 306.  0342  ED B0                 LDIR
 307.  0344  2A 00 01              LHLD    STORE
 308.  0347  DD 2A 00 01           LIXD    STORE
 309.  034B  FD 2A 00 01           LIYD    STORE
 310.  034F  ED 7B 00 01           LSPD    STORE
 311.                          ;
 312.  0353  01 00 01              LXI     B,STORE
 313.  0356  11 00 01              LXI     D,STORE
 314.  0359  21 00 01              LXI     H,STORE
 315.  035C  31 00 01              LXI     SP,STORE
 316.  035F  DD 21 00 01           LXI     X,STORE
 317.  0363  FD 21 00 01           LXI     Y,STORE
 318.                          ;
 319.  0367  7F                    MOV     A,A
 320.  0368  78                    MOV     A,B
 321.  0369  79                    MOV     A,C
```

```
322.  036A  7A              MOV   A,D
323.  036B  7B              MOV   A,E
324.  036C  7C              MOV   A,H
325.  036D  7D              MOV   A,L
326.  036E  7E              MOV   A,M
327.  036F  DD 7E 46         MOV   A,[X+NEW]
328.  0372  FD 7E 46         MOV   A,[Y+NEW]
329.                      ;
330.  0375  47              MOV   B,A
331.  0376  40              MOV   B,B
332.  0377  41              MOV   B,C
333.  0378  42              MOV   B,D
334.  0379  43              MOV   B,E
335.  037A  44              MOV   B,H
336.  037B  45              MOV   B,L
337.  037C  46              MOV   B,M
338.  037D  DD 46 46         MOV   B,[X+NEW]
339.  0380  FD 46 46         MOV   B,[Y+NEW]
340.                      ;
341.  0383  4F              MOV   C,A
342.  0384  48              MOV   C,B
343.  0385  49              MOV   C,C
344.  0386  4A              MOV   C,D
345.  0387  4B              MOV   C,E
346.  0388  4C              MOV   C,H
347.  0389  4D              MOV   C,L
348.  038A  4E              MOV   C,M
349.  038B  DD 4E 46         MOV   C,[X+NEW]
350.  038E  FD 4E 46         MOV   C,[Y+NEW]
351.                      ;
352.  0391  57              MOV   D,A
353.  0392  50              MOV   D,B
354.  0393  51              MOV   D,C
355.  0394  52              MOV   D,D
356.  0395  53              MOV   D,E
357.  0396  54              MOV   D,H
358.  0397  55              MOV   D,L
359.  0398  56              MOV   D,M
360.  0399  DD 56 46         MOV   D,[X+NEW]
361.  039C  FD 56 46         MOV   D,[Y+NEW]
362.                      ;
363.  039F  5F              MOV   E,A
364.  03A0  58              MOV   E,B
365.  03A1  59              MOV   E,C
366.  03A2  5A              MOV   E,D
367.  03A3  5B              MOV   E,E
368.  03A4  5C              MOV   E,H
369.  03A5  5D              MOV   E,L
370.  03A6  5E              MOV   E,M
371.  03A7  DD 5E 46         MOV   E,[X+NEW]
372.  03AA  FD 5E 46         MOV   E,[Y+NEW]
373.                      ;
374.  03AD  67              MOV   H,A
375.  03AE  60              MOV   H,B
```

```
376.  03AF  61              MOV   H,C
377.  03B0  62              MOV   H,D
378.  03B1  63              MOV   H,E
379.  03B2  64              MOV   H,H
380.  03B3  65              MOV   H,L
381.  03B4  66              MOV   H,M
382.  03B5  DD 66 46        MOV   H,[X+NEW]
383.  03B8  FD 66 46        MOV   H,[Y+NEW]
384.                        ;
385.  03BB  6F              MOV   L,A
386.  03BC  68              MOV   L,B
387.  03BD  69              MOV   L,C
388.  03BE  6A              MOV   L,D
389.  03BF  6B              MOV   L,E
390.  03C0  6C              MOV   L,H
391.  03C1  6D              MOV   L,L
392.  03C2  6E              MOV   L,M
393.  03C3  DD 6E 46        MOV   L,[X+NEW]
394.  03C6  FD 6E 46        MOV   L,[Y+NEW]
395.                        ;
396.  03C9  77              MOV   M,A
397.  03CA  70              MOV   M,B
398.  03CB  71              MOV   M,C
399.  03CC  72              MOV   M,D
400.  03CD  73              MOV   M,E
401.  03CE  74              MOV   M,H
402.  03CF  75              MOV   M,L
403.                        ;
404.  03D0  DD 77 46        MOV   [X+NEW],A
405.  03D3  FD 77 46        MOV   [Y+NEW],A
406.  03D6  DD 70 46        MOV   [X+NEW],B
407.  03D9  FD 70 46        MOV   [Y+NEW],B
408.  03DC  DD 71 46        MOV   [X+NEW],C
409.  03DF  FD 71 46        MOV   [Y+NEW],C
410.  03E2  DD 72 46        MOV   [X+NEW],D
411.  03E5  FD 72 46        MOV   [Y+NEW],D
412.  03E8  DD 73 46        MOV   [X+NEW],E
413.  03EB  FD 73 46        MOV   [Y+NEW],E
414.  03EE  DD 74 46        MOV   [X+NEW],H
415.  03F1  FD 74 46        MOV   [Y+NEW],H
416.  03F4  DD 75 46        MOV   [X+NEW],L
417.  03F7  FD 75 46        MOV   [Y+NEW],L
418.                        ;
419.  03FA  3E 00           MVI   A,0
420.  03FC  06 00           MVI   B,0
421.  03FE  0E 00           MVI   C,0
422.  0400  16 00           MVI   D,0
423.  0402  1E 00           MVI   E,0
424.  0404  26 00           MVI   H,0
425.  0406  2E 00           MVI   L,0
426.  0408  36 00           MVI   M,0
427.  040A  DD 36 46 00     MVI   [X+NEW],0
428.  040E  FD 36 46 00     MVI   [Y+NEW],0
429.                        ;
```

```
430.  0412  ED 44              NEG
431.  0414  00                 NOP
432.  0415  B7                 ORA     A
433.  0416  B0                 ORA     B
434.  0417  B1                 ORA     C
435.  0418  B2                 ORA     D
436.  0419  B3                 ORA     E
437.  041A  B4                 ORA     H
438.  041B  B5                 ORA     L
439.  041C  B6                 ORA     M
440.  041D  DD B6 46           ORA     [X+NEW]
441.  0420  FD B6 46           ORA     [Y+NEW]
442.                         ;
443.  0423  F6 00              ORI     0
444.  0425  ED BB              OTDR
445.  0427  ED B3              OTIR
446.                         ;
447.  0429  D3 46              OUT     NEW
448.  042B  ED 79              OUTP    A
449.  042D  ED 41              OUTP    B
450.  042F  ED 49              OUTP    C
451.  0431  ED 51              OUTP    D
452.  0433  ED 59              OUTP    E
453.  0435  ED 61              OUTP    H
454.  0437  ED 69              OUTP    L
455.                         ;
456.  0439  ED A3              OUTI
457.  043B  ED AB              OUTD
458.  043D  E9                 PCHL
459.  043E  DD E9              PCIX
460.  0440  FD E9              PCIY
461.                         ;
462.  0442  C1                 POP     B
463.  0443  D1                 POP     D
464.  0444  E1                 POP     H
465.  0445  F1                 POP     PSW
466.  0446  DD E1              POP     X
467.  0448  FD E1              POP     Y
468.  044A  C5                 PUSH    B
469.  044B  D5                 PUSH    D
470.  044C  E5                 PUSH    H
471.  044D  F5                 PUSH    PSW
472.  044E  DD E5              PUSH    X
473.  0450  FD E5              PUSH    Y
474.                         ;
475.  0452  17                 RAL
476.  0453  CB 17              RALR    A
477.  0455  CB 10              RALR    B
478.  0457  CB 11              RALR    C
479.  0459  CB 12              RALR    D
480.  045B  CB 13              RALR    E
481.  045D  CB 14              RALR    H
482.  045F  CB 15              RALR    L
483.  0461  CB 16              RALR    M
```

```
484.  0463  DD CB 46 16          RALR    [X+NEW]
485.  0467  FD CB 46 16          RALR    [Y+NEW]
486.                       ;
487.  046B  1F                   RAR
488.  046C  CB 1F                RARR    A
489.  046E  CB 18                RARR    B
490.  0470  CB 19                RARR    C
491.  0472  CB 1A                RARR    D
492.  0474  CB 1B                RARR    E
493.  0476  CB 1C                RARR    H
494.  0478  CB 1D                RARR    L
495.  047A  CB 1E                RARR    M
496.  047C  DD CB 46 1E          RARR    [X+NEW]
497.  0480  FD CB 46 1E          RARR    [Y+NEW]
498.                       ;
499.  0484  D8                   RC
500.  0485  CB 87                RES     0,A
501.  0487  CB 80                RES     0,B
502.  0489  CB 81                RES     0,C
503.  048B  CB 82                RES     0,D
504.  048D  CB 83                RES     0,E
505.  048F  CB 84                RES     0,H
506.  0491  CB 85                RES     0,L
507.  0493  CB 86                RES     0,M
508.                       ;
509.  0495  CB 8F                RES     1,A
510.  0497  CB 88                RES     1,B
511.  0499  CB 89                RES     1,C
512.  049B  CB 8A                RES     1,D
513.  049D  CB 8B                RES     1,E
514.  049F  CB 8C                RES     1,H
515.  04A1  CB 8D                RES     1,L
516.  04A3  CB 8E                RES     1,M
517.                       ;
518.  04A5  CB 97                RES     2,A
519.  04A7  CB 90                RES     2,B
520.  04A9  CB 91                RES     2,C
521.  04AB  CB 92                RES     2,D
522.  04AD  CB 93                RES     2,E
523.  04AF  CB 94                RES     2,H
524.  04B1  CB 95                RES     2,L
525.  04B3  CB 96                RES     2,M
526.                       ;
527.  04B5  CB 9F                RES     3,A
528.  04B7  CB 98                RES     3,B
529.  04B9  CB 99                RES     3,C
530.  04BB  CB 9A                RES     3,D
531.  04BD  CB 9B                RES     3,E
532.  04BF  CB 9C                RES     3,H
533.  04C1  CB 9D                RES     3,L
534.  04C3  CB 9E                RES     3,M
535.                       ;
536.  04C5  CB A7                RES     4,A
537.  04C7  CB A0                RES     4,B
```

```
538.  04C9  CB A1          RES   4,C
539.  04CB  CB A2          RES   4,D
540.  04CD  CB A3          RES   4,E
541.  04CF  CB A4          RES   4,H
542.  04D1  CB A5          RES   4,L
543.  04D3  CB A6          RES   4,M
544.                       ;
545.  04D5  CB AF          RES   5,A
546.  04D7  CB A8          RES   5,B
547.  04D9  CB A9          RES   5,C
548.  04DB  CB AA          RES   5,D
549.  04DD  CB AB          RES   5,E
550.  04DF  CB AC          RES   5,H
551.  04E1  CB AD          RES   5,L
552.  04E3  CB AE          RES   5,M
553.                       ;
554.  04E5  CB B7          RES   6,A
555.  04E7  CB B0          RES   6,B
556.  04E9  CB B1          RES   6,C
557.  04EB  CB B2          RES   6,D
558.  04ED  CB B3          RES   6,E
559.  04EF  CB B4          RES   6,H
560.  04F1  CB B5          RES   6,L
561.  04F3  CB B6          RES   6,M
562.                       ;
563.  04F5  CB BF          RES   7,A
564.  04F7  CB B8          RES   7,B
565.  04F9  CB B9          RES   7,C
566.  04FB  CB BA          RES   7,D
567.  04FD  CB BB          RES   7,E
568.  04FF  CB BC          RES   7,H
569.  0501  CB BD          RES   7,L
570.  0503  CB BE          RES   7,M
571.                       ;
572.  0505  DD CB 46 86     RES   0,[X+NEW]
573.  0509  FD CB 46 86     RES   0,[Y+NEW]
574.  050D  DD CB 46 8E     RES   1,[X+NEW]
575.  0511  FD CB 46 8E     RES   1,[Y+NEW]
576.  0515  DD CB 46 96     RES   2,[X+NEW]
577.  0519  FD CB 46 96     RES   2,[Y+NEW]
578.  051D  DD CB 46 9E     RES   3,[X+NEW]
579.  0521  FD CB 46 9E     RES   3,[Y+NEW]
580.  0525  DD CB 46 A6     RES   4,[X+NEW]
581.  0529  FD CB 46 A6     RES   4,[Y+NEW]
582.  052D  DD CB 46 AE     RES   5,[X+NEW]
583.  0531  FD CB 46 AE     RES   5,[Y+NEW]
584.  0535  DD CB 46 B6     RES   6,[X+NEW]
585.  0539  FD CB 46 B6     RES   6,[Y+NEW]
586.  053D  DD CB 46 BE     RES   7,[X+NEW]
587.  0541  FD CB 46 BE     RES   7,[Y+NEW]
588.                       ;
589.  0545  C9              RET
590.  0546  ED 45           RETN
591.  0548  ED 4D           RETI
```

```
592.  054A  07                      RLC
593.  054B  CB 07                   RLCR    A
594.  054D  CB 00                   RLCR    B
595.  054F  CB 01                   RLCR    C
596.  0551  CB 02                   RLCR    D
597.  0553  CB 03                   RLCR    E
598.  0555  CB 04                   RLCR    H
599.  0557  CB 05                   RLCR    L
600.  0559  CB 06                   RLCR    M
601.  055B  DD CB 46 06             RLCR    [X+NEW]
602.  055F  FD CB 46 06             RLCR    [Y+NEW]
603.                              ;
604.  0563  ED 6F                   RLD
605.  0565  F8                      RM
606.  0566  D0                      RNC
607.  0567  C0                      RNZ
608.  0568  F0                      RP
609.  0569  E8                      RPE
610.  056A  E0                      RPO
611.                              ;
612.  056B  0F                      RRC
613.  056C  CB 0F                   RRCR    A
614.  056E  CB 08                   RRCR    B
615.  0570  CB 09                   RRCR    C
616.  0572  CB 0A                   RRCR    D
617.  0574  CB 0B                   RRCR    E
618.  0576  CB 0C                   RRCR    H
619.  0578  CB 0D                   RRCR    L
620.  057A  CB 0E                   RRCR    M
621.  057C  DD CB 46 0E             RRCR    [X+NEW]
622.  0580  FD CB 46 0E             RRCR    [Y+NEW]
623.                              ;
624.  0584  ED 67                   RRD
625.  0586  C7                      RST     0
626.  0587  CF                      RST     1
627.  0588  D7                      RST     2
628.  0589  DF                      RST     3
629.  058A  E7                      RST     4
630.  058B  EF                      RST     5
631.  058C  F7                      RST     6
632.  058D  FF                      RST     7
633.                              ;
634.  058E  C8                      RZ
635.  058F  9F                      SBB     A
636.  0590  98                      SBB     B
637.  0591  99                      SBB     C
638.  0592  9A                      SBB     D
639.  0593  9B                      SBB     E
640.  0594  9C                      SBB     H
641.  0595  9D                      SBB     L
642.  0596  9E                      SBB     M
643.  0597  DD 9E 46                SBB     [X+NEW]
644.  059A  FD 9E 46                SBB     [Y+NEW]
645.                              ;
```

```
646.  059D  ED 43 00 01          SBCD   STORE
647.  05A1  DE 00                SBI    0
648.  05A3  ED 53 00 01          SDED   STORE
649.                             ;
650.  05A7  CB C7                SET    0,A
651.  05A9  CB C0                SET    0,B
652.  05AB  CB C1                SET    0,C
653.  05AD  CB C2                SET    0,D
654.  05AF  CB C3                SET    0,E
655.  05B1  CB C4                SET    0,H
656.  05B3  CB C5                SET    0,L
657.  05B5  CB C6                SET    0,M
658.                             ;
659.  05B7  CB CF                SET    1,A
660.  05B9  CB C8                SET    1,B
661.  05BB  CB C9                SET    1,C
662.  05BD  CB CA                SET    1,D
663.  05BF  CB CB                SET    1,E
664.  05C1  CB CC                SET    1,H
665.  05C3  CB CD                SET    1,L
666.  05C5  CB CE                SET    1,M
667.                             ;
668.  05C7  CB D7                SET    2,A
669.  05C9  CB D0                SET    2,B
670.  05CB  CB D1                SET    2,C
671.  05CD  CB D2                SET    2,D
672.  05CF  CB D3                SET    2,E
673.  05D1  CB D4                SET    2,H
674.  05D3  CB D5                SET    2,L
675.  05D5  CB D6                SET    2,M
676.                             ;
677.  05D7  CB DF                SET    3,A
678.  05D9  CB D8                SET    3,B
679.  05DB  CB D9                SET    3,C
680.  05DD  CB DA                SET    3,D
681.  05DF  CB DB                SET    3,E
682.  05E1  CB DC                SET    3,H
683.  05E3  CB DD                SET    3,L
684.  05E5  CB DE                SET    3,M
685.                             ;
686.  05E7  CB E7                SET    4,A
687.  05E9  CB E0                SET    4,B
688.  05EB  CB E1                SET    4,C
689.  05ED  CB E2                SET    4,D
690.  05EF  CB E3                SET    4,E
691.  05F1  CB E4                SET    4,H
692.  05F3  CB E5                SET    4,L
693.  05F5  CB E6                SET    4,M
694.                             ;
695.  05F7  CB EF                SET    5,A
696.  05F9  CB E8                SET    5,B
697.  05FB  CB E9                SET    5,C
698.  05FD  CB EA                SET    5,D
699.  05FF  CB EB                SET    5,E
```

```
700.  0601  CB EC                SET   5,H
701.  0603  CB ED                SET   5,L
702.  0605  CB EE                SET   5,M
703.                             ;
704.  0607  CB F7                SET   6,A
705.  0609  CB F0                SET   6,B
706.  060B  CB F1                SET   6,C
707.  060D  CB F2                SET   6,D
708.  060F  CB F3                SET   6,E
709.  0611  CB F4                SET   6,H
710.  0613  CB F5                SET   6,L
711.  0615  CB F6                SET   6,M
712.                             ;
713.  0617  CB FF                SET   7,A
714.  0619  CB F8                SET   7,B
715.  061B  CB F9                SET   7,C
716.  061D  CB FA                SET   7,D
717.  061F  CB FB                SET   7,E
718.  0621  CB FC                SET   7,H
719.  0623  CB FD                SET   7,L
720.  0625  CB FE                SET   7,M
721.                             ;
722.  0627  DD CB 46 C6          SET   0,[X+NEW]
723.  062B  FD CB 46 C6          SET   0,[Y+NEW]
724.  062F  DD CB 46 CE          SET   1,[X+NEW]
725.  0633  FD CB 46 CE          SET   1,[Y+NEW]
726.  0637  DD CB 46 D6          SET   2,[X+NEW]
727.  063B  FD CB 46 D6          SET   2,[Y+NEW]
728.  063F  DD CB 46 DE          SET   3,[X+NEW]
729.  0643  FD CB 46 DE          SET   3,[Y+NEW]
730.  0647  DD CB 46 E6          SET   4,[X+NEW]
731.  064B  FD CB 46 E6          SET   4,[Y+NEW]
732.  064F  DD CB 46 EE          SET   5,[X+NEW]
733.  0653  FD CB 46 EE          SET   5,[Y+NEW]
734.  0657  DD CB 46 F6          SET   6,[X+NEW]
735.  065B  FD CB 46 F6          SET   6,[Y+NEW]
736.  065F  DD CB 46 FE          SET   7,[X+NEW]
737.  0663  FD CB 46 FE          SET   7,[Y+NEW]
738.                             ;
739.  0667  22 00 01             SHLD  STORE
740.  066A  DD 22 00 01          SIXD  STORE
741.  066E  FD 22 00 01          SIYD  STORE
742.                             ;
743.  0672  CB 27                SLAR  A
744.  0674  CB 20                SLAR  B
745.  0676  CB 21                SLAR  C
746.  0678  CB 22                SLAR  D
747.  067A  CB 23                SLAR  E
748.  067C  CB 24                SLAR  H
749.  067E  CB 25                SLAR  L
750.  0680  CB 26                SLAR  M
751.  0682  DD CB 46 26          SLAR  [X+NEW]
752.  0686  FD CB 46 26          SLAR  [Y+NEW]
753.                             ;
```

```
754.  068A  F9                    SPHL
755.  068B  DD F9                 SPIX
756.  068D  FD F9                 SPIY
757.                              ;
758.  068F  CB 2F                 SRAR    A
759.  0691  CB 28                 SRAR    B
760.  0693  CB 29                 SRAR    C
761.  0695  CB 2A                 SRAR    D
762.  0697  CB 2B                 SRAR    E
763.  0699  CB 2C                 SRAR    H
764.  069B  CB 2D                 SRAR    L
765.  069D  CB 2E                 SRAR    M
766.  069F  DD CB 46 2E           SRAR    [X+NEW]
767.  06A3  FD CB 46 2E           SRAR    [Y+NEW]
768.                              ;
769.  06A7  CB 3F                 SRLR    A
770.  06A9  CB 38                 SRLR    B
771.  06AB  CB 39                 SRLR    C
772.  06AD  CB 3A                 SRLR    D
773.  06AF  CB 3B                 SRLR    E
774.  06B1  CB 3C                 SRLR    H
775.  06B3  CB 3D                 SRLR    L
776.  06B5  CB 3E                 SRLR    M
777.  06B7  DD CB 46 3E           SRLR    [X+NEW]
778.  06BB  FD CB 46 3E           SRLR    [Y+NEW]
779.                              ;
780.  06BF  ED 73 00 01           SSPD    STORE
781.  06C3  02                    STAX    B
782.  06C4  12                    STAX    D
783.  06C5  32 00 01              STA     STORE
784.  06C8  ED 47                 STAI
785.  06CA  37                    STC
786.                              ;
787.  06CB  97                    SUB     A
788.  06CC  90                    SUB     B
789.  06CD  91                    SUB     C
790.  06CE  92                    SUB     D
791.  06CF  93                    SUB     E
792.  06D0  94                    SUB     H
793.  06D1  95                    SUB     L
794.  06D2  96                    SUB     M
795.  06D3  DD 96 46              SUB     [X+NEW]
796.  06D6  FD 96 46              SUB     [Y+NEW]
797.                              ;
798.  06D9  D6 00                 SUI     0
799.  06DB  EB                    XCHG
800.                              ;
801.  06DC  AF                    XRA     A
802.  06DD  A8                    XRA     B
803.  06DE  A9                    XRA     C
804.  06DF  AA                    XRA     D
805.  06E0  AB                    XRA     E
806.  06E1  AC                    XRA     H
807.  06E2  AD                    XRA     L
```

```
808.  06E3  AE              XRA     M
809.  06E4  DD AE 46        XRA     [X+NEW]
810.  06E7  FD AE 46        XRA     [Y+NEW]
811.                        ;
812.  06EA  EE 00           XRI     0
813.  06EC  E3              XTHL
814.  06ED  DD E3           XTIX
815.  06EF  FD E3           XTIY
      0100
```

ACI	0	ADC	A,N
ADC	B	ADC	A,A
ADC	B	ADC	A,B
ADC	C	ADC	A,C
ADC	D	ADC	A,D
ADC	E	ADC	A,E
ADC	H	ADC	A,H
ADC	L	ADC	A,L
ADC	M	ADC	A,(HL)
ADC	[X+0]	ADC	A,(IX+D)
ADC	[Y+0]	ADC	A,(IY+D)
ADD	A	ADD	A,A
ADD	B	ADD	A,B
ADD	C	ADD	A,C
ADD	D	ADD	A,D
ADD	E	ADD	A,E
ADD	H	ADD	A,H
ADD	L	ADD	A,L
ADD	M	ADD	A,(HL)
ADD	[X+0]	ADD	A,(IX+D)
ADD	[Y+0]	ADD	A,(IY+D)
ADI	0	ADD	A,N
ANA	A	AND	A
ANA	B	AND	B
ANA	C	AND	C
ANA	D	AND	D
ANA	E	AND	E
ANA	H	AND	H
ANA	L	AND	L
ANA	M	AND	(HL)
ANA	[X+0]	AND	(IX+D)
ANA	[Y+0]	AND	(IY+D)
ANI	0	AND	N
BIT	0,A	BIT	0,A
BIT	0,B	BIT	0,B
BIT	0,C	BIT	0,C
BIT	0,D	BIT	0,D
BIT	0,E	BIT	0,E
BIT	0,H	BIT	0,H
BIT	0,L	BIT	0,L
BIT	0,M	BIT	0,(HL)
BIT	1,A	BIT	1,A
BIT	1,B	BIT	1,B
BIT	1,C	BIT	1,C
BIT	1,D	BIT	1,D
BIT	1,E	BIT	1,E
BIT	1,H	BIT	1,H
BIT	1,L	BIT	1,L
BIT	1,M	BIT	1,(HL)
BIT	2,A	BIT	2,A
BIT	2,B	BIT	2,B
BIT	2,C	BIT	2,C
BIT	2,D	BIT	2,D
BIT	2,E	BIT	2,E
BIT	2,H	BIT	2,H
BIT	2,L	BIT	2,L
BIT	2,M	BIT	2,(HL)
BIT	3,A	BIT	3,A

```
BIT     3,B
BIT     3,C
BIT     3,D
BIT     3,E
BIT     3,H
BIT     3,L
BIT     3,M
BIT     4,A
BIT     4,B
BIT     4,C
BIT     4,D
BIT     4,E
BIT     4,H
BIT     4,L
BIT     4,M
BIT     5,A
BIT     5,B
BIT     5,C
BIT     5,D
BIT     5,E
BIT     5,H
BIT     5,L
BIT     5,M
BIT     6,A
BIT     6,B
BIT     6,C
BIT     6,D
BIT     6,E
BIT     6,H
BIT     6,L
BIT     6,M
BIT     7,A
BIT     7,B
BIT     7,C
BIT     7,D
BIT     7,E
BIT     7,H
BIT     7,L
BIT     7,M
BIT     0,[X+0]
BIT     1,[X+0]
BIT     2,[X+0]
BIT     3,[X+0]
BIT     4,[X+0]
BIT     5,[X+0]
BIT     6,[X+0]
BIT     7,[X+0]
BIT     0,[Y+0]
BIT     1,[Y+0]
BIT     2,[Y+0]
BIT     3,[Y+0]
BIT     4,[Y+0]
BIT     5,[Y+0]
BIT     6,[Y+0]
BIT     7,[Y+0]
CALL    0
CC      0
CCD
```

```
BIT     3,B
BIT     3,C
BIT     3,D
BIT     3,E
BIT     3,H
BIT     3,L
BIT     3,(HL)
BIT     4,A
BIT     4,B
BIT     4,C
BIT     4,D
BIT     4,E
BIT     4,H
BIT     4,L
BIT     4,(HL)
BIT     5,A
BIT     5,B
BIT     5,C
BIT     5,D
BIT     5,E
BIT     5,H
BIT     5,L
BIT     5,(HL)
BIT     6,A
BIT     6,B
BIT     6,C
BIT     6,D
BIT     6,E
BIT     6,H
BIT     6,L
BIT     6,(HL)
BIT     7,A
BIT     7,B
BIT     7,C
BIT     7,D
BIT     7,E
BIT     7,H
BIT     7,L
BIT     7,(HL)
BIT     0,(IX+D)
BIT     1,(IX+D)
BIT     2,(IX+D)
BIT     3,(IX+D)
BIT     4,(IX+D)
BIT     5,(IX+D)
BIT     6,(IX+D)
BIT     7,(IX+D)
BIT     0,(IY+D)
BIT     1,(IY+D)
BIT     2,(IY+D)
BIT     3,(IY+D)
BIT     4,(IY+D)
BIT     5,(IY+D)
BIT     6,(IY+D)
BIT     7,(IY+D)
CALL    NN
CALL    C,NN
CPD
```

CCDR		CPDR	
CCI		CPI	
CCIR		CPIR	
CM	0	CALL	M,NN
CMA		CPL	
CMC		CCF	
CMP	A	CP	A
CMP	B	CP	B
CMP	C	CP	C
CMP	D	CP	D
CMP	E	CP	E
CMP	H	CP	H
CMP	L	CP	L
CMP	M	CP	(HL)
CMP	[X+0]	CP	(IX+D)
CMP	[Y+0]	CP	(IY+D)
CNC	0	CALL	NC,NN
CNZ	0	CALL	NZ,NN
CP	0	CALL	P,NN
CPE	0	CALL	PE,NN
CPI	0	CP	N
CPO	0	CALL	PO,NN
CZ	0	CALL	Z,NN
DAA		DAA	
DAD	B	ADD	HL,BC
DAD	D	ADD	HL,DE
DAD	H	ADD	HL,HL
DAD	SP	ADD	HL,SP
DADX	B	ADD	IX,BC
DADX	D	ADD	IX,DE
DADX	X	ADD	IX,IX
DADX	SP	ADD	IX,SP
DADC	B	ADC	HL,BC
DADC	D	ADC	HL,DE
DADC	H	ADC	HL,HL
DADC	SP	ADC	HL,SP
DADY	B	ADD	IY,BC
DADY	D	ADD	IY,DE
DADY	Y	ADD	IY,IY
DADY	SP	ADD	IY,SP
DCR	A	DEC	A
DCR	B	DEC	B
DCR	C	DEC	C
DCR	D	DEC	D
DCR	E	DEC	E
DCR	H	DEC	H
DCR	L	DEC	L
DCR	M	DEC	(HL)
DCR	[X+0]	DEC	(IX+D)
DCR	[Y+0]	DEC	(IY+D)
DCX	B	DEC	BC
DCX	D	DEC	DE
DCX	H	DEC	HL
DCX	SP	DEC	SP
DCX	X	DEC	IX
DCX	Y	DEC	IY
DI		DI	
DJNZ	$	DJNZ	DIS

DSBB	B	SBC	HL,BC
DSBB	D	SBC	HL,DE
DSBB	H	SBC	HL,HL
DSBB	SP	SBC	HL,SP
EI		EI	
EXAF		EX	AF,AF'
EXX		EXX	
HLT		HALT	
IM0		IM0	
IM1		IM1	
IM2		IM2	
IN	0	IN	A,(N)
IND		IND	
INDR		INDR	
INI		INI	
INIR		INIR	
INP	A	IN	A,(C)
INP	B	IN	B,(C)
INP	C	IN	C,(C)
INP	D	IN	D,(C)
INP	E	IN	E,(C)
INP	H	IN	H,(C)
INP	L	IN	L,(C)
INR	A	INC	A
INR	B	INC	B
INR	C	INC	C
INR	D	INC	D
INR	E	INC	E
INR	H	INC	H
INR	L	INC	L
INR	M	INC	(HL)
INR	[X+0]	INC	(IX+D)
INR	[Y+0]	INC	(IY+D)
INX	B	INC	BC
INX	D	INC	DE
INX	H	INC	HL
INX	SP	INC	SP
INX	X	INC	IX
INX	Y	INC	IY
JC	0	JP	C,NN
JM	0	JP	M,NN
JMP	0	JP	NN
JMPR	$	JR	DIS
JNC	0	JP	NC,NN
JNZ	0	JP	NZ,NN
JP	0	JP	P,NN
JPE	0	JP	PE,NN
JPO	0	JP	PO,NN
JRC	$	JR	C,DIS
JRNZ	$	JR	NZ,DIS
JRNC	$	JR	NC,DIS
JRZ	$	JR	Z,DIS
JZ	0	JP	Z,NN
LBCD	0	LD	BC,(NN)
LDAX	B	LD	A,(BC)
LDAX	D	LD	A,(DE)
LDA	0	LD	A,(NN)
LDAI		LD	A,I

LDD		LDD	
LDDR		LDDR	
LDED	0	LD	DE,(NN)
LDI		LDI	
LDIR		LDIR	
LHLD	0	LD	HL,(NN)
LIXD	0	LD	IX,(NN)
LIYD	0	LD	IY,(NN)
LSPD	0	LD	SP,(NN)
LXI	B,0	LD	BC,NN
LXI	D,0	LD	DE,NN
LXI	H,0	LD	HL,NN
LXI	SP,0	LD	SP,NN
LXI	X,0	LD	IX,NN
LXI	Y,0	LD	IY,NN
MOV	A,A	LD	A,A
MOV	A,B	LD	A,B
MOV	A,C	LD	A,C
MOV	A,D	LD	A,D
MOV	A,E	LD	A,E
MOV	A,H	LD	A,H
MOV	A,L	LD	A,L
MOV	A,M	LD	A,(HL)
MOV	B,A	LD	B,A
MOV	B,B	LD	B,B
MOV	B,C	LD	B,C
MOV	B,D	LD	B,D
MOV	B,E	LD	B,E
MOV	B,H	LD	B,H,NN
MOV	B,L	LD	B,L
MOV	B,M	LD	B,(HL)
MOV	C,A	LD	C,A
MOV	C,B	LD	C,B
MOV	C,C	LD	C,C
MOV	C,D	LD	C,D
MOV	C,E	LD	C,E
MOV	C,H	LD	C,H
MOV	C,L	LD	C,L
MOV	C,M	LD	C,(HL)
MOV	D,A	LD	D,A
MOV	D,B	LD	D,B
MOV	D,C	LD	D,C
MOV	D,D	LD	D,D
MOV	D,E	LD	D,E
MOV	D,H	LD	D,H
MOV	D,L	LD	D,L
MOV	D,M	LD	D,(HL)
MOV	E,A	LD	E,A
MOV	E,B	LD	E,B
MOV	E,C	LD	E,C
MOV	E,D	LD	E,D
MOV	E,E	LD	E,E
MOV	E,H	LD	E,H
MOV	E,L	LD	E,L
MOV	E,M	LD	E,(HL)
MOV	H,A	LD	H,A
MOV	H,B	LD	H,B
MOV	H,C	LD	H,C

MOV	H,D	LD	H,D
MOV	H,E	LD	H,E
MOV	H,H	LD	H,H
MOV	H,L	LD	H,L
MOV	H,M	LD	H,(HL)
MOV	L,A	LD	L,A
MOV	L,B	LD	L,B
MOV	L,C	LD	L,C
MOV	L,D	LD	L,D
MOV	L,E	LD	L,E
MOV	L,H	LD	L,H
MOV	L,L	LD	L,L
MOV	L,M	LD	L,(HL)
MOV	M,A	LD	(HL),A
MOV	M,B	LD	(HL),B
MOV	M,C	LD	(HL),C
MOV	M,D	LD	(HL),D
MOV	M,E	LD	(HL),E
MOV	M,H	LD	(HL),H
MOV	M,L	LD	(HL),L
MOV	A,[X+0]	LD	A,(IX+D)
MOV	B,[X+0]	LD	B,(IX+D)
MOV	C,[X+0]	LD	C,(IX+D)
MOV	D,[X+0]	LD	D,(IX+D)
MOV	E,[X+0]	LD	E,(IX+D)
MOV	H,[X+0]	LD	H,(IX+D)
MOV	L,[X+0]	LD	L,(IX+D)
MOV	A,[Y+0]	LD	A,(IY+D)
MOV	B,[Y+0]	LD	B,(IY+D)
MOV	C,[Y+0]	LD	C,(IY+D)
MOV	D,[Y+0]	LD	D,(IY+D)
MOV	E,[Y+0]	LD	E,(IY+D)
MOV	H,[Y+0]	LD	H,(IY+D)
MOV	L,[Y+0]	LD	L,(IY+D)
MOV	[X+0],A	LD	(IX+D),A
MOV	[X+0],B	LD	(IX+D),B
MOV	[X+0],C	LD	(IX+D),C
MOV	[X+0],D	LD	(IX+D),D
MOV	[X+0],E	LD	(IX+D),E
MOV	[X+0],H	LD	(IX+D),H
MOV	[X+0],L	LD	(IX+D),L
MOV	[Y+0],A	LD	(IY+D),A
MOV	[Y+0],B	LD	(IY+D),B
MOV	[Y+0],C	LD	(IY+D),C
MOV	[Y+0],D	LD	(IY+D),D
MOV	[Y+0],E	LD	(IY+D),E
MOV	[Y+0],H	LD	(IY+D),H
MOV	[Y+0],L	LD	(IY+D),L
MVI	A,0	LD	A,N
MVI	B,0	LD	B,N
MVI	C,0	LD	C,N
MVI	D,0	LD	D,N
MVI	E,0	LD	E,N
MVI	H,0	LD	H,N
MVI	L,0	LD	L,N
MVI	M,0	LD	(HL),N
MVI	[X+0],0	LD	(IX+D),N
MVI	[Y+0],0	LD	(IY+D),N

NEG	NEG
NOP	NOP
ORA A	OR A
ORA B	OR B
ORA C	OR C
ORA D	OR D
ORA E	OR E
ORA H	OR H
ORA L	OR L
ORA M	OR (HL)
ORA [X+0]	OR (IX+D)
ORA [Y+0]	OR (IY+D)
ORI 0	OR N
OTDR	OTDR
OTIR	OTIR
OUT 0	OUT (N),A
OUTP A	OUT (C),A
OUTP B	OUT (C),B
OUTP C	OUT (C),C
OUTP D	OUT (C),D
OUTP E	OUT (C),E
OUTP H	OUT (C),H
OUTP L	OUT (C),L
OUTI	OUTI
OUTD	OUTD
PCHL	JP (HL)
PCIX	JP (IX)
PCIY	JP (IY)
POP B	POP BC
POP D	POP DE
POP H	POP HL
POP PSW	POP AF
POP X	POP IX
POP Y	POP IY
PUSH B	PUSH BC
PUSH D	PUSH DE
PUSH H	PUSH HL
PUSH PSW	PUSH AF
PUSH X	PUSH IX
PUSH Y	PUSH IY
RAL	RLA
RALR A	RL A
RALR B	RL B
RALR C	RL C
RALR D	RL D
RALR E	RL E
RALR H	RL H
RALR L	RL L
RALR M	RL (HL)
RALR [X+0]	RL (IX+D)
RALR [Y+0]	RL (IY+D)
RAR	RRA
RARR A	RR A
RARR B	RR B
RARR C	RR C
RARR D	RR D
RARR E	RR E
RARR H	RR H

```
RARR    L
RARR    M
RARR    [X+0]
RARR    [Y+0]
RC
RES     0,A
RES     0,B
RES     0,C
RES     0,D
RES     0,E
RES     0,H
RES     0,L
RES     0,M
RES     1,A
RES     1,B
RES     1,C
RES     1,D
RES     1,E
RES     1,H
RES     1,L
RES     1,M
RES     2,A
RES     2,B
RES     2,C
RES     2,D
RES     2,E
RES     2,H
RES     2,L
RES     2,M
RES     3,A
RES     3,B
RES     3,C
RES     3,D
RES     3,E
RES     3,H
RES     3,L
RES     3,M
RES     4,A
RES     4,B
RES     4,C
RES     4,D
RES     4,E
RES     4,H
RES     4,L
RES     4,M
RES     5,A
RES     5,B
RES     5,C
RES     5,D
RES     5,E
RES     5,H
RES     5,L
RES     5,M
RES     6,A
RES     6,B
RES     6,C
RES     6,D
RES     6,E
```

```
RR      L
RR      (HL)
RR      (IX+D)
RR      (IY+D)
RET     C
RES     0,A
RES     0,B
RES     0,C
RES     0,D
RES     0,E
RES     0,H
RES     0,L
RES     0,(HL)
RES     1,A
RES     1,B
RES     1,C
RES     1,D
RES     1,E
RES     1,H
RES     1,L
RES     1,(HL)
RES     2,A
RES     2,B
RES     2,C
RES     2,D
RES     2,E
RES     2,H
RES     2,L
RES     2,(HL)
RES     3,A
RES     3,B
RES     3,C
RES     3,D
RES     3,E
RES     3,H
RES     3,L
RES     3,(HL)
RES     4,A
RES     4,B
RES     4,C
RES     4,D
RES     4,E
RES     4,H
RES     4,L
RES     4,(HL)
RES     5,A
RES     5,B
RES     5,C
RES     5,D
RES     5,E
RES     5,H
RES     5,L
RES     5,(HL)
RES     6,A
RES     6,B
RES     6,C
RES     6,D
RES     6,E
```

```
RES     6,H
RES     6,L
RES     6,M
RES     7,A
RES     7,B
RES     7,C
RES     7,D
RES     7,E
RES     7,H
RES     7,L
RES     7,M
RES     0,[X+0]
RES     1,[X+0]
RES     2,[X+0]
RES     3,[X+0]
RES     4,[X+0]
RES     5,[X+0]
RES     6,[X+0]
RES     7,[X+0]
RES     0,[Y+0]
RES     1,[Y+0]
RES     2,[Y+0]
RES     3,[Y+0]
RES     4,[Y+0]
RES     5,[Y+0]
RES     6,[Y+0]
RES     7,[Y+0]
RET
RETN
RETI
RLC
RLCR    A
RLCR    B
RLCR    C
RLCR    D
RLCR    E
RLCR    H
RLCR    L
RLCR    M
RLCR    [X+0]
RLCR    [Y+0]
RLD
RM
RNC
RNZ
RP
RPE
RPO
RRC
RRCR    A
RRCR    B
RRCR    C
RRCR    D
RRCR    E
RRCR    H
RRCR    L
RRCR    M
RRCR    [X+0]
```

```
RES     6,H
RES     6,L
RES     6,(HL)
RES     7,A
RES     7,B
RES     7,C
RES     7,D
RES     7,E
RES     7,H
RES     7,L
RES     7,(HL)
RES     0,(IX+D)
RES     1,(IX+D)
RES     2,(IX+D)
RES     3,(IX+D)
RES     4,(IX+D)
RES     5,(IX+D)
RES     6,(IX+D)
RES     7,(IX+D)
RES     0,(IY+D)
RES     1,(IY+D)
RES     2,(IY+D)
RES     3,(IY+D)
RES     4,(IY+D)
RES     5,(IY+D)
RES     6,(IY+D)
RES     7,(IY+D)
RET
RETN
RETI
RLCA
RLC     A
RLC     B
RLC     C
RLC     D
RLC     E
RLC     H
RLC     L
RLC     (HL)
RLC     (IX+D)
RLC     (IY+D)
RLD
RET     M
RET     NC
RET     NZ
RET     P
RET     PE
RET     PO
RRCA
RRC     A
RRC     B
RRC     C
RRC     D
RRC     E
RRC     H
RRC     L
RRC     (HL)
RRC     (IX+D)
```

RRCR	[Y+0]	RRC	(IY+D)
RRD		RRD	
RST	0	RST	0
RST	1	RST	8
RST	2	RST	10H
RST	3	RST	18H
RST	4	RST	20H
RST	5	RST	28H
RST	6	RST	30H
RST	7	RST	38H
RZ		RET	Z
SBB	A	SBC	A,A
SBB	B	SBC	A,B
SBB	C	SBC	A,C
SBB	D	SBC	A,D
SBB	E	SBC	A,E
SBB	H	SBC	A,H
SBB	L	SBC	A,L
SBB	M	SBC	A,(HL)
SBB	[X+0]	SBC	A,(IX+D)
SBB	[Y+0]	SBC	A,(IY+D)
SBCD	0	LD	(NN),BC
SBI	0	SBC	A,N
SDED	0	LD	(NN),DE
SET	0,A	SET	0,A
SET	0,B	SET	0,B
SET	0,C	SET	0,C
SET	0,D	SET	0,D
SET	0,E	SET	0,E
SET	0,H	SET	0,H
SET	0,L	SET	0,L
SET	0,M	SET	0,(HL)
SET	1,A	SET	1,A
SET	1,B	SET	1,B
SET	1,C	SET	1,C
SET	1,D	SET	1,D
SET	1,E	SET	1,E
SET	1,H	SET	1,H
SET	1,L	SET	1,L
SET	1,M	SET	1,(HL)
SET	2,A	SET	2,A
SET	2,B	SET	2,B
SET	2,C	SET	2,C
SET	2,D	SET	2,D
SET	2,E	SET	2,E
SET	2,H	SET	2,H
SET	2,L	SET	2,L
SET	2,M	SET	2,(HL)
SET	3,A	SET	3,A
SET	3,B	SET	3,B
SET	3,C	SET	3,C
SET	3,D	SET	3,D
SET	3,E	SET	3,E
SET	3,H	SET	3,H
SET	3,L	SET	3,L
SET	3,M	SET	3,(HL)
SET	4,A	SET	4,A
SET	4,B	SET	4,B

SET	4,C
SET	4,D
SET	4,E
SET	4,H
SET	4,L
SET	4,M
SET	5,A
SET	5,B
SET	5,C
SET	5,D
SET	5,E
SET	5,H
SET	5,L
SET	5,M
SET	6,A
SET	6,B
SET	6,C
SET	6,D
SET	6,E
SET	6,H
SET	6,L
SET	6,M
SET	7,A
SET	7,B
SET	7,C
SET	7,D
SET	7,E
SET	7,H
SET	7,L
SET	7,M
SET	0,[X+0]
SET	1,[X+0]
SET	2,[X+0]
SET	3,[X+0]
SET	4,[X+0]
SET	5,[X+0]
SET	6,[X+0]
SET	7,[X+0]
SET	0,[Y+0]
SET	1,[Y+0]
SET	2,[Y+0]
SET	3,[Y+0]
SET	4,[Y+0]
SET	5,[Y+0]
SET	6,[Y+0]
SET	7,[Y+0]
SHLD	0
SIXD	0
SIYD	0
SLAR	A
SLAR	B
SLAR	C
SLAR	D
SLAR	E
SLAR	H
SLAR	L
SLAR	M
SLAR	[X+0]

SET	4,C
SET	4,D
SET	4,E
SET	4,H
SET	4,L
SET	4,(HL)
SET	5,A
SET	5,B
SET	5,C
SET	5,D
SET	5,E
SET	5,H
SET	5,L
SET	5,(HL)
SET	6,A
SET	6,B
SET	6,C
SET	6,D
SET	6,E
SET	6,H
SET	6,L
SET	6,(HL)
SET	7,A
SET	7,B
SET	7,C
SET	7,D
SET	7,E
SET	7,H
SET	7,L
SET	7,(HL)
SET	0,(IX+D)
SET	1,(IX+D)
SET	2,(IX+D)
SET	3,(IX+D)
SET	4,(IX+D)
SET	5,(IX+D)
SET	6,(IX+D)
SET	7,(IX+D)
SET	0,(IY+D)
SET	1,(IY+D)
SET	2,(IY+D)
SET	3,(IY+D)
SET	4,(IY+D)
SET	5,(IY+D)
SET	6,(IY+D)
SET	7,(IY+D)
LD	(NN),HL
LD	(NN),IX
LD	(NN),IY
SLA	A
SLA	B
SLA	C
SLA	D
SLA	E
SLA	H
SLA	L
SLA	(HL)
SLA	(IX+D)

SLAR	[Y+0]	SLA	(IY+D)
SPHL		LD	SP,HL
SPIX		LD	SP,IX
SPIY		LD	SP,IY
SRAR	A	SRA	A
SRAR	B	SRA	B
SRAR	C	SRA	C
SRAR	D	SRA	D
SRAR	E	SRA	E
SRAR	H	SRA	H
SRAR	L	SRA	L
SRAR	M	SRA	(HL)
SRAR	[X+0]	SRA	(IX+D)
SRAR	[Y+0]	SRA	(IY+D)
SRLR	A	SRL	A
SRLR	B	SRL	B
SRLR	C	SRL	C
SRLR	D	SRL	D
SRLR	E	SRL	E
SRLR	H	SRL	H
SRLR	L	SRL	L
SRLR	M	SRL	(HL)
SRLR	[X+0]	SRL	(IX+D)
SRLR	[Y+0]	SRL	(IY+D)
SSPD	0	LD	(NN),SP
STAX	B	LD	(BC),A
STAX	D	LD	(DE),A
STA	0	LD	(NN),A
STAI		LD	I,A
STC		SCF	
SUB	A	SUB	A
SUB	B	SUB	B
SUB	C	SUB	C
SUB	D	SUB	D
SUB	E	SUB	E
SUB	H	SUB	H
SUB	L	SUB	L
SUB	M	SUB	(HL)
SUB	[X+0]	SUB	(IX+D)
SUB	[Y+0]	SUB	(IY+D)
SUI	0	SUB	N
XCHG		EX	DE,HL
XRA	A	XOR	A
XRA	B	XOR	B
XRA	C	XOR	C
XRA	D	XOR	D
XRA	E	XOR	E
XRA	H	XOR	H
XRA	L	XOR	L
XRA	M	XOR	(HL)
XRA	[X+0]	XOR	(IX+D)
XRA	[Y+0]	XOR	(IY+D)
XRI	0	XOR	N
XTHL		EX	(SP),HL
XTIX		EX	(SP),IX
XTIY		EX	(SP),IY

4-21-81

SECTION 2: GETTING CP/M RUNNING WITH THE INTERFACE

One of the major problems confronting the implementers of new micro-computer systems, has been the lack of input/output (I/O) standards. The emergence of Digital Research's CP/M(r) disk operating system as a standard I/O environment has contributed greatly to alleviating this problem. Now the problem is reduced to implementing CP/M on the target hardware system, which consists of tailoring the BIOS part of CP/M to the situation. Unfortunately, since we can't assume any particular console interface at the factory, there is no way to make the system generation completely automatic.

Because of all the different possible system configurations, and because we try to update our hardware and software as quickly as possible, it has been difficult to create and maintain a set of documentation that is useful and correct for getting our double-density floppy disk interface working under CP/M. These instructions represent a major rewrite effort in this direction. We hope that most of the faults in the earlier instructions have been corrected in this set.

These instructions explain how to get the Tarbell Double Density Floppy Disk Interface going with Digital Research's CP/M 1.4 or 2.x disk operating system. It is important not to try and make more than one change in your system at a time. For example, if you wish to go from our single density interface operating under CP/M 1.4, to our double density interface operating DMA under CP/M 2.2 with a different memory size, DON'T try to do it all at once. First the single-density to double density, then to DMA, then to 2.2, then to different memory size.

Be sure that the title of the instructions you are going to use, matches the situation you have. If it doesn't, and you can't seem to find one that does match, call or write to us, and we'll try to help. If you don't think you are capable of carrying out the required instructions yourself, we can generate a customized system for you. Just have us send you an I/O tailoring questionaire. The cost is usually about $50.

INSTALLATION NOTES

1. The MWRITE option available on our board is only for those computers where the MWRITE is not generated directly from the bus signals PWR and SOUT. It is not intended as a substitute for the normal MWRITE line, which must be implemented somewhere on the bus (usually on the CPU or front panel).

2. Install the double density board as close to the CPU as possible, between the CPU and memory board(s).

INSTRUCTIONS FOR GETTING THE TARBELL DOUBLE DENSITY INTERFACE OPERATING WITH CP/M 1.4 WHEN YOU HAVE CP/M 1.4 ALREADY GOING ON A TARBELL SINGLE-DENSITY INTERFACE

1. First make sure that your situation matches the title above. If it doesn't, find another sheet that does match.

2. Check the option jumpers on your double-density interface board against the manual to make sure the board is addressed for E0 through F8 (hex), and that all other options are correct.

Use your current single-density interface, operating under CP/M 1.4 to do the following steps:

3. Use the FORMAT91 program on the public domain #2 disk (provided with the interface) to format at least two disks. DON'T use any of your old format programs to do this. When it says "READY TO FORMAT?" be SURE to get the public domain disk out of there before typing Y. Test the disks using the DISKTEST program.

4. Put one of the newly formatted disks in drive B. Put a disk with your normal CP/M 1.4 system and system programs in drive A. Now perform the following steps:

a) logged into drive A, type SYSGEN. Answer source as drive A, destination as drive B. Reboot.

b) type PIP with no arguments, then the following steps.
```
*B:=A:DDT.COM
*B:=A:ASM.COM
*B:=A:SYSGEN.COM
*B:=A:ED.COM
```

c) while still in PIP program, remove your system diskette from drive A, and insert into drive A the Public Domain #2 diskette that came with the double-density interface. Then continue as shown below:
```
*B:=A:ABIOS24.ASM
*B:=A:DBOOT24.ASM
```

5. Now take out the public domain disk #2 and put it aside. Take the newly formatted disk out of drive B and put it into drive A. Boot up on it. It should come up normally, since a copy of your system was just put onto it.

6. Using ED.COM, edit the ABIOS24.ASM to change the EQU's for your memory size, console, printer, drives, etc. Leave the DMACNTL and DUBSID EQU's set to FALSE. Set the MSIZE EQU to the same size as the CP/M 1.4 system you are now running on this disk. Be sure to set the console port numbers correctly. Exit from the editor. Rename the file to ABIOSxx.ASM, where xx is your MSIZE.

8. Assemble ABIOSxx with ASM.COM. Print the .PRN file if desired, then erase it.

9. Using ED.COM, edit DBOOT24.ASM. Set the MSIZE EQU to the

size used above. Leave the DOUBSID, DOUBDEN, and DMACNTL EQU's set to FALSE. Exit from the editor. Rename the file to DBOOTxx.ASM.

10. Assemble DBOOTxx.ASM with ASM.COM. Print the .PRN file if desired, then erase it.

11. Use SYSGEN to put a copy of your current CP/M 1.4 system onto the disk as a file. When it asks for source, answer A. When it asks for destination, press carriage-return to reboot. Then do a SAVE 32 CPMxx.COM, where xx is your system size.

12. Use DDT to bring in the CPMxx.COM file and to overlay the BIOS and BOOT hex files onto it. Type DDT CPMxx.COM . Then type IABIOSxx.HEX . Then type Rbias where bias is in the table below:

xx	bias	xx	bias	xx	bias	xx	bias
20	D480	24	C480	28	B480	32	A480
36	9480	40	8480	44	7480	48	6480
52	5480	56	4480	60	3480	64	2480

Now type IDBOOTxx.HEX . Then type R900 . Then do Ctl-C.

13. Next enter SYSGEN. When it asks for source, press return to skip. When it asks for destination, type A. At this point you may write this system onto more than one disk. After you are finished writing onto the disk(s), DON'T press return to reboot.

14. You can now shut off your computer, remove the single density interface, and put the double-density interface in. Then turn your computer back on.

15. The system you have just written onto one or more disks should now boot up correctly on the double-density interface. If it doesn't, check over the BIOS and BOOT .PRN files to make sure all EQU's were set correctly. Check your board to verify again that all the jumper options are right. If you still can't get it going, read section 2-3 of these instructions.

16. If the system does come up correctly, congratulations! You are now running the double-density interface in non-DMA mode. If you want to operate double-density next, see section 2-2 of these instructions. If you want to try operating in DMA mode, go to step 6 in this section, changing the DMACNTL EQU to TRUE in both the BIOS and the BOOT .ASM files. The rest of the instructions are the same.

17. Finally, if you notice any errors in this documentation, PLEASE call or write about it.

HOW TO MAKE THE TARBELL DOUBLE DENSITY INTERFACE OPERATE IN THE DOUBLE DENSITY MODE ASSUMING YOU HAVE THE DOUBLE DENSITY INTERFACE OPERATING IN THE SINGLE DENSITY MODE.

1. Check your situation against the title above. If it doesn't match, look for other instructions that do. In order to operate in double density mode, you will either need to be operating at 4 or above 4 Mhz (Z80 or 8085), or you need to be operating in DMA mode. To set DMA mode, see step 16 of the instructions in section 2-1.

2. Format some disks double density with DFORMAT, and test them using DTEST.

3. If you boot up on a single density system which was created using the auto-select I/O section (ABIOS or 2ABIOS), all you have to do is put the formatted double-density diskette in drive B. Files may be transferred to the new double density disk using PIP. Don't try to use the COPY utility to copy from single density to double density or vise-versa.

4. If you want to put a system from the first two tracks on the single density disk onto a double density disk, SYSGEN alone will not work. This is because the first sector of the first track contains a byte which has to be DD (hex) for double density, and your single-density disk doesn't have that byte. To perform this operation correctly, follow these steps:

a) On your single density disk, edit the file called DBOOTxx.ASM to change the DOUBDEN EQU from FALSE to TRUE. It is important that the MSIZE match your current CP/M system size (xx).
b) Assemble the new file: ASM DBOOTxx
c) Do a SYSGEN, answering source on A, skip the destination and reboot. Enter SAVE 34 CPMxx.COM where xx is system size.
d) Then overlay the CPMxx.COM system image with the new DBOOT:
```
DDT CPMxx.COM
IDBOOTxx.HEX
R900
```
e) Then press control-C to return to CP/M.
f) Type B:,<cr> to log in drive B.
Type A:,<cr>
g) Then do another SYSGEN, this time skipping the source, and answering B to the destination. (This assumes you still have your double-density disk in B.)

5. Now you can take the double density disk out of drive B and put it into drive A and boot up on it.

WHAT TO DO IF YOUR TARBELL DOUBLE DENSITY FLOPPY INTERFACE IS NOT WORKING

1. Recheck the jumper options on the interface board against your manual in section 6. Note that manuals of boards rev B and earlier have an error in the board addressing section. People with these manuals can get a new manual free by sending us the cover of their old manual.

2. Recheck the EQU's in the BIOS and BOOT .ASM files to make sure that all are set correctly.

3. If you have a friend with a working Tarbell Double Interface, try using your interface in his computer. If his works and yours doesn't, there is probably something actually wrong with your interface. If so, you might want to consider sending it back to Tarbell for repair. If your interface does work in your friend's computer, the problem might be in your software, or in some other component of your system. Just because the other components of your system work under other circumstances, doesn't mean that there is nothing wrong with them.

4. Another thing to check is the diskette that you're using. Is it formatted correctly? How do you know it is?

5. Do you have dynamic memory in your computer. If so, how is it refreshed? It is possible that the way it is refreshed interferes with our interface, or that the way our interface works interferes with the memory's refresh circuitry.

6. Does your CPU board fully implement the new IEEE S-100 standard? In particular, does it use pin 67 (the phantom line) for anything besides phantom? Does it implement the control-disable, data-disable, and status-disable lines? Does it implement the PSYNC, PHOLD, and PHLDA lines? Neither the SDS SBC-100 or SBC-200 CPU boards meet this requirement.

7. Does the memory which occupies address 0000 in your system have a phantom line on pin 67?

8. Do you have other boards in your system that use the XRDY and PRDY lines (pins 3 and 72) besides the Tarbell interface and the CPU? If so, it might be best to disconnect those lines completely.

9. Since the Tarbell Double Density Floppy Disk Interface uses lines on your motherboard that aren't normally used, some of these lines could be shorted or open, or the connector pins could be dirty.

10. Check your system power supply, with a scope if possible, to make sure that all your voltages are steady, clean, and the right level, both on the drives and the motherboard. It is very important that on the drive power supply, the 24 volt, 5 volt, and -5 volt returns be connected together at the power supply end.

11. If you are having problems with the bootstrap, it's possible that C17 is not quite the right value to reduce the effects of ringing on the bus. You might try 100, 220, 390, 470, or 680 pf capacitors, in that order. The symptom is that the bootstrap flip-flop gets reset before it has a chance to read a complete sector if the value of C17 is to too small, or, the bootstrap will not release soon enough if the value of C17 is to big. The value of C17 is not as critical as it may seem, but it's value in some cases will be controlled to some extent by the noise and impedance effects of your computer bus. If you have a good fast scope, you can observe the bootstrap operation by looking at pin 19 on the 8257. This is the DRQ line, and it should be a series of short pulses that happen over a period of about 2 ms. If they don't last that long, you may have to adjust C17.

12. If the DRQ line mentioned above never goes high at all, that means the interface is never receiving a valid data byte. This could be caused by a variety of factors, including a bad data separator component, bad 1793, bad drive, etc.

13. If the interface is picking up excessive errors after warming up, it could be the 1793. We are now testing these IC's more carefully.

*** NOTE ***

If you decide to send the interface back for repair, be sure to include a copy of your receipt, showing the date you bought it.

INSTRUCTIONS FOR GETTING THE TARBELL DOUBLE DENSITY INTERFACE OPERATING WITH CP/M 2.x WHEN YOU HAVE CP/M 2.x ALREADY GOING ON A TARBELL SINGLE-DENSITY INTERFACE

1. First make sure that your situation matches the title above. If it doesn't, find another sheet that does match.

2. Check the option jumpers on your double-density interface board against the manual to make sure the board is addressed for E0 through F8 (hex), and that all other options are correct.

Use your current single density interface, operating under CP/M 2.x to do the following steps:

3. Use the FORMAT91 program on the public domain #2 disk (provided with the interface) to format at least two disks. DON'T use any of your old format programs to do this. When it says "READY TO FORMAT?" be SURE to get the public domain disk out of there before typing Y. Test the disks using the DISKTEST program.

4. Put one of the newly formatted disks in drive B. Put a disk with your normal CP/M 2.x system and system programs in drive A. Now perform the following steps:

a) logged into drive A, type SYSGEN. Answer source as drive A, destination as drive B. Reboot.
b) type PIP with no arguments, then the following steps.
```
*B:=A:CPM.COM
*B:=A:DDT.COM
*B:=A:ASM.COM
*B:=A:SYSGEN.COM
*B:=A:ED.COM
```
c) while still in the PIP program, remove your system diskette from drive A, then insert into drive A the Public Domain #2 diskette that came with the double-density interface. Then continue as shown below:
```
*B:=A:2ABIOS24.ASM
*B:=A:2DBOOT24.ASM
```

5. Now take out the public domain disk #2 and put it aside. Take the newly formatted disk out of drive B and put it into drive A. Boot up on it. It should come up normally, since a copy of your system was just put onto it.

6. Using ED.COM, edit the 2ABIOS24.ASM to change the EQU's for your memory size, console, printer, drives, etc. Leave the DMACNTL and DUBSID EQU's set to FALSE. Set the MSIZE EQU to the same size as the CP/M 2.x system you are now running on this disk. Be sure to set the console port numbers correctly. If you have Shugart 800 drives, don't set the step rate any faster than 10 ms. Exit from the editor. Rename the file to 2ABIOSxx.ASM, where xx is your MSIZE.

8. Assemble 2ABIOSxx with ASM.COM. Print the .PRN file if desired, then erase it.

9. Using ED.COM, edit 2DBOOT24.ASM. Set the MSIZE EQU to the size used above. Leave the DOUBSID and DMACNTL EQU's set to FALSE. Exit from the editor. Rename the file to 2DBOOTxx.ASM.

10. Assemble 2DBOOTxx.ASM with ASM.COM. Print the .PRN file if desired, then erase it.

11. Use SYSGEN to put a copy of your current CP/M 2.x system onto the disk as a file. When it asks for source, answer A. When it asks for destination, press carriage-return to reboot. Then do a SAVE 34 CPMxx.COM, where xx is your system size.

12. Use DDT to bring in the CPMxx.COM file and to overlay the BIOS and BOOT hex files onto it. Type DDT CPMxx.COM . Then type I2ABIOSxx.HEX . Then type Rbias where xx is MSIZE and bias is in the table below:

xx	bias	xx	bias	xx	bias	xx	bias
20	D580	24	C580	28	B580	32	A580
36	9580	40	8580	44	7580	48	6580
52	5580	56	4580	60	3580	64	2580

Now type I2DBOOTxx.HEX . Then type R900 . Then do Ctl-C.

13. Next enter SYSGEN. When it asks for source, press return to skip. When it asks for destination, type A. At this point you may write this system onto more than one disk. After you are finished writing onto the disk(s), DON'T press return to reboot.

14. You can now shut off your computer, remove the single-density interface, and put the double-density interface in. Then turn your computer back on.

15. The system you have just written onto one or more disks should now boot up correctly on the double-density interface. If it doesn't, check over the BIOS and BOOT .PRN files to make sure all EQU's were set correctly. Check your board to verify again that all the jumper options are right. If you still can't get it going, read section 2-3 of these instructions.

16. If the system does come up correctly, congratulations! You are now running the double-density interface in non-DMA mode. If you want to operate double-density next, see section 2-2 of these instructions. If you want to try operating in DMA mode, go to step 6 in this section, changing the DMACNTL EQU to TRUE in both the BIOS and the BOOT .ASM files. The rest of the instructions are the same.

17. Finally, if you notice any errors in this documentation, PLEASE call or write about it.

GETTING THE TARBELL VERSION OF CP/M 1.4 OR 2.X RUNNING ON YOUR TARBELL DOUBLE-DENSITY FLOPPY DISK INTERFACE WITHOUT A CURRENTLY RUNNING CP/M SYSTEM OF ANY KIND

1. First make sure that your situation matches the title above. If not, you may find that another set of instructions will get your system going sooner.

2. You need to have the following hardware installed:

   a) An assembled and tested Tarbell Double Density Interface
   b) At least 24k bytes of random access memory, of which at least the first 32 bytes can be disabled by phantom line pin 67 going low.
   c) A Z-80, 8085, or 8080 CPU board which conforms to the IEEE S-100 standard.
   d) A console interface of some type, preferably not memory-mapped video, which supports an alphanumeric keyboard and a CRT display or teleprinter. If possible, this interface should be addressed for status on port 0, data on port 1, with bit 0 of the status low meaning keyboard ready, and with bit 7 of the status low meaning CRT display ready. If these port and status requirements are met, the Tarbell CP/M 1.4 or 2.x disks for the DD controller should boot up with no further work. Just put the disk in, push reset, and run. Skip to step 8 if so. If not, you will need to fulfill the requirements of substep (e) below and continue.
   e) Either a front panel or a ROM monitor (any ROM should be outside the 24k RAM), which allows depositing bytes into specified RAM addresses and executing at an address.

3. If possible, have a friend make a copy of your original CP/M disk, and don't use it except to make further copies. Then use the copy for the following steps.

4. Turn the computer on, then the CRT-keyboard, then the drive power.

5. Put the CP/M disk into the disk drive (on most drives, the label on the disk should face the door of the drive). Close the door. Push reset (and run if you have one) buttons on the computer.

6. The head should load against the disk and move in one track. If it doesn't do this, something is wrong with the hardware setup, and you should try a few times more. If it still doesn't do it, FIRST remove the diskette, then shut down the system. Something is either wrong with the hardware or the diskette. If so, have someone look at it or call Tarbell. If it does load and step ok, go onto the next step.

7. Either stop the computer from running, if you have a front panel, or jump into your ROM monitor, if you have one.

8. Look at the BIOS (Basic Input Output System) listing that

came with our CP/M. Find the label BOOT. After the LXI SP instruction, you will see a series of NOP's. This area is reserved for initializing console interfaces that require it. Using either front panel or ROM, deposit the initialization routine required, if any, at the address indicated by the listing. There should be a copy of any required initialization routine in the manual on your console interface. Assembly language code for the initialization of some common console interfaces can be seen in the following lines on the page.

9. Still looking at the BIOS listing, find the label CONST. Examine the code there for our "standard" interface. Put the code here to do a status check on your console interface. Notice that if your status bits are true when high, instead of low like ours, you will need to change the RNZ to an RZ. Other changes which might be required are the port number after the IN, and the mask after the ANI. Check your console interface manual for examples and instructions.

10. The next routine is labeled CONIN. Deposit the code to read a byte from your console keyboard into register A. Notice that you might need to make similar changes, such as JNZ to JZ, mask, and port numbers.

11. The last routine to change is labeled CONOT. Deposit the code to write the byte in register C to your console. Again, you might need to replace our JNZ with a JZ and make port number and mask changes. Be sure to end each of these routines with an RET instruction.

12. This should be all the patches you need to make to the CP/M system residing in memory, to get going temporarily. Now examine the content of address 5A00 (hex), which should be a C3 (hex for JMP) and execute (run) at that location.

13. Our BIOS should give you an opening message. If so, you're on the air, so go to step 14. If not, the system may not have loaded properly, and something may be wrong with the diskette or hardware setup. In that case, refer to section 2-3.

14. If you haven't already done so, copy the system and files onto another disk. In order to do this, keep your system disk in drive A and put a blank disk into drive B. Then type: COPY ALL. This will copy your original disk onto the blank disk. Note that the system you are running is only in memory, and the system on the disk hasn't yet been modified. Leave the new disk in drive B until you press return to reboot. Then take the original disk out of drive A and never use it again except to copy it. Now remove the copy you made from drive B and label it exactly the same as the original. You will find that it is important to keep the disk labels current, as it is easy to get confused and make a mistake. Put the new copy into drive A for further work. Then press Ctl-C.

15. The next thing to do is edit the BIOS and BOOT .ASM files and overlay them onto your system. Use the method described in the

Tarbell CP/M 1.4 or 2.x User's Guide, as this will properly document all your changes and allow you to make use of memory larger than 24k.

16. The latest ABIOS (Auto-density select Basic Input Output System) is always available from Tarbell for $15. Just ask for Public Domain Disk # 2.

CONTENTS OF PUBLIC DOMAIN DISK #2 NOV 5, 1980

This is a disk which is regularly updated with our latest ABIOS and 2ABIOS for the Tarbell Double Density Floppy Disk Interface. Other utilities are also maintained on this disk, such as format and test routines. We also had room to include the source for the FORTH language from the Forth Interest Group. Their name and address are included on the .ASM file. Following is a short description of each file. For further information, see the comments in the file itself, or the Tarbell CP/M User's Guide. The latest version of this disk is always available within 1 week from Tarbell for $15. If you are having problems, it's always wise to see if there is a newer version of this disk available than the one you have.

1. DBOOT24.ASM

This is the secondary coldstart loader for CPM 1.4 only.

2. DDUMP.ASM & DDUMP.COM

THIS IS A MODIFIED DUMP.COM FROM SAM SINGER AND FROM THE CP/M USERS GROUP. THIS PROGRAM WILL ALLOW YOU TO VIEW ONLY TRACKS 2 - 76 OF A DOUBLE DENSITY DISK. THE ONLY LIMITATION IN THE PROGRAM IS THAT IT WILL NOT DUMP BY GROUP NUMBERS. ALL OTHER FEATURES ARE USEABLE.

3. DFORMAT.ASM & DFORMAT.COM

This is the double density format program. It will also format a double sided disk if asked to. This format program formats track 0 = 26, 128 byte sectors, tracks 1 - 76 in 51, 128 bytes sectors.

4. DTEST.ASM & DTEST.COM

THIS PROGRAM IS USED TO TEST A DOUBLE DENSITY DISK FOR ERRORS. WHEN THE PROGRAM FIRST COMES UP IT WILL ASK YOU FOR A "TITLE:". YOU MAY TYPE IN ANYTHING YOU WANT SUCH AS <FORMATTED WITH 62.5 NSEC,187.5 NSEC> AND THEN A CNTL-P, CARRIAGE RET, OR IF YOU DON'T WANT TO TYPE ANYTHING, JUST TYPE A CARRIAGE RET. THE TITLE ALLOWS YOU TO KEEP A RUNNING TAB ON THE ERRORS AND USING CNTL-P WILL TURN ON THE LIST DEVICE FOR MAKING A HARDCOPY LISTING. THE NEXT QUESTION WILL BE STARTING TRACK. YOU MUST ANSWER THIS WITH A TRACK NUMBER OF 0 OR GREATER. THE REST OF THE PROGRAM SHOULD BE CLEAR. THIS PROGRAM READS A TRACK AT A TIME AND KEEPS A RUNNING TAB OF ERRORS FOUND. DURING THE READING OF THE TRACK, IF A SECTOR IS BAD IT WILL DISPLAY THE SECTOR NUMBER AND THE NUMBER OF RETRYS IT TOOK TO READ IT. IT SHOULD BE NOTED THAT IT WILL DO 11 RETRYS MAX, AND THEN GO ON TO THE NEXT SECTOR. IF IT TAKES MORE THAN 10 RETRYS, THEN YOU SHOULD REFORMAT THE DISK AND CHECK IT AGAIN, AS OUR DBIOS ONLY DOES 10 RETRYS BEFORE INDICATING A FAILURE. RETRYS ON THE ORDER OF 1 TO 5 IS TYPICAL, IF THEY OCCUR AT ALL, WITH THIS INTERFACE. THIS PROGRAM DOES NOT WRITE ON THE DISK, IT IS READ ONLY.

5. FORMAT.ASM & FORMAT.COM

IF YOU ARE USING OUR OLD SINGLE DENSITY FORMAT PROGRAM, YOU WILL NOT BE ABLE TO READ THEM ON THE NEW INTERFACE IN SINGLE DENSITY. THIS IS BECAUSE THERE IS A BYTE IN THE INNER RECORD GAPS THAT THE 1771 WILL

READ BUT THE 1791/1793 WON'T. THIS FORMAT PROGRAM FIXES THAT PROBLEM FOR BOTH THE 1791/1793 AND WILL STILL ALLOW YOU TO USE IT WITH YOUR PRESENT 1771 CONTROLLER ALSO. YOU SHOULD DESTROY AND OLD COPIES OF THE OLD FORMAT PROGRAM YOU HAVE, AND USE THIS ONE FROM HERE ON OUT. THIS PROGRAM ALSO FORMATS A DOUBLE SIDED DISK.

*** NOTE ***

THIS FORMATS SINGLE DENSITY ONLY, 26 SECTORS OF 128 BYTES AND ONLY RUNS ON THE NEW CONTROLLER BOARD.

6. FORMAT91.ASM & FORMAT91.COM

This program will only run on the single-density interface. It will format disks in standard IBM single-density format, to read correctly on the double density interface.

7. DFRAND.ASM

This is another format program, which only runs on the double-density interface, and which formats disks double-density in a random format. This is very useful to use in conjunction with the DTEST program, while setting up precomp. It gives a more realistic representation of the way that data may be present on the disk. Do NOT use this program to format disks that are to be used next with CP/M, as the directory needs to be filled with E5's.

8. MACRO.LIB & SKEW.LIB

THIS LIBRARY IS NECESSARY IF YOU HAVE DIGITALS MACRO ASSEMBLER AND WISH TO CHANGE AND ASSEMBLE DDUMP.ASM AND DTEST.ASM. THESE PROGRAMS USE MACROS.

9. STAT.COM (FOR CPM V1.4 ONLY)

THIS IS AN UPDATED VERSION OF THE STAT PROGRAM FOR THE ORIGINAL DISTURBUTION. IT FUNCTIONS THE SAME AS THE OLD ONE. THE ONLY IMPROVEMENT WAS TO MAKE IT DISPLAY THE CORRECT CAPACITY OF A DOUBLE DENSITY DISK. IT WILL STILL WORK SINGLE DENSITY.

10. ABIOS24.ASM

THIS IS THE AUTO-DENSITY SELECT VERSION OF THE BIOS FOR CPM V1.4. THIS BIOS WILL AUTOMATICALLY SELECT THE DENSITY OF THE DISK YOU ARE USING IN EITHER DRIVE, AND WILL ALLOW YOU TO CHANGE THE DENSITY AT ANY TIME. IF YOU ARE GOING TO CHANGE THE DENSITY OF THE "A" DRIVE, YOU MUST HAVE A DISK WITH THE SAME SYSTEM SIZE AS THE ONE YOU REMOVED. FILE TRANSFERS FROM SINGLE TO DOUBLE OR DOUBLE TO SINGLE IS COMPLETELY AUTOMATIC. YOU MUST SET DOUBDEN = TRUE BEFORE YOU USE THE AUTO-DENSITY CAPABILITY OF ABIOS24.ASM, AS THIS IS THE ONLY WAY THE PROGRAM KNOWS IT IS LOOKING FOR A DOUBLE DENSITY DISK IN ANY DRIVE.

11. 2ABIOS24.ASM

THIS IS THE AUTO-DENSITY SELECT VERSION FOR CPM V2.x AND THE NEW INTERFACE. THIS BIOS MUST BE USED WITH 2DBOOT24.ASM TO BRING UP THE SYSTEM. PLEASE NOTE THAT 2ABIOS24 AND 2DBOOT24 ARE ONLY FOR CPM V2.x AND WILL NOT RUN ON CPM V1.4 OR CONVERSELY.

*** NOTE ***

YOU MUST SET DMACNTL = TRUE IF YOU WILL BE RUNNING DOUBLE DENSITY at 2 MHz.

12. 2DBOOT24.ASM

THIS IS THE SECONDARY COLD START LOADER FOR CPM V2.x FOR USE WITH 2ABIOS24.ASM. SEVERAL EQU'S APPEAR IN THIS LOADER. DMACNTL - SETTING THIS TRUE WILL ALLOW THE PROGRAM TO BOOT IN THE SYSTEM USING DMA CONTROL. IF FALSE, BOOTS SYSTEM UNDER PROGRAM DATA TRANSFER. DOUBDEN - SETTING THIS TRUE PUTS THE SPECIAL ID BYTE INTO THE DISK DURING GENERATION OF A DOUBLE DENSITY SYSTEM DISK THAT WILL BE BOOTED IN FROM DRIVE 'A'. SETTING THIS FALSE ALLOWS BUILDING A SYSTEM ON A SINGLE DENSITY DISK. THIS BYTE IS HOW THE SYSTEM KNOWS WHETHER OR NOT A SINGLE OR DOUBLE DENSITY IS ON LINE.

**** NOTE ****

IF YOU HAVE TROUBLE READING A SINGLE DENSITY DISK ON THIS CONTROLLER, YOU MAY HAVE A DISK WITH THE WRONG SECTOR FORMATTING. TO FIND OUT, TAKE ANOTHER DISK AND USE THE NEW FORMAT.COM FILE ON THIS DISK TO REFORMAT IT. THEN USING YOUR OLD CONTROLLER, TRANSFER ALL THE PROGRAMS YOU WISH TO SAVE FROM THE DISK THAT WOULD NOT RUN ON THE NEW CONTROLLER BOARD TO THE NEWLY FORMATTED DISK. WE REALIZE THAT THIS IS ALSO A REAL HASSEL TOO, BUT IT IS A NECESSARY EVIL. BESIDES, THE NEW FROMATTED DISK WILL STILL WORK WITH THE OLD CONTROLLER BOARD.

*** NOTE ***

IF ALL ELSE FAILS, EVEN AFTER READING THE DIRECTIONS, FEEL FREE TO CALL ME HERE AT TARBELL ELECTRONICS. AND IF YOU WOULD LIKE TO DISCUSS ANYTHING ABOUT THE BOARD OR SOFTWARE, CALL ME.

THANKS FOR INPUTS AND OUTPUTS ON THIS PRODUCT AND HOPE YOU WILL FIND THIS PRODUCT BOTH INFORMATIVE AND FUN TO WORK WITH.

GERALD.W.MULCHIN
ENGINEERING DEPT.
TARBELL ELECTRONICS
213 538 4251

4-21-81

TARBELL CP/M* 2.0 USER'S GUIDE

This guide was written to help users of TARBELL CP/M get up as quickly as possible, and to complement the CP/M manuals from Digital Research. It is best to read this guide before attempting to use CP/M, in order to avoid various pitfalls.

Table of Contents

* CP/M is a trademark/tradename of Digital Research, Inc.

9-12-79

Getting Your Tarbell CP/M System Up and Going

1. Be sure that you have successfully completed the Checkout section of the Tarbell Floppy Disk Interface Manual.
2. On the Floppy Disk Interface (FDI), set switches 1 to 5 off, (device address = F8-FC), switch 6 on (write-protected), and switch 7 on (bootstrap enabled).
3. Turn on your computer, then the disk drive and power supply.
4. Press Reset on the computer.
5. Insert the CP/M disk into the drive and close the door.
6. Press RUN on the computer (RESET if no front panel switches).
7. The disk drive head should engage, then step once. If it doesn't, go back to the Tarbell FDI manual, page 7-7.
8. If the CP/M opening message comes up on your console device, answer the question with a 1, 2, 3, or 4 on your keyboard. CP/M should then type: A> which means it is ready for a valid CP/M command or transient program name to be typed in.
9. If the above all goes well, type "DIR" . CP/M should then list all 20 or more file names that are on the disk. Look for a file ending in "ASM" , then type "TYPE filename.ASM" . The content of the assembly language file that you selected should be typed by CP/M on your console. Now go on to the next page in this guide: "Making Backups of your CP/M Disk".
10. If at some point in steps 8 or 9 above the system does not perform as shown, the interface to your console device may be set up with different port numbers or status bits than that which the Tarbell Basic Input/Output System (BIOS) expects. This is port 0 for status, port 1 for data, bit 0 of status low means keyboard ready, bit 7 of status low means printer ready. You may at this point decide whether to change your console interface to match BIOS, or change BIOS to match your interface.
11. You may temporarily patch BIOS for your console by following the procedure outlined below.
12. Stop your computer, and look at the listing of BIOS (FBIOS24 or CBIOS24) that came with your CP/M disk. Find the routines marked CONST, CONIN, and CONOT. Compare these routines with the ones normally used with your console interface. They may need changing.
13. Before making the changes, first disable the bootstrap by doing an EXAMINE from the front panel with the address switch 5 up, or turning dipswitch 7 on the FDI board temporarily off.
14. Change the console routines for your interface by depositing the correct I/O port numbers, status mask, and conditional jump instructions into the locations occupied by the original ones. It is also possible to deposit jumps into your ROM routines, as long as you make sure the right registers are used.
15. Turn dipswitch 7 back on (if your turned it off), examine at zero and run, or jump to location zero (do not reset).
16. Now try steps 8 and 9 again. When completed, go on to the next page in this guide: "Making Backups of your CP/M Disk".

B>

2

Using CP/M V1.4 to modify CP/M V2.0

1. Make a copy of your master disk using CP/M 1.4 and the "copy" program. Put your master disk away for safe keeping.

2. Copy onto a blank disk the "cbios" and "boot" files for Ver. 2.0

3. Next, invoke the sysgen program and get the 2.0 system from the first 2 tracks into memory.
   e.g. A:SYSGEN<cr>
   put your CP/M v2.0 disk into drive A and type the letter A in answer to the first question, and then type <cr>. Now return your CP/M V1.4 disk back to the A drive and type <cr> to the second question, as you will need to return back to the operating system.

4. Type save 34 cpm24.com<cr>

5. You now have a copy of your operating on disk and are now ready to modify your bios and boot files for your system. Do not change the memory size at this time, as the system you copied from your disk is set up for 24K and you have no way to make another size until after you get your new CP/M V2.0 running. If you try to make a new system size while using your CP/M V1.4, you will get a SYNC error.

6. After assembling both the bios and boot files, you are ready to build your new system disk with your changed bios. Be sure that your copy of the system you just saved (cpm24.com) is on the same disk as your bios and boot files.

7. At this point, the steps needed to finish are on page 8 starting with step 4. Use the relocation values for 24 k as shown in the table.

B>

Making Backups of your CP/M System Disk

IMPORTANT! BE SURE TO FOLLOW ONE OF THESE BACKUP PROCEDURES BEFORE PROCEEDING TO USE CP/M FOR ANYTHING ELSE. TO FAIL TO DO SO MAY CAUSE THE LOSS OF YOUR OPERATING SYSTEM SOFTWARE.

A disk system is highly vulnerable to a phenomenon called a "crash". A crash can occur when the CPU starts executing instructions in memory locations that are not in the proper sequence normally executed by the running program. This may be caused by hardware failure, powerline surges, operator errors, and the execution of programs which have not been completely debugged. It is possible during a crash sequence, and likely, sooner or later, for the CPU to execute instructions which cause invalid data to be written onto the disk, such that parts of or all the programs on the disk will be irrecoverable (a crashed disk).

The main weapon you have against crashing disks, other than clean hardware and operating proceedures, is to create frequent backups. The first thing you want to do is create a backup of the CP/M system disk. There are two good ways to do this outlined below.

If you have two disk drives, or a dual drive, it is relativly easy to back up your disk. Once you're up and running on disk A, put a fresh disk in disk B, then type:
"COPY ALL" This will give you a message; then press return, and whatever is on disk A will be copied onto disk B track by track. Be sure to mark the new disk with the copyright notice according to your license agreement.

Useful CP/M Operating System Notes

These are items of interest that may not be immediately obvious, but are nevertheless very useful to know.

1. After pressing reset and run with a CP/M diskette in the drive, you should get a message stating the version of CP/M you have. If it asks you, type in the number of diskettes which you will be using. For a dual drive or for two drives, type 2. If it comes back with a "A>", you're ready to start entering commands. If you have not yet made a backup, do so now, referring to page 2 of this guide.

2. You will find that the follwoing control characters are useful:
   - CTL-C Terminates output, escapes to monitor. If done after a prompt, may cause a warm boot to execute.
   - CTL-S Freezes printing temporarily. Any other character will resume printing after the CTL-S is used.
   - CTL-Z Terminates input from keyboard to editor or PIP.
   - CTL-U Cancels the currentl line being entered.
   - RUBOUT Prints and cancels the last character entered.
   - CTL-P Echos all console output onto the LIST device. Another CTL-P deactivates this feature.
   - CTL-H or BACKSPACE Deletes last character typed from console.

3. When the hard bootstrap is activated by reset and run, the 128-byte (one sector) cold-start loader is read in at zero an executed (source is xxBOOTxx.ASM). It then reads in the 51 sectors of CP/M into the top end of RAM. A "Warm-Boot" routine in BIOS can be activated by jumping to location zero. This reads in all of CP/M except the BIOS. This may be used to exit from programs which overlay all of CP/M except BIOS.

4. The CP/M system that is brought in from disk A with the bootstrap stays in memory until another warm boot is executed, perhaps by leaving from the Editor, or other system program. It is important to realize that when a warm boot is performed, the memory size of the system being booted in must match the memory size of the old one, since BIOS will load CP/M at the previous location. In order to change to a different memory size, you must do a cold boot.

5. The CPM.COM file on the disk is a program which may be used to generate CP/M systems for different memory sizes. This is not the CP/M system itself, which is brought in from the first two tracks.

6. The BIOS and BOOT contained in the CPM.COM file are set up for the INTEL MDS system, and will not work with the Tarbell Floppy Disk Interface. This is why it is important to follow the procedure in this guide on pages 8,9, and 10, when you wish to make changes in your Tarbell CP/M system.

7. If you use PIP to transfer .INT (BASIC intermediate) files, be sure to use the O parameter as described in the CPM manual under the PIP program description.

8. Don't use the file extention (like .ASM) when running the ASM, BASIC, RUN, and LOAD programs.

2CBIOS24.ASM - Standard input/output system for Tarbell CP/M and Tarbell Floppy Disk Interface (FDI). Notice that there is now a VDM-1 driver in it, which may be activated by setting the proper EQU's.

2SBOOT24.ASM - Source for the standard Tarbell FDI coldstart loader program, set up for 24k.

COPY.COM, COPY.ASM - Command and source files for a copy program from the CP/M user's group, which has been modified to work with the Tarbell FDI. To use it, type "COPY ALL", "COPY SYSTEM", or "COPY DATA". After typing in the command, it will come back and wait for a carriage-return so you can put your master and fresh disks in. The first form copies the whole disk, track by track, from A to B. The second form copies just the first two tracks, and the third copies all but the first two tracks. After it comes back with the ending message, a carriage-return will do a warm boot, and an ampersand (&) followed by a carriage-return will repeat the operation.

FORMAT.ASM - This program, written by Dick Culbertson, allows you to completely format a blank or crashed disk. It wipes out whatever is on the disk, so be sure you have all the information off the disk that you need. Also be sure to remove the disk you have in drive A and only have the one you want reformatted in there, when it comes up with it's opening message. This program also needs assembling before use.

2FBIOS24.ASM - This is the assembly-language source for the dual PerSci version of the Basic Input/Output System. It is usually the same as the standard version, except for the DUAL and FAST EQU statements. Notice that it also contains a VDM driver. Follow the procedure in the CP/M operating system notes entitled "Making Changes in the CP/M System" in order to integrate this module into your system.

2FBOOT24.ASM - This is the assembly-language source for the dual PerSci version of the coldstart loader program.

BASIC.COM, RUN.COM - These are the command modules for the BASIC-E compiler and run-time interpreter, written by Gordon Eubanks, Jr.

EXAMPLE OF HOW TO CREATE, COMPILE, AND RUN A BASIC PROGRAM.

```
A>DIR
A: PIP      CØM
A: ED       CØM
A: ASM      CØM
A: BASIC    CØM
A: RUN      CØM
A: SYSGEN   CØM
A: CPM      CØM
A: SUBMIT   CØM
A: DDT      CØM
A: STAT     CØM
A: LØAD     CØM
A: DUMP     CØM
A: SBØØT24  ASM
A: CASDSK   ASM
A: DSKCAS   ASM
A: CBIØS    ASM
```

THE DIRECTORY

```
A>ED TEST-B.BAS
```

USE FOR ALL BASIC PROGRAMS.
NAME OF PROGRAM.

```
NEW FILE
*I
10 FØR N=1 TØ 5
20 PRINT N,SQR(N)
30 NEXT N
40 END
*Ø^Z
*BØT
10 FØR N=1 TØ 5
20 PRINT N,SQR(N)
30 NEXT N
40 END
*E
```

*I ← Insert command to EDITOR.

YOU TYPE THIS IN.

PRESS ctl-Z to get out of entry mode.

MOVE POINTER TO BEGINNING OF BUFFER, TYPE ALL THE LINES IN THE BUFFER.

EXIT FROM EDITOR AND SAVE THE FILE.

```
A>BASIC TEST-B
BASIC-E COMPILER  VER 2.0
   1*  10 FØR N=1 TØ 5
   2*  20 PRINT N,SQR(N)
   3*  30 NEXT N
   4*  40 END
   Ø ERRØRS DETECTED
```

RUN THE BASIC COMPILER ON YOUR FILE.

THE COMPUTER PRINTS THIS.

```
A>RUN TEST-B
BASIC-E INTERPRETER: VER 2.0

1             1
2             1.414213
3             1.732051
4             2
5             2.236068
```

RUN YOUR PROGRAM.

YOUR PROGRAM PRINTS THIS.

EXAMPLE OF HOW TO EDIT (CHANGE), COMPILE, AND RUN A BASIC PROGRAM.

```
^C          <-- NAME OF FILE YOU CREATED PREVIOUSLY.
A>ED TEST-B.BAS
*#A         <-- APPEND ALL THE LINES INTO THE BUFFER THAT WILL FIT.
*F20        <-- FIND THE STRING "20".
*T          <-- PRINT THE REST OF THE LINE.
 PRINT N,SQR(N)
*SSQR(N)^ZN*N   <-- SUBSTITUTE N*N FOR SQR(N).
*0LT        <-- GO BACK TO BEGINNING OF LINE AND TYPE IT.
20 PRINT N,N*N
*B#T        <-- GO BACK TO BEGINNING OF BUFFER AND TYPE IT.
10 FØR N=1 TØ 5
20 PRINT N,N*N
30 NEXT N
40 END
*I          <-- WITH POINTER STILL AT BEGINNING, INSERT THIS LINE.
5  PRINT "START"
*4LT        <-- MOVE AHEAD 4 LINES AND PRINT A LINE.
*-LT        <-- TOO FAR, GO BACK ONE LINE AND PRINT IT.
40 END
*I          <-- INSERT THIS LINE.
   PRINT "STØP"
*B#T        <-- PRINT THE WHOLE BUFFER AGAIN.
5  PRINT "START"
10 FØR N=1 TØ 5
20 PRINT N,N*N
30 NEXT N
   PRINT "STØP"
40 END
*E          <-- EXIT FROM EDITOR AND SAVE THE FILE.

A>BASIC TEST-B   <-- COMPILE THE NEW VERSION.
BASIC-E CØMPILER  VER 2.0
   1*  5  PRINT "START"
   2*  10 FØR N=1 TØ 5
   3*  20 PRINT N,N*N
   4*  30 NEXT N
   5:     PRINT "STØP"
   6*  40 END
   0 ERRØRS DETECTED
                    } COMPUTER PRINTS THIS.
A>RUN TEST-B     <-- RUN THE PROGRAM.
BASIC-E INTERPRETER: VER 2.0

START
1                1
2                4
3                9
4                16
5                25
STØP
                 } YOUR PROGRAM NOW PRINTS THIS.
```

SAMPLE CREATE, ASSEMBLE, AND RUN ASSEMBLY LANGUAGE PROGRAM

```
^C
A>ED TEST-A.ASM
NEW FILE
*I
; THIS IS A TEST ASSEMBLY-LANGUAGE PRØGRAM
; WHICH PRINTS TEXT ØN THE CØNSØLE BY
; USING THE ØPERATING SYSTEM PRINT RØUTINES.
;
FDØS    EQU  5          ;ENTRY PØINT TØ SYSTEM.
        ØRG  100H       ;ALL PRØGRAMS START HERE.
START:  MVI  C,9        ;PRINT CØNSØLE BUFFER.
        LXI  D,MESAG    ;GET ADDRESS ØF MESSAGE.
        CALL FDØS       ;NØW DØ THE ØPERATIØN.
        JMP  0          ;DØ A WARM BØØT.
MESAG:  DB   ODH,OAH,'THIS IS A TEST','$'
        END
*E
A>ASM TEST-A
CP/M ASSEMBLER - VER  .0
011C
000H USE FACTØR
END ØF ASSEMBLY
A>TYPE TEST-A.PRN
```

NAME OF NEW FILE

FOR ALL ASSEMBLY-LANGUAGE PROGRAMS.

*I ← Insert command to EDITOR.

COMMENTS

LABELS

YOU TYPE ALL THIS IN.

ASSEMBLY LANGUAGE CODE.

*E ← EXIT FROM EDITOR, SAVE PROGRAM.

A>ASM TEST-A ← ASSEMBLE PROGRAM. (CONVERT TO MACHINE CODE)

011C ← NEXT ADDRESS AVAILABLE.

000H USE FACTØR ← PERCENTAGE OF SYMBOL TABLE USED.

A>TYPE TEST-A.PRN ← TYPE THE LISTING FILE THAT THE ASSEMBLER MADE.

ADDRESSES

EQU VALUE

MACHINE CODE

```
                ; THIS IS A TEST ASSEMBLY-LANGUAGE PRØGRAM
                ; WHICH PRINTS TEXT ØN THE CØNSØLE BY
                ; USING THE ØPERATING SYSTEM PRINT RØUTINES.
                ;
0005 =          FDØS    EQU  5          ;ENTRY PØINT TØ SYSTEM.
0100                    ØRG  100H       ;ALL PRØGRAMS START HERE.
0100 0E09       START:  MVI  C,9        ;PRINT CØNSØLE BUFFER.
0102 110B01             LXI  D,MESAG    ;GET ADDRESS ØF MESSAGE.
0105 CD0500             CALL FDØS       ;NØW DØ THE ØPERATIØN.
0108 C30000             JMP  0          ;DØ A WARM BØØT.
010B 0D0A544849MESAG:   DB   ODH,OAH,'THIS IS A TEST','$'
011C                    END
```

```
A>LØAD TEST-A
FIRST ADDRESS 0100
LAST  ADDRESS 011B
BYTES READ    001C
RECØRDS WRITTEN 01
```

← CONVERT TEST-A.HEX FILE TO TEST-A.COM FILE.

```
A>TEST-A
THIS IS A TEST
A>
```

← RUN THE PROGRAM.

← THE PROGRAM PRINTS THIS.

```
A>B:
B>PIP A:LØAD.CØM=B:LØAD.CØM
```

DESCRIPTION OF EQU'S IN CBIOS

The EQU's which are at the beginning of CBIOS are used to control the assembly of the CBIOS file at assembly time.

Careful examination of the CBIOS listing provided you will give you a better insight into it's function.

You will notice that there is a line called MSIZE, with an EQU after it and then a number, e.g. 24. What this means is that during assembly time, the assembler will store the value 24 in a location called MSIZE, and every time the assembler encounters the word MSIZE in the file, it will use the value of 24 in it's place.

The rest of the equates control such things as I/O, type of drives, data ports, etc...

In our STANDARD version, the STD EQU is set to TRUE. The rest of the Equates are False. In this way, we control the assembling of this file so that it will be configured for whats called Standard I/O, (status on 0, data on 1).

At any point in time, you can totally reconfigure your system to any I/O setup you wish just by changing the equates in the CBIOS file to what you will be using.
It must be understood that if you are going to make this kind of change, you must do them before you actually change the hardware. Otherwise, your new version of CP/M and CBIOS will not work with your new system. Also, if you can check out the new operating system on someone elses system that matches the type you are changing to, you will know that the software is is working or not.

The Assembler book provided by Digital Research is a good tutorial on assembly language techniques and it is recomended that you at least skim over it and know what some of the features are.
This will help you understand what is happening when you make changes in your CBIOS file and will also give you a little insight into how a computer language functions.

Making Changes in the CP/M Operating System

This sheet outlines the procedure for changing the CP/M system for different sizes of memory or for different I/O configurations.

You should have the following files on your disk in order to make the changes by the method outlined. The little x's denote characters which are different on different CP/M systems.
xBOOTxx.ASM, CPM.COM (sometimes called MOVCPM.COM), DDT.COM, SYSGEN.COM, ED.COM,and xxBIOSxx.ASM.

1. Using the Editor, change xBIOSxx.ASM and xBOOTxx.ASM to the desired configuration. Listings of these routines are provided. To change the memory size, just change the statement MSIZE EQU 24 to whatever size you would like. Note that in BIOS, there are a bunch of EQU's at the beginning that define the I/O setup. If the driver for your console is contained in the system, you may be able to just change the STD EQU to FALSE and yours to TRUE. Also check the port numbers CSTAT, CCOM, and CDATA to see if they match your console interface. The editor's S command is very useful to make these changes, and the N command is good for finding the place.

2. Assemble BIOS and BOOT by typing ASM xBIOSxx and then ASM xBOOTxx . Then type CPM xx * (remember the asterisk), where xx is the memory size you want . For a 32k system, type CPM 32 * .

3. Type SAVE 34 CPMxx.COM just like it says. Note: the 34 here is always 34. If you get a "Syncronization Error", that indicates that the serial number imbedded in your CPM.COM file doesn't match the serial number in the system which is running in memory. Sometimes you can alleviate this problem by first getting the size system in memory that matches that of the system on the disk with the CPM.COM module on it, then put that disk in, and do a contorl-C.

4. Type in DDT CPMxx.COM. This reads the relocated CP/M sytem into memory under DDT, down low in memory. DDT will now be used to overlay the BIOS and BOOT modules onto the system.

5. Type IxBIOSxx.HEX, return, then Rbias, return, where bias is given by the table below:

xx	bias	xx	bias	xx	bias	xx	bias
20k	D580	24K	C580	28K	B580	32K	A580
36K	9580	40K	8580	44K	7580	48K	6580
52K	5580	56K	4580	60K	3580	64K	2580

** NOTE **
CPM 2.0 allows relocation on 1k boundrys. Refer to CPM 2.0 manual.

6. Type in IxBOOTxx.HEX, then R900, then Ctl-C, then SYSGEN.

7. For "SOURCE DRIVE NAME", type return. For "DESTINATION DRIVE NAME", tell it which drive to put the new system onto.

8. Now mount the fresh new disk in that drive and press return. The new system will now be written onto this disk.

NOTE: Do not press carriage-return to reboot.

SEE THE EXAMPLE OF THIS PROCEDURE WHICH FOLLOWS.

11

EXAMPLE OF HOW TO CHANGE YOUR OPERATING SYSTEM FOR A DIFFERENT MEMORY SIZE AND I/O.

```
A:ED 2FBIOS24.ASM          <- EDIT YOUR SOURCE FILE
     : *NMSIZE             <- SEARCH FOR MSIZE STATGMENT
   26: *0LT                <- TYPE CURRENT LINE
   26:  MSIZE   EQU  (24)  <- YOU WANT 32K
   26: *S24^Z32^Z0LT       <- SUBSITUTE 32 FOR 24 AND TYPE IT OUT
   26:  MSIZE   EQU  32          ;MEMORY SIZE IN KBYTES.
   26: *-1
   29:  STD     EQU (TRUE)       ;TRUE IF STANDARD I/O.
   29: *STRUE^ZFALSE^Z0LT  <- WE DON'T WANT STD I/O, SO CHANGE IT TO FALSE
   29:  STD     EQU  FALSE       ;TRUE IF STANDARD I/O.
   29: *                   <- WE WANT MITS I/O
   30:  MSIO2   EQU (FALSE)      ;TRUE IF MITS 2SIO.
   30: *SFALSE^ZTRUE^Z0LT  <- SET MSIO2 TO TRUE AND TYPE IT AGAIN
   30:  MSIO2   EQU  TRUE        ;TRUE IF MITS 2SIO.
   30: *                   <- THERE ARE A BUNCH OF EQU'S IN FILE.
   31:  DELTA   EQU (FALSE)      ;TRUE IF USING DELTA PRODUCTS CPU.
   31: *E                  <- WE DON'T WANT ANYMORE CHANGES, SO EXIT
                              THE EDITOR.
A:ED 2FBOOT24.ASM          <- EDIT BOOT FILE ALSO
     : *NMSIZE             <- SEARCH FOR MSIZE STATGMENT
   18: *0LT                <- TYPE THE LINE.
   18:  MSIZE   EQU (24)         ;MEMORY SIZE IN DECIMAL KB.     **
                           <- WE WANT TO CHANGE THIS
   18: *S24^Z32^Z0LT       <- SUBSITUTE 32 FOR 24 AND TYPE IT AGAIN
   18:  MSIZE   EQU  32          ;MEMORY SIZE IN DECIMAL KB.     **
   18: *E                  <- NO MORE CHANGES, SO EXIT THE EDITOR.

REN 2FBIOS(32).ASM=2FBIOS24.ASM   } RENAME BOTH FILES
B>REN 2FBOOT(32).ASM=2FBOOT24.ASM } TO SHOW MEMORY SIZE THATS USED
B>A:ASM 2FBIOS32           <- NOW ASSEMBLE THE FILE
CP/M ASSEMBLER - VER 2.0
7E7D
007H USE FACTOR
END OF ASSEMBLY

B>A:ASM 2FBOOT32           <- NOW ASSEMBLE THE FILE
CP/M ASSEMBLER - VER 2.0
0080
001H USE FACTOR
END OF ASSEMBLY

B>CPM(32)*                 <- CREATE CPM OPERATING IMIAGE
                           <- MEMORY SIZE DESIRED
CONSTRUCTING 32k CP/M vers 2.0
READY FOR "SYSGEN" OR
"SAVE 34 CPM32.COM"        <- TYPE THIS JUST AS IT APPEARS IN QUOTES!
B>SAVE 34 CPM32.COM        <- JUST LIKE ITS DONE HERE.
B>A:DDT CPM32.COM          <- BRING IN THE SYSTEM USING DDT
DDT VERS 2.0
NEXT  PC
2300 0100
-I2FBIOS32.HEX             <- INPUT YOUR BIOS FILE USING THE 'I' COMMAND.
-RA580                     <- NOW DO ACTUAL READ OF THE FILE USING A BIAS VALUE
NEXT  PC                      FROM TABLE.
2300 0000
-I2FBOOT32.HEX             <- INPUT YOUR BOOT FILE USING THE 'I' COMMAND.
-R900                      <- READ THE FILE WITH A BIAS OF 900.
NEXT  PC
2300 0000
-^C                        <- EXIT BACK TO SYSTEM LEVEL USING CNTL-C.
```

12

```
B>A:
A>SYSGEN
SYSGEN VER 2.0
SOURCE DRIVE NAME (OR RETURN TO SKIP)
DESTINATION DRIVE NAME (OR RETURN TO REBOOT) B
DESTINATION ON B, THEN TYPE RETURN
FUNCTION COMPLETE
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)

A>
```

GET THE SYSGEN PROGRAM

<cr> ← TYPE CARRIAGE RETURN

B ← THIS IS THE DRIVE WE WILL PUT SYSTEM ON.

<cr> ← TYPE CARRIAGE RETURN

SYSTEM IS NOW ON THE NEW DISK

DO NOT TYPE <CR>, BUT PUT NEW SYSTEM INTO DRIVE A AND RESET & RUN TO CHECK IT OUT.

```
                ; CP/M BASIC INPUT/OUTPUT OPERATING SYSTEM (BIOS)
                ; TARBELL ELECTRONICS CPM 1.4 VERSION OF 7-14-80.
                ; Copyright (c) 1980 Tarbell Electronics.
                ;
                ; THIS MODULE CONTAINS ALL THE INPUT/OUTPUT ROUTINES FOR
                ; THE CP/M SYSTEM, INCLUDING THE DISK ROUTINES FOR AUTOMATIC
                ; SELECTION OF THE DISK DENSITY AND THE NECESSARY DMA ROUTINES.
                ; THIS SECTION ALSO DEFINES THE I/O PORTS AND STATUS BITS. BY
                ; SETTING THE PROPER VALUES FOR THE EQU STATEMENTS, THE I/O MAY BE
                ; AUTOMATICALLY RECONFIGURED TO FIT MOST SITUATIONS. THE TRUE AND
                ; FALSE ONES CONTROL CONDITIONAL ASSEMBLIES OF DIFFERENT SECTIONS
                ; OF I/O ROUTINES TO FIT DIFFERENT INTERFACE REQUIREMENTS.

FFFF =          TRUE    EQU  0FFFFH     ;DEFINE VALUE OF TRUE.
0000 =          FALSE   EQU  NOT TRUE   ;DEFINE VALUE OF FALSE.

                ************************************************
                *** THIS BEGINS THE AREA WHICH REQUIRES CHANGES ***
                ***      FOR DIFFERENT CONSOLE I/O SYSTEMS      ***
                ************************************************

0018 =          MSIZE   EQU  24         ;MEMORY SIZE IN KBYTES.
0000 =          INTRP   EQU  FALSE      ;TRUE IF INTERRUPTS ALLOWED.
0000 =          TESTING EQU  FALSE      ;TRUE FOR TESTING ERRORS
0000 =          TARBELL EQU  FALSE      ;TRUE IF USING THE TARBELL Z-80 CPU.
0000 =          IOBASE  EQU  0          ;BASE IO ADDR FOR TARBELL CPU (0 or 10 hex).
0000 =          TIMER   EQU  FALSE      ;TRUE IF USING CPU TIMER (Tarbell CPU board).
FFFF =          STD     EQU  TRUE       ;TRUE IF STANDARD I/O.
0000 =          VDB8024 EQU  FALSE      ;TRUE IF USING SD SALES VDB-8024.
0000 =          DELTA   EQU  FALSE      ;TRUE IF USING DELTA CPU CARD.
0000 =          MSIO2   EQU  FALSE      ;TRUE IF MITS 2SIO.
0000 =          ISIO2   EQU  FALSE      ;TRUE IF IMSAI SIO-2.
0000 =          TUART   EQU  FALSE      ;TRUE IF CROMEMCO TUART.
0000 =          VDM     EQU  FALSE      ;TRUE IF PROC TECH VDM.
0000 =          FLASH   EQU  FALSE      ;TRUE IF VG FLASHWRITER.
0000 =          VB1     EQU  FALSE      ;TRUE IF SSM VB1-B.
0000 =          OTHER   EQU  FALSE      ;TRUE IF SOMETHING ELSE.
0000 =          SOLOS   EQU  FALSE      ;TRUE IF PROC TECH SOLOS.
FFFF =          BACKSP  EQU  TRUE       ;AUTO-BACKSPACE FOR CRT'S.
0000 =          DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED DRIVES.(1 LOGICAL DRIVE)
0000 =          DMACNTL EQU  FALSE      ;TRUE IF DMA CONTROLLER
0000 =          SPOOL   EQU  FALSE      ;TRUE IF USING KLH SPOOLER
                ;
                        IF NOT SOLOS AND NOT TARDEL     ;IF NOT PROC TECH SOLOS,
0000 =          CSTAT   EQU  0          ;CONSOLE STATUS PORT.
0000 =          CCOM    EQU  0          ;CONSOLE COMMAND PORT.
0001 =          CDATA   EQU  1          ;CONSOLE DATA PORT.
0002 =          LSTAT   EQU  2          ;LIST STATUS PORT.
0002 =          LCOM    EQU  2          ;LIST COMMAND PORT.
0003 =          LDATA   EQU  3          ;LIST DATA PORT.
                        ENDIF
                ;
0000 =          CONUL   EQU  FALSE      ;CONSOLE NULLS?
0010 =          CNULL   EQU  16         ;CONSOLE NULL COUNT.
0000 =          LSTNUL  EQU  FALSE      ;LIST DEVICE NULLS?
0000 =          LNULL   EQU  0          ;LIST NULL COUNT.
0000 =          LSTPAG  EQU  FALSE      ;LIST DEVICE PAGING?
0042 =          LINCNT  EQU  66         ;LINES PER PAGE.
0000 =          HLAB    EQU  0          ;8 FOR HD LD AT BEG OF SEEK.
0002 =          STPRAT  EQU  2          ;RATE 0=3ms,1=6MS, 2=10MS, 3=20MS.
0000 =          DUAL    EQU  FALSE      ;TRUE IF DUAL DRIVE.(2 HEADS MOVE TOGETHER)

                ************************************************
                *** THIS IS THE END OF THE AREA WHICH NORMALLY NEED ***
                ***     BE CHANGED FOR MOST CONSOLE I/O SYSTEMS     ***
                ************************************************

0000 =          VIDEO   EQU  VDM OR FLASH OR VB1    ;TRUE FOR ANY VIDEO.
FFFF =          RDYLO   EQU  STD OR SOLOS OR OTHER  ;STATUS READY WHEN LOW.
0000 =          RDYHI   EQU  NOT RDYLO
0000 =          TARDEL  EQU  TARBELL OR DELTA   ;IF USING TARBELL OR DELTA CPU.
                ;
                        IF   TARDEL     ;IF USING TARBELL OR DELTA CPU
                CCOM    EQU  IOBASE+1   ;CONSOLE COMMAND PORT
                CSTAT   EQU  IOBASE+1   ;CONSOLE STATUS PORT ( CHAN A.)
                CDATA   EQU  IOBASE+0   ;CONSOLE DATA PORT
                LCOM    EQU  IOBASE+3   ;LIST COMMAND PORT
                LSTAT   EQU  IOBASE+3   ;LIST STATUS PORT (CHAN B.)
```

```
                LDATA   EQU  IOBASE+2   ;LIST DATA PORT
                        ENDIF
                ;
                        IF      TIMER AND TARBELL       ;MUST BE USING TARBELL CPU.
                ;
                ; TIMER EQUATES
                ;
                TCH0    EQU     IOBASE+4        ;TIMER CHAN 0 ADDRESS
                TCH1    EQU     IOBASE+5        ;TIMER CHAN 1 ADDRESS
                TCH2    EQU     IOBASE+6        ;TIMER CHAN 2 ADDRESS
                TCMND   EQU     IOBASE+7        ;TIMER COMMAND PORT
                IMASK   EQU     IOBASE+8        ;INTERRUPT MASKING PORT
                CNTR0   EQU     00000000B       ;counter 0
                CNTR1   EQU     01000000B       ;counter 1
                CNTR2   EQU     10000000B       ;counter 2
                RLWORD  EQU     00110000B       ;read/load lsb 1st, msb 2nd.
                RLHBYTE EQU     00100000B       ;read/load msb only.
                RLLBYTE EQU     00010000B       ;read/load lsb only.
                CNTRLT  EQU     00000000B       ;counter latching operation.
                BINARY  EQU     00000000B       ;select binary operation.
                BCD     EQU     00000001B       ;select BCD operation.
                MODE0   EQU     00000000B       ;interrupt on terminal count.
                MODE1   EQU     00000010B       ;programmable One-shot.
                MODE2   EQU     00000100B       ;rate generator.
                MODE3   EQU     00000110B       ;square wave rate generator.
                MODE4   EQU     00001000B       ;software triggered strobe.
                MODE5   EQU     00001010B       ;hardware triggered strobe.
                        ENDIF

                        IF   VDM                ;IF PROC TECH VDM1,
                SCREEN  EQU  0CC00H             ;SCREEN PLACE IN MEMORY.
                        ENDIF

                        IF  FLASH OR VB1
                SCREEN  EQU  0F800H             ;SCREEN PLACE IS DIFFERENT.
                        ENDIF

                        IF  FLASH               ;IF VG FLASHWRITTER
                ENDSCR  EQU (SCREEN+2048) SHR 8
                        ENDIF

                        IF  VB1 OR VDM                  ;IF SSM VB1-B BOARD
                ENDSCR  EQU (SCREEN+1024) SHR 8
                        ENDIF
                ;
                        IF  VIDEO
                LINES15 EQU SCREEN + 64         ;VIDEO LINES
                SCRLCNT EQU 960                 ;LINES TO SCROLL
                SCRNTOP EQU SCREEN + SCRLCNT    ;TOP OF SCREEN
                        ENDIF
                ;
0008 =          BKSP    EQU 08H                 ;BACKSPACE EQUATE
000A =          LF      EQU 0AH                 ;LINEFEED EQUATE
000D =          CR      EQU 0DH                 ;CARRIAGE RET EQUATE
000C =          FF      EQU 0CH                 ;FORM-FEED EQUATE.
                ;
0008 =          DDEN    EQU  08H        ;DOUBLE DENSITY ENABLE CODE
00F7 =          DDDSB   EQU  0F7H       ;DOUBL. DENSITY DISABLE CODE
                ;
                        IF   SOLOS              ;IF PROC TECH SOLOS,
                CSTAT   EQU  0FAH               ;CONSOLE STATUS PORT.
                KBD     EQU  0C02EH             ;SOLOS KEYBOARD.
                CLRSCR  EQU  0C0D5H             ;CLEAR SCREEN.
                SCRN    EQU  0C054H             ;SOLOS OUTPUT.
                        ENDIF

                        IF NOT SOLOS            ;IF NOT PROC TECH SOLOS,
00E0 =          DMAP    EQU  0E0H               ;DMA BASE ADDRESS.
00F8 =          DISK    EQU  0F8H               ;DISK BASE ADDRESS.
                        ENDIF
                ;
                        IF SOLOS                ;IF PROC TECH SOLOS,
                DMAP    EQU  060H               ;DMA BASE ADDRESS.
                DISK    EQU  078H               ;DIFFERENT DISK PORTS.
                        ENDIF
                ;
00E0 =          ADR0    EQU  DMAP+0             ;DMA ADDRESS REG PORT.
00E1 =          WCT0    EQU  DMAP+1             ;DMA WORD COUNT REG PORT.
00E8 =          CMND    EQU  DMAP+8             ;DMA COMMAND PORT.
00F8 =          DCOM    EQU  DISK               ;DISK COMMAND PORT.
```

```
00F8 =          DSTAT   EQU  DISK               ;DISK STATUS PORT.
00F9 =          TRACK   EQU  DISK+1             ;DISK TRACK PORT.
00FA =          SECTP   EQU  DISK+2             ;DISK SECTOR PORT.
00FB =          DDATA   EQU  DISK+3             ;DISK DATA PORT.
00FC =          WAIT    EQU  DISK+4             ;DISK WAIT PORT.
00FC =          DCONT   EQU  DISK+4             ;DISK CONTROL PORT.
00FD =          DMACHK  EQU  DISK+5             ;DMA CHECK PORT.
000A =          RTCNT   EQU  10                 ;RETRY COUNT.

                        IF   STD                ;IF STANDARD I/O,
0001 =          CKBR    EQU  00000001B          ;KEYBOARD READY BIT.
0080 =          CPTR    EQU  10000000B          ;CONS OUTPUT RDY BIT.
                        ENDIF

                        IF   VDB8024
                CKBR    EQU  00000010B          ;KEYBOARD RDY BIT.
                CPTR    EQU  00000100B          ;CON RDY BIT.
                        ENDIF

                        IF   MSIO2              ;IF MITS 2SIO,
                CKBR    EQU  00000001B          ;KEYBOARD READY BIT.
                CPTR    EQU  00000010B          ;PRINT READY BIT.
                        ENDIF

                        IF   TARDEL OR ISIO2            ;IF IMSAI SIO-2,
                CKBR    EQU  00000010B          ;KEYBOARD READY BIT.
                CPTR    EQU  00000001B          ;PRINT READY BIT.
                        ENDIF

                        IF   TUART              ;IF CROMEMCO TUART,
                CKBR    EQU  01000000B          ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B          ;PRINT READY BIT.
                        ENDIF

                        IF   SOLOS              ;IF PROC TECH SOLOS,
                CKBR    EQU  00000001B          ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B          ;DUMMY EQU.
                        ENDIF

                        IF OTHER                ;IF SOMETHING ELSE,
                CKBR    EQU  00000010B          ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B          ;PRINTER READY BIT.
                        ENDIF

0080 =          LRBIT   EQU  CPTR               ;LISTER READY BIT.
0003 =          IOBYTE  EQU  3                  ;ADDRESS OF I/O BYTE.
1C00 =          CBASE   EQU  (MSIZE-17)*1024    ;BIAS FOR LARGER THAN 17K.
4500 =          CPMB    EQU  CBASE+2900H        ;START OF CPM.
4D06 =          BDOS    EQU  CBASE+3106H        ;START OF BDOS 1.4.
1500 =          CPML    EQU  1500H              ;LENGTH OF CPM SYSTEM-BIOS.
0019 =          NSECTS  EQU  25                 ;NUMBER OF SECTORS TO READ.
                ;
5572                    ORG  CPMB+1072H         ;CP/M PATCH.
5572 CDF44F             CALL CPMB+0AF4H
5575 000000             NOP!NOP!NOP
5578 79                 MOV  A,C
5579 21F859             LXI  H,CPMB+14F8H
                ;
5A00                    ORG  CPMB+1500H         ;START OF BIOS.
                ;
                ; I/O JUMP VECTOR
                ; THIS IS WHERE CPM CALLS WHENEVER IT NEEDS
                ; TO DO ANY INPUT/OUTPUT OPERATION.
                ; USER PROGRAMS MAY USE THESE ENTRY POINTS
                ; ALSO, BUT NOTE THAT THE LOCATION OF THIS
                ; VECTOR CHANGES WITH THE MEMORY SIZE.
                ;
5A00 C33B5A             JMP  BOOT               ;FROM COLD START LOADER.
5A03 C39B5A     WBOOTE: JMP  WBOOT              ;FROM WARM BOOT.
5A06 C3E05A             JMP  CONST              ;CHECK CONSOLE KB STATUS.
5A09 C3E95A             JMP  CONIN              ;READ CONSOLE CHARACTER.
5A0C C3FF5A             JMP  CONOT              ;WRITE CONSOLE CHARACTER.
5A0F C3AC5D             JMP  LIST               ;WRITE LISTING CHAR.
5A12 C3B75D             JMP  PUNCH              ;WRITE PUNCH CHAR.
5A15 C3B85D             JMP  READER             ;READ READER CHAR.
5A18 C3C35B             JMP  HOME               ;MOVE DISK TO TRACK ZERO.
5A1B C3255B             JMP  SELDSK             ;SELECT DISK DRIVE.
5A1E C3C95B             JMP  SETTRK             ;SEEK TO TRACK IN REG A.
5A21 C3455C             JMP  SETSEC             ;SET SECTOR NUMBER.
5A24 C34A5C             JMP  SETDMA             ;SET DISK STARTING ADR.
```

```
5A27 C3835C             JMP  READ               ;READ SELECTED SECTOR.
5A2A C3F85C             JMP  WRITE              ;WRITE SELECTED SECTOR.
                ;
                        IF   SPOOL              ;IF USING KLH SPOOLER
                        DB   0FFH               ;FLAG FOR SPOOLER.
                        DW   LTBSY              ;LISTER STATUS LOCATION
                        DW   LTBSY              ;FOR SPOOLER - -
                        DW   LTBSY              ;I DON'T KNOW WHY IT'S
                        DW   LTBSY              ;HERE 4 TIMES EITHER.
                        ENDIF
                ;
                ;THE FOLLOWING TABLE DEFINES CP/M AS EITHER SINGLE OR
                ;DOUBLE DENSITY AND IS CHANGED ON THE FLY WHEN A DISK
                ;IS SELECTED. THE ORDER OF THIS TABLE MUST BE AS SHOWN.
                ;
5A2D 004E1A3F03SDTAB:   DB   00H,4EH,26,63,3,7,242      ;SINGLE DENSITY TABLE

                        IF   DUBSID
                        DB   02H,4EH,26,63,4,15,242     ;SINGLE DEN. DOUB SIDED.
                        ENDIF

5A34 010C335F04         DB   01H,0CH,51,95,4,15,237     ;SINGLE SIDED, DOUB DEN.

                        IF   DUBSID
                        DB   03H,0CH,51,95,5,31,237     ;DOUB SIDED, DOUB DEN.
                        ENDIF
                ;
                ; BOOT
                ; THIS SECTION IS EXECUTED WHENEVER RESET AND RUN
                ; IS PUSHED, AFTER THE COLDSTART LOADER READS IN
                ; THE CPM SYSTEM.
                ;
5A3B 318000     BOOT:   LXI  SP,80H     ;SET STACK POINTER.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        EI              ;ENABLE THEM HERE.
                        ENDIF

                        IF   TIMER AND TARBELL  ;IF USING TARBELL CPU
                        MVI  A,CNTR0+RLWORD+MODE2+BINARY        ;INIT 8253
                        OUT  TCMND              ;SEND IT TO COMMAND PORT
                        LXI  B,33333            ;TIME CONSTANT FOR 60 HZ
                        MOV  A,C
                        OUT  TCH0               ;LS BYTE OF COUNT
                        MOV  A,B
                        OUT  TCH0               ;MS BYTE OF COUNT
                        ENDIF

                        IF   STD        ;IF STANDARD I/O,
5A3E 00000000           NOP!NOP!NOP!NOP ;LEAVE SPACE FOR INIT.
5A42 00000000           NOP!NOP!NOP!NOP
5A46 00000000           NOP!NOP!NOP!NOP
5A4A 00000000           NOP!NOP!NOP!NOP
                        ENDIF

                        IF   MSIO2      ;IF MITS 2SIO,
                        MVI  A,3        ;INITIALIZE 2SIO.
                        OUT  CCOM
                        OUT  LCOM
                        MVI  A,11H
                        OUT  CCOM
                        OUT  LCOM
                        ENDIF

                        IF   TARDEL OR ISIO2
                        LXI  H,IOINIT           ;point to 8251 init. bytes
                        MVI  B,4                ;there are 4 of them
                INITIO: MOV  A,M                ;get a byte
                        OUT  CCOM               ;out to command port of console
                        OUT  LCOM               ;out to command port of lister
                        INX  H                  ;bump pointer
                        DCR  B                  ;decrease count
                        JNZ  INITIO             ;loop till done.
                        ENDIF

                        IF   TUART      ;IF CROMEMCO TUART,
                        MVI  A,1        ;SET A = 1.
                        OUT  54H        ;SELECT DEVICE A.
                        OUT  52H        ;RESET DEVICE B.
                        LXI  H,BAUDRS   ;GET ADR OF BAUD RATE TABLE.
```

```
                        MVI  A,11H       ;OCTUPLE THE CLOCK.
                ITl:    OUT  02H        ;& RESET CURRENT DEV.
                        MOV  A,M        ;GET BAUD RATE FROM TABLE.
                        INX  H          ;INCREMENT POINTER.
                        OUT  0          ;SET BAUD RATE.
                        CALL CONIN      ;READ KEYBOARD.
                        CALL CONIN      ;READ KEYBOARD AGAIN.
                        CPI  0DH        ;IF NOT CARRIAGE-RETURN,
                        MVI  A,1        ;SLOW THE CLOCK.
                        JNZ  ITl        ;UNTIL A CARRIAGE-RETURN.
                        ENDIF

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                        CALL CLRSCR     ;CLEAR SCREEN.
                        ENDIF

5A4E AF                 XRA  A          ;CLEAR SCRATCH AREA.
                        IF   VDM        ;IF PROC TECH VDM,
                        OUT  0C8H       ;CLEAR SCROLL PORT
                        ENDIF
5A4F 32CC5D             STA  LATCH      ;LATCH = 0
5A52 320300             STA  IOBYTE     ;CLEAR I/O BYTE.
5A55 0E0C               MVI  C,ENDZ-STARTZ    ;GET LENGTH OF ZERO AREA.
5A57 21BD5D             LXI  H,STARTZ   ;GET SCRATCH ADDRESS.
5A5A 77         BOOTL:  MOV  M,A        ;PUT ZERO IN MEMORY.
5A5B 23                 INX  H          ;INCREMENT POINTER.
5A5C 0D                 DCR  C          ;DECREMENT COUNTER.
5A5D C25A5A             JNZ  BOOTL      ;LOOP TILL DONE.

                        IF   VIDEO      ;IF ANY VIDEO BOARD,
                        CALL CLEAR      ;CLEAR SCREEN.
                        ENDIF

5A60 DB01               IN   CDATA      ;CLEAR CONSOLE STATUS.
5A62 21695D             LXI  H,SMSG     ;PRINT OPENING MESSAGE.
5A65 CD275D             CALL PMSG
5A68 CDE95A             CALL CONIN      ;READ # OF DISKS.
5A6B 4F                 MOV  C,A        ;ECHO THE CHAR.
5A6C CDFF5A             CALL CONOT
5A6F E607               ANI  7          ;LOOK AT 3 LSB'S.
5A71 32C95D             STA  NODSKS     ;SAVE IT.
5A74 0E00               MVI  C,0        ;SELECT DRIVE 0
5A76 CD805A     GOCPM:  CALL SETUP      ;SET UP JUMPS.
5A79 3ABD5D             LDA  DISKNO     ;GET DISK NUMBER TO
5A7C 4F                 MOV  C,A        ;PASS TO CCP IN C.
5A7D C30045             JMP  CPMB       ;JUMP TO CCP.

                        IF   TARDEL OR ISIO2
                IOINIT: DB   0AAH,040H,0CEH,037H
                        ENDIF

                        IF   TUART      ;IF CROMEMCO TUART,
                BAUDRS: DB   94H,0CEH,0A2H,92H,88H,84H,82H,1
                        ENDIF
                ;
                ;DISK SET UP ROUTINE. THIS ROUTINE IS COMMON TO BOTH THE
                ;READ AND WRITE ROUTINES FOR DMA OPERATION.
                ;
                        IF DMACNTL              ;IF USING DMA CONTROL
                ;
                DMARW:  STA  ERCNT              ;SAVE ERROR COUNT.
                RWDMA:  LDA  SECT               ;GET SECTOR TO READ/WRITE
                        OUT  SECTP              ;AND SEND IT FLOPPY CHIP.
                        LHLD DMAADD             ;GET CPM DMA ADDRESS.
                DMARWE: XRA  A                  ;CLEAR ACCUM.
                        OUT  CMND               ;RESET DMA CHIP.
                        MOV  A,C                ;FORCE INTERRUPT COMMAND BYTE
                        OUT  DCOM               ;SEND IT TO CONTROLLER.
                        MOV  A,E                ;BYTE COUNT TO TRANSFER
                        DCR  A                  ;COUNT = COUNT - 1.
                        OUT  WCT0               ;SEND IT TO DMA CHIP.
                        MOV  A,D                ;GET READ/WRITE CODE.
                        OUT  WCT0               ;AND TELL DMA CHIP WHAT TO DO.
                        MOV  A,L                ;GET LOW ADDRESS BYTE
                        OUT  ADR0               ;AND SEND IT TO DMA CHIP.
                        MOV  A,H                ;GET HIGH ADDRESS BYTE
                        OUT  ADR0               ;AND SEND IT TO DMA CHIP.
                        MVI  A,41H              ;SET UP FOR REQUEST CH. 0
                        OUT  CMND               ;SEND IT TO CONTROLLER.
                        CALL HDLD               ;CHECK HEAD LOAD BIT.
```

```
                        ORA  B                  ;'OR' IN THE READ/WRITE BITS.
                        OUT  DCOM               ;TELL FLOPPY CHIP WHAT TO DO.
                ;
                ;ADJUST H,L FOR 128 BYTE INCREASE.
                ;
                        PUSH D                  ;SAVE D,E
                        MOV  A,D                ;GET DMA COMMAND BYTE
                        ANI  3FH                ;STRIP OFF COMMAND BITS 7 & 6
                        MOV  D,A                ;NOW SET FOR H,L ADJUST.
                        DAD  D                  ;ADD IT TO H,L.
                        POP  D                  ;RESTORE D,E
                ;
                ;GENERAL PURPOSE WAIT ROUTINE.
                ;
                        MVI  A,60H              ;COUNT VALUE.
                CNTLOOP:DCR  A
                        JNZ  CNTLOOP            ;LOOP TILL = ZERO.
                SLOOP:  IN   DMACHK             ;CHECK FOR OPERATION DONE.
                        RLC                     ;BY LOOKING AT BIT 7.
                        JC   SLOOP              ;LOOP TILL BIT 7 = 0.
                        IN   DSTAT              ;CHECK AND RETURN DISK STATUS.
                        RET                     ;RETURN TO CALLER.
                        ENDIF
                ;
                ; SET UP JUMPS INTO CP/M IN LOWER MEMORY.
                ;
5A80 3EC3       SETUP:  MVI  A,0C3H     ;PUT JMP TO WBOOT
5A82 320000             STA  0          ;ADR AT ZERO.
5A85 21035A             LXI  H,WBOOTE
5A88 220100             SHLD 1
5A8B 320500             STA  5
5A8E 21064D             LXI  H,BDOS     ;PUT JUMP TO BDOS
5A91 220600             SHLD 6          ;AT ADR 5,6,7.
5A94 218000             LXI  H,80H      ;SET DEFAULT DMA ADR.
5A97 22BB5D             SHLD DMAADD
5A9A C9                 RET             ;RETURN FROM SETUP.
                ;
                ; WARM-BOOT:  READ ALL OF CPM BACK IN
                ; EXCEPT BIOS, THEN JUMP TO CCP.
                ;
5A9B 318000     WBOOT:  LXI  SP,80H     ;SET STACK POINTER.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        EI              ;ALLOW THEM HERE.
                        ENDIF

                        IF   LSTPAG     ;IF LIST DEVICE PAGING,
                        XRA  A          ;RESET LINE-FEED COUNT.
                        STA  LPCNT
                        ENDIF

5A9E 3ABD5D             LDA  DISKNO     ;SAVE DISK NUMBER.
5AA1 F5                 PUSH PSW        ;SAVE ON STACK
5AA2 0E00               MVI  C,0        ;SELECT DRIVE A
5AA4 CD255B             CALL SELDSK     ;SELECT DRIVE A
5AA7 CDC35B             CALL HOME       ;HOME THE DRIVE
5AAA 210000             LXI  H,0        ;CLEAR H,L
5AAD 22C45D             SHLD DRVFLG     ;CLEAR DRIVE FLAGS
5AB0 22C65D             SHLD DRVFLG+2
5AB3 0619               MVI  B,NSECTS   ;GET # SECTORS FOR CPM READ.
5AB5 0E02               MVI  C,2        ;TRACK (B)=0, SECTOR (C)=2.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        DI              ;DISABLE THEM HERE.
                        ENDIF

5AB7 210045             LXI  H,CPMB     ;GET STARTING ADDRESS.
5ABA 22BB5D     RBLK1:  SHLD DMAADD     ;SET STARTING ADDRESS.
5ABD CD455C             CALL SETSEC     ;READ STARTING AT SECTOR IN C.
5AC0 C5                 PUSH B
5AC1 CD835C             CALL READ
5AC4 C1                 POP  B
5AC5 C2D45A             JNZ  RDERR      ;IF ERROR, PRINT MESSAGE.
5AC8 0C                 INR  C          ;INCREMENT SECTOR NUMBER.
5AC9 05                 DCR  B          ;DECREMENT SECTOR COUNT.
5ACA C2BA5A             JNZ  RBLK1      ;ALL DONE WHEN D=0.
5ACD F1         ALDON:  POP  PSW        ;RESTORE DISK NUMBER.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        EI              ;ALLOW THEM AGAIN HERE.
```

```
                        ENDIF

5ACE 32BD5D             STA  DISKNO
5AD1 C3765A             JMP  GOCPM      ;GO BACK TO CPM.
                        ;
5AD4 214C5D     RDERR:  LXI  H,BTMSG    ;GET ADDRESS OF "BOOT ERROR".
5AD7 CD275D             CALL PMSG       ;PRINT IT.
5ADA CDE95A             CALL CONIN      ;READ A CHAR FROM CONSOLE.
5ADD C39B5A             JMP  WBOOT      ;DO A WARM BOOT.
                        ;
                        ; CHECK CONSOLE INPUT STATUS.
                        ;
5AE0 DB00       CONST:  IN   CSTAT      ;READ CONSOLE STATUS.
5AE2 E601               ANI  CKBR       ;LOOK AT KB READY BIT.
5AE4 3E00               MVI  A,0        ;SET A=0 FOR RETURN.

                        IF   RDYLO      ;IF STATUS READY LOW,
5AE6 C0                 RNZ             ;NOT READY WHEN NOT 0.
                        ENDIF

                        IF   RDYHI      ;IF STATUS READY HIGH,
                        RZ              ;NOT READY WHEN ZERO.
                        ENDIF

5AE7 2F                 CMA             ;IF READY A=FF.
5AE8 C9                 RET             ;RETURN FROM CONST.
                        ;
                        ; READ A CHARACTER FROM CONSOLE.
                        ;
                        CONIN:
                        IF NOT SOLOS    ;IF NOT PROC TECH SOLOS,
5AE9 DB00               IN   CSTAT      ;READ CONSOLE STATUS.
5AEB E601               ANI  CKBR       ;IF NOT READY,
                        ENDIF

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                        CALL KBD        ;READ SOL KEYBOARD.
                        JZ   CONIN      ;READY WHEN NOT ZERO.
                        ENDIF

                        IF RDYLO AND NOT SOLOS
5AED C2E95A             JNZ  CONIN      ;LOOP UNTIL LOW.
                        ENDIF

                        IF   RDYHI      ;IF READY WHEN HIGH,
                        JZ   CONIN      ;LOOP UNTIL HIGH.
                        ENDIF

                        IF NOT SOLOS    ;IF NOT PROC TECH SOLOS,
5AF0 DB01               IN   CDATA      ;READ A CHARACTER.
                        ENDIF

5AF2 E67F               ANI  7FH        ;MAKE MOST SIG. BIT = 0.

                        IF BACKSP       ;IF BACKSPACE ACTIVATED,
5AF4 FE7F               CPI  7FH        ;IS IT A RUBOUT?
5AF6 C0                 RNZ             ;RETURN IF NOT.
5AF7 3EFF               MVI  A,0FFH     ;SET NO PRINT FLAG.
5AF9 32BE5D             STA  CONOTF
5AFC 3E7F               MVI  A,7FH      ;RESTORE RUBOUT.
                        ENDIF

5AFE C9                 RET             ;RETURN FROM CONIN.
                        ;
                        ; WRITE A CHARACTER TO THE CONSOLE DEVICE.
                        ;
                        CONOT:
                        IF   BACKSP     ;IF BACKSPACE ACTIVATED,
5AFF 79                 MOV  A,C        ;GET CHARACTER.
5B00 FE7F               CPI  7FH        ;IS IT A RUBOUT?
5B02 C8                 RZ              ;IF SO, DON'T PRINT IT.
5B03 3ABE5D             LDA  CONOTF     ;GET NO PRINT FLAG.
5B06 B7                 ORA  A          ;SET CPU FLAGS.
5B07 CA1A5B             JZ   CONOTA     ;NOT SET, SO PRINT.
5B0A AF                 XRA  A          ;RESET THE FLAG
5B0B 32BE5D             STA  CONOTF     ;TO ZERO.
5B0E 0E08               MVI  C,8        ;PRINT BACKSPACE.
5B10 CD1A5B             CALL CONOTA
5B13 0E20               MVI  C,20H      ;PRINT SPACE.
5B15 CD1A5B             CALL CONOTA
```

```
5B18 0E08               MVI  C,8          ;ANOTHER BACKSPACE.
                CONOTA:
                        ENDIF

                        IF CONUL AND NOT VIDEO  ;IF NULLS REQUIRED,
                        MVI  A,0DH      ;IF IT'S A CR,
                        CMP  C          ;THEN HOP OUT
                        JZ   CONULL     ;TO NULL ROUTINE.
                        ENDIF
                CONOT1:
                        IF NOT VIDEO AND NOT SOLOS
5B1A DB00               IN   CSTAT      ;READ CONSOLE STATUS.
5B1C E680               ANI  CPTR       ;IF NOT READY,
                        ENDIF

                        IF RDYLO AND NOT VIDEO AND NOT SOLOS
5B1E C21A5B             JNZ  CONOT1     ;LOOP UNTIL LOW.
                        ENDIF

                        IF RDYHI AND NOT VIDEO  ;IF READY WHEN HIGH,
                        JZ   CONOT1     ;LOOP UNTIL HIGH.
                        ENDIF

                        IF NOT VIDEO AND NOT SOLOS
5B21 79                 MOV  A,C        ;GET CHARACTER.
5B22 D301               OUT  CDATA      ;PRINT IT.
5B24 C9                 RET             ;RETURN.
                        ENDIF

                        IF CONUL AND NOT VIDEO
                CONULL: PUSH B          ;SAVE B&C.
                        MVI  B,CNULL+1  ;GET NULL COUNT.
                CONUL1: CALL CONOT1     ;PRINT CR.
                        MVI  C,0        ;GET NULL CHAR.
                        DCR  B          ;DECREMENT COUNTER.
                        JNZ  CONUL1     ;DO NEXT NULL.
                        POP  B          ;RESTORE B&C.
                        MOV  A,C        ;RESTORE A.
                        RET             ;RETURN.
                        ENDIF

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                        PUSH B          ;SAVE B&C.
                        MOV  B,C        ;PUT CHAR IN B REG.
                        CALL SCRN       ;OUTPUT CHAR TO SOLOS.
                        POP  B          ;RESTORE B&C.
                        MOV  A,C        ;PUT CHAR IN A.
                        RET             ;RETURN FROM CONOT.
                        ENDIF

                        IF   VIDEO      ;IF ANY VIDEO BOARD,
                ;
                ;       VIDEO DRIVER FOR VB1-B OR VDM 1 BOARD.
                ;       WRITTEN BY G.W.MULCHIN
                ;       9-16-78
                ;
                        MOV A,C         ;GET THE CHAR INTO REG A
                        PUSH H          ;SAVE REGISTERS
                        PUSH D
                        PUSH B
                        PUSH PSW        ;CHAR. IS IN REG A
                        CALL VIDPRO     ;DO VIDEO ROUTINE
                        POP PSW         ;RESTORE REGISTERS
                        POP B
                        POP D
                        POP H
                        RET             ;BACK TO CALLER
                ;
                VIDPRO: LHLD VDMP       ;GET SCREEN POSITION POINTER
                        CPI CR          ;IS IT CARRIAGE RETURN?
                        JZ CARRET
                        CPI LF          ;IS IT LINEFEED?
                        JZ LFNOT        ;PUT IN A BLANK
                        CPI BKSP        ;IS IT A RUBOUT?
                        JZ BS
                        CPI  FF         ;IS IT CNTL-L?
                        JZ CLEAR
                        MOV M,A         ;IT HAS TO BE DATA
                UPDATE: INX H           ;UPDATE POSITION
                GONE:   MVI M,0A0H      ;PUT CURSOR ON SCREEN
```

```
                        JMP MAXLIN      ;CHECK FOR LINE > 64
                LFNOT:  MVI M,' '       ;PUT IN A SPACE
                        JMP UPDATE      ;GET OUT NOW
                BS:     MVI M,' '
                        LHLD VDMP       ;GET CURRENT POSITION
                        DCX H
                        SHLD VDMP       ;SAVE CURSOR POSITION
                        JMP GONE
                CARRET: MVI M,' '       ;CHAR. IS A CARRIAGE RET.
                        MOV A,L         ;UPDATE NEXT POSITION
                        ANI 0C0H
                        ADI 40H         ;SET UP FOR NEW LINE
                        EOV L,A         ;ADDRESS OF NEW LINE
                        MVI A,0
                        ADC H
                        MOV H,A
                MAXLIN: SHLD VDMP       ;SAVE POINTER FOR NEXT CHAR.
                        MVI A,7FH
                        ANA L
                        RNR             ;EXIT BACK TO MAIN PROGRAM
                        MVI M,' '
                        LXI H,SCRNTOP
                        SHLD VDMP
                        LXI H,LINES15   ;15 LINES OF SCREEN DATA
                        LXI D,SCREEN    ;TOP OF SCREEN. SET UP
                        LXI B,SCRLCNT   ; TO SCROLL 15 LINES
                SCROLL: MOV A,M         ;START SCROLLLING UP
                        STAX D          ;STUFF REG A BY WAY OF D,E
                        INX H
                        INX D
                        DCX B
                        XRA A
                        CMP B           ;15 LINES YET?
                        JNZ SCROLL
                        CMP C
                        JNZ SCROLL      ;NOT DONE YET!
                        LXI H,SCRNTOP
                BLANK:  MVI M,' '       ;PUT BLANK ON SCREEN
                        INX H           ;BLANK ENTIRE DATA LINE
                        MOV A,L
                        ANI 3FH
                        JNZ BLANK
                        LXI H,SCRNTOP
                        MVI M,0A0H      ;STUFF CURSOR BACK
                        RET             ;ALL DONE.
                CLEAR:  LXI H,SCREEN    ;CLEAR SCREEN
                        MVI A,ENDSCR    ;THIS IS END CHECK
                CLERA:  CMP H           ;IS IT END YET?
                        JZ FINISH
                        MVI M,' '       ;PUT SPACE ON SCREEN
                        INX H           ;BUMP POINTER
                        JMP CLERA       ;GO BACK IF NOT DONE
                FINISH: LXI H,SCRNTOP
                        MVI M,0A0H      ;STUFF CURSOR BACK AGAIN
                        SHLD VDMP       ;SAVE CURSOR POSITION.
                        RET             ;ALL DONE.
                        ENDIF           ;END OF VDM DRIVER.
                ;
                ; SELECT DISK NUMBER ACCORDING TO REGISTER C.
                ; ALSO CHECK IF DISK HAS BEEN LOGGED IN BEFORE.
                ; IF NOT, LOG IT IN AND CHECK DENSITY
                ;
5B25 79         SELDSK: MOV  A,C        ;GET NEW DISK NUMBER.
5B26 E603               ANI  3          ;ONLY LOOK AT 2 LSB'S.
5B28 21BD5D             LXI  H,DISKNO   ;GET ADR OF OLD DISK NO.
5B2B BE                 CMP  M          ;NEW = OLD?
5B2C C8                 RZ              ;IF SO, RETURN.
5B2D F5                 PUSH A          ;SAVE DISK NUMBER.
5B2E 3AC95D             LDA  NODSKS     ;GET NUMBER OF DISKS.
5B31 3D                 DCR  A          ;IF MORE THAN ONE DISK,
5B32 C24A5B             JNZ  SELMOR     ;TAKE CARE OF IT.
5B35 21605D             LXI  H,MNTMSG   ;GET ADR OF MOUNT MESSAGE.
5B38 CD275D             CALL PMSG       ;PRINT "MOUNT ".
5B3B F1                 POP  A          ;GET DISK NUMBER.
5B3C 32BD5D             STA  DISKNO     ;UPDATE OLD WITH NEW.
5B3F C641               ADI  'A'        ;ADD ASCII FOR 'A'.
5B41 4F                 MOV  C,A        ;PUT INTO C.
5B42 CDFF5A             CALL CONOT      ;PRINT IT.
5B45 CDE95A             CALL CONIN      ;READ A CARRIAGE RETURN.
5B48 AF                 XRA  A          ;SET A=0 FOR NO ERRO IND.
```

```
5B49 C9                 RET             ;RETURN FROM SELDSK.
5B4A F1         SELMOR: POP  A          ;MAKE STACK RIGHT.
5B4B 7E                 MOV  A,M        ;GET OLD DISK NUMBER.
                        IF   DUAL       ;IF DUAL DRIVE,
                        ANI  0FEH       ;CLEAR OUT BIT 0.
                        ENDIF

5B4C 5F                 MOV  E,A        ;PUT OLD DISK NO. IN D&E.
5B4D 1600               MVI  D,0
5B4F 21C05D             LXI  H,TRTAB    ;GET ADDRESS OF TRACK TABLE.
5B52 E5                 PUSH H          ;SAVE FOR LATER
5B53 19                 DAD  D          ;ADD DISK NO. TO ADDRESS.
5B54 DBF9               IN   TRACK      ;READ 1771 TRACK REGISTER.
5B56 77                 MOV  M,A        ;PUT INTO TABLE.
5B57 79                 MOV  A,C        ;GET NEW DISK NUMBER.

                        IF   DUAL       ;IF A DUAL DRIVE,
                        ANI  0FEH       ;CLEAR BIT 0.
                        ENDIF

5B58 5F                 MOV  E,A        ;PUT NEW DISK NO. IN D&E.
5B59 E1                 POP  H          ;RESTORE ADDRESS OF TRACK TABLE
5B5A 19                 DAD  D          ;ADD DISK NO. TO ADDRESS.
5B5B 7E                 MOV  A,M        ;GET NEW TRACK NUMBER.
5B5C D3F9               OUT  TRACK      ;PUT INTO 1771 TRACK REG.
5B5E 79                 MOV  A,C        ;UPDATE OLD DISK NUMBER.
5B5F 32BD5D             STA  DISKNO
5B62 87                 ADD  A          ;PUT BITS 1&2 AT 4&5.
5B63 87                 ADD  A
5B64 87                 ADD  A
5B65 87                 ADD  A
5B66 32CC5D             STA  LATCH      ;SAVE NEW LATCH CODE.
                ;
                ;SELECT DENSITY BASED ON BYTE VALUE
                ;IN TRACK 0 SECTOR 1 ADDRESS 7EH.
                ;IF THIS BYTE IS A 0DDH, THEN THE
                ;DISK IS DOUBLE DENSITY, ELSE, IT'S
                ;SINGLE DENSITY.
                ;
5B69 C5         DENSITY:PUSH B          ;SAVE B,C
5B6A 21C45D             LXI H,DRVFLG    ;INDEX INTO DRIVE BYTE FLAG
5B6D 0600               MVI B,0         ;ZERO REG B
5B6F 09                 DAD B           ;ADD THE DRIVE NUMBER
5B70 7E                 MOV A,M         ;GET THE BYTE FLAG
5B71 B7                 ORA A           ;SET THE FLAGS
5B72 FA9B5B             JM LOGED        ;SKIP IF LOGGED IN BEFORE
5B75 E5                 PUSH H          ;SAVE DRIVE TABLE POINTER.
5B76 3ACC5D             LDA  LATCH
5B79 D3FC               OUT  DCONT      ;SELECT DRIVE
5B7B 2ABB5D             LHLD DMAADD     ;GET PRESENT DMA ADDRESS
5B7E E5                 PUSH H          ;SAVE ON STACK
5B7F CDC35B             CALL HOME       ;HOME DRIVE
5B82 3E01               MVI A,1         ;SET FOR SECTOR 1
5B84 32BA5D             STA  SECT       ;SAVE SECTOR TO READ
5B87 21CE5D             LXI H,DBUFF     ;POINT TO DMA BUFFER
5B8A 22BB5D             SHLD DMAADD     ;SET UP THE DMA ADDRESS
5B8D CD835C             CALL READ       ;READ TRK 0, SEC 1 INTO DBUFF
5B90 E1                 POP  H          ;RECOVER DMAADD FROM STACK
5B91 22BB5D             SHLD DMAADD     ;RESTORE IT
5B94 E1                 POP  H          ;RESTORE DRIVE TABLE POINTER.
5B95 3A4C5E             LDA DBUFF+7EH   ;GET THE DENSITY CODE BYTE
5B98 F680               ORI  80H        ;set logged in bit
5B9A 77                 MOV  M,A        ;place it in flag table.
5B9B 010700     LOGED:  LXI  B,7        ;index value through drive table.
5B9E E612               ANI  12H        ;MASK DENSITY AND SIDE BITS OUT.
5BA0 B7                 ORA  A          ;SINGLE DENSITY?
5BA1 212D5A             LXI  H,SDTAB    ;point to start of tables
5BA4 CAA85B             JZ   DENSIT1    ;yes, overlay param. block
                ;
                        IF   DUBSID
                        DAD  B          ;no, add offset to next table
                        CPI  2          ;single den doub sided?
                        JZ   DENSIT1    ;yes
                        DAD  B          ;no
                        CPI  10H        ;doub den single sided?
                        JZ   DENSIT1    ;yes
                        ENDIF
                ;
5BA7 09                 DAD  B          ;no, must be doub den , doub sided
5BA8 EB         DENSIT1:XCHG            ;drive table pointer --> d,e
```

```
5BA9 1A                 LDAX D          ;GET LOG BYTE
5BAA 13                 INX  D          ;BUMP POINTER
5BAB 32C85D             STA  DENS       ;SET DENSITY
5BAE 1A                 LDAX D          ;GET A BYTE FROM TABLE
5BAF 13                 INX  D          ;BUMP POINTER
5BB0 32154D             STA BDOS+15     ;CHANGE TRANS FUNCTION
5BB3 213A4D             LXI H,BDOS+52   ;POINT TO CONSTANT DATA AREA (CP/M)
5BB6 0605               MVI B,5         ;COUNT = 5
5BB8 1A         MOVE:   LDAX D          ;GET A BYTE FROM TABLE
5BB9 77                 MOV M,A         ;OVERLAY CP/M
5BBA 13                 INX D           ;BUMP
5BBB 23                 INX H           ; POINTERS
5BBC 05                 DCR B           ;DECREASE COUNT
5BBD C2B85B             JNZ MOVE        ; AND LOOP TILL DONE
5BC0 C1                 POP  B          ;RESTORE B,C
5BC1 AF                 XRA  A          ;SET A = 0.
5BC2 C9                 RET             ;RETURN FROM SELDSK.
                ;
                ; MOVE DISK TO TRACK ZERO.
                ;
5BC3 3E02       HOME:   MVI  A,0+STPRAT ;RESTORE
5BC5 D3F8               OUT  DCOM
5BC7 0E00               MVI  C,0        ;TRACK 0
                ;
                ; SET TRACK NUMBER TO WHATEVER IS IN REGISTER C.
                ; ALSO PERFORM MOVE TO THE CORRECT TRACK (SEEK).
                ;
5BC9 2ACC5D     SETTRK: LHLD LATCH              ;get new and old latch.
5BCC 7C                 MOV  A,H                ;get latch value.
5BCD E6B7               ANI  0B7H               ;strip density and side bits.
5BCF 67                 MOV  H,A                ;restore it.
                ;
                        IF   DUBSID             ;if using double sided drive.
                        LDA  DENS               ;check if double sided.
                        RRC
                        RRC                     ;look at bit 1.
                        JNC  NOTSID             ;if bit 1 = 0, it's single sided.
                        MOV  A,C                ;it's doub sided, so get track number.
                        RRC                     ;divide by 2.
                        MOV  B,A                ;save it in reg b.
                        MOV  A,L                ;get old latch value.
                        JC   SIDE2              ;change side if odd track.
                        ANI  0BFH               ;clear side bit from latch.
                        JMP  SETLAT             ;go set the latch.
                ;
                SIDE2:  ORI  40H                ;turn on side select bit.
                SETLAT: STA  CLATCH             ;save it for later.
                        ANI  0BFH               ;clear side bit.
                        CALL OLDLAT             ;check for drive change.
                        MOV  A,B                ;restore doub sided trk number.
                        ANI  7FH                ;clear bit 7.
                        MOV  C,A                ;trk number now in reg c.
                        JMP  TRKSET             ;check for density of track going to.
                        ENDIF
                ;
                        IF   NOT DUBSID         ;if not using double sided drive
5BD0 C3DE5B             JMP  NOTSID             ;jump around subroutine.
                        ENDIF
                ;
5BD3 BC         OLDLAT: CMP  H                  ;new = old?
5BD4 3EFF               MVI  A,0FFH             ;if not, set = ff
5BD6 C2DA5B             JNZ  SFLAG
5BD9 2F                 CMA                     ;new = old, set = 0.
5BDA 32BF5D     SFLAG:  STA  HLSF               ;save head load select flag.
5BDD C9                 RET
                ;
5BDE 7D         NOTSID: MOV  A,L                ;get latch value.
5BDF 32CD5D             STA  CLATCH             ;save it
5BE2 CDD35B             CALL OLDLAT             ;check for drive change.
                ;
5BE5 3AC85D     TRKSET: LDA  DENS       ;CHECK DRIVE DENSITY FLAG
5BE8 0F                 RRC             ;IS IT ZERO ?
5BE9 D2FA5B             JNC  TRKSD      ;YES, WE ARE SINGLE DENSITY
5BEC 79                 MOV  A,C        ;RESTORE TRACK NUMBER
5BED FE01               CPI  1          ;IS IT TRACK 1 ?
5BEF DAFA5B             JC   TRKSD      ;IF LESS THEN, SET SINGLE DEN.
5BF2 3ACD5D             LDA  CLATCH     ;GET THE CURRENT LATCH VALUE
5BF5 F608               ORI  DDEN       ;SET BIT 4 ON (DOUB DENSITY ON)
5BF7 C3FF5B             JMP  TRKDD
5BFA 3ACD5D     TRKSD:  LDA  CLATCH     ;GET CURRENT LATCH VALUE
```

```
5BFD E6F7               ANI  DDDSB      ;TURN OFF BIT 4 (SINGLE DENSITY ON)
5BFF 32CD5D     TRKDD:  STA  CLATCH     ;SAVE NEW CURRENT LATCH VALUE
5C02 D3FC               OUT  DCONT      ;  AND SET THE HARDWARE LATCH
5C04 79                 MOV  A,C        ;RESTORE TRACK NUMBER
5C05 32B95D             STA  TRK        ;UPDATE OLD WITH NEW.
                ;
                ; MOVE THE HEAD TO THE TRACK IN REGISTER A.
                ;
5C08 C5         SEEK:   PUSH B          ;SAVE B&C.
5C09 47                 MOV  B,A        ;SAVE DESTINATION TRACK.
5C0A 3E0A               MVI  A,RTCNT    ;GET RETRY COUNT.
5C0C 32CB5D     SRETRY: STA  SERCNT     ;STORE IN ERROR COUNTER.
5C0F DBF9               IN   TRACK      ;READ PRESENT TRACK NO.
5C11 4F                 MOV  C,A        ;SAVE IN C.
5C12 79                 MOV  A,C        ;DELAY.
5C13 B8                 CMP  B          ;SAME AS NEW TRACK NO.?
5C14 78                 MOV  A,B
5C15 C21A5C             JNZ  NOTHR      ;JUMP IF NOT THERE.
5C18 C1         THERE:  POP  B          ;RESTORE B&C.
5C19 C9                 RET             ;RETURN FROM SEEK.
                NOTHR:
                ;DELAY LOOP TO ALLOW TUNNEL
                ;ERASE TO END DURING WRITE.
                ;
5C1A F5                 PUSH PSW
5C1B 3ED0               MVI  A,0D0H     ;COUNT = 208
5C1D 3D         BUSY1:  DCR  A          ;LOOP
5C1E C21D5C             JNZ  BUSY1      ;  TILL ZERO
5C21 F1                 POP  PSW
5C22 78                 MOV  A,B        ;RESTORE A FROM B.
5C23 D3FB               OUT  DDATA      ;TRACK TO DATA REGISTER.
5C25 3E16               MVI  A,14H+STPRAT+HLAB  ;GET STEP RATE, DO
5C27 D3F8               OUT  DCOM       ;SEEK WITH VERIFY.

                        IF   NOT DMACNTL
5C29 DBFC               IN   WAIT
5C2B DBF8               IN   DSTAT      ;CHECK STATUS
                        ENDIF

                        IF   DMACNTL
                BUSY2:  CALL SLOOP      ;USE DMA CHECK PORT
                        ENDIF

5C2D E691               ANI  91H        ;LOOK AT BITS.
5C2F CA185C             JZ   THERE      ;OK IF ZERO.

                        IF TESTING      ;IF TESTING FOR ERRORS
                        PUSH H          ;SAVE H&L.
                        LXI  H,SECNT    ;GET ADR OF SEEK ERR CTR.
                        INR  M          ;ONE MORE SEEK ERROR.
                        POP  H          ;RESTORE H&L.
                        ENDIF

5C32 3ACB5D             LDA  SERCNT     ;GET ERROR COUNT.
5C35 3D                 DCR  A          ;DECREMENT COUNT.
5C36 C20C5C             JNZ  SRETRY     ;RETRY SEEK.
5C39 C1                 POP  B          ;RESTORE B&C.
5C3A 21585D             LXI  H,SKMSG    ;PRINT "SEEK ".
5C3D DBF8               IN   DSTAT      ;READ DISK STATUS.
5C3F E691               ANI  91H        ;LOOK AT ERROR BITS.
5C41 57                 MOV  D,A        ;PUT IN REG D.
5C42 C3AC5C             JMP  ERMSG      ;DO COMMON ERR MESSAGES.
                ;
                ; SET DISK SECTOR NUMBER.
                ;
5C45 79         SETSEC: MOV  A,C        ;GET SECTOR NUMBER.
5C46 32BA5D             STA  SECT       ;PUT AT SECT # ADDRESS.
5C49 C9                 RET             ;RETURN FROM SETSEC.
                ;
                ; SET DISK DMA ADDRESS.
                ;
5C4A 60         SETDMA: MOV  H,B        ;MOVE B&C TO H&L.
5C4B 69                 MOV  L,C
5C4C 22BB5D             SHLD DMAADD     ;PUT AT DMA ADR ADDRESS.
5C4F C9                 RET             ;RETURN FROM SETDMA.
                ;
                ; HDLD - GET HEAD-LOAD BIT IF REQUIRED.
                ;
5C50 3ABF5D     HDLD:   LDA  HLSF       ;GET HEAD-LOAD FLAG.
5C53 B7                 ORA  A          ;IS A = ZERO?
```

```
5C54 CA655C             JZ   HDLD1        ;HOP IF SO.
5C57 2F                 CMA               ;SET A = 0.
5C58 32BF5D             STA  HLSF         ;SET FLAG = 0 IF NOT.
                        ;
                        ; IF CHANGING TO A NEW DRIVE, PERFORM A SEEK
                        ; TO THE SAME TRACK TO ALLOW THE HEAD TO UNLOAD.
                        ;
5C5B DBF9               IN   TRACK        ;GET PRESENT TRACK
5C5D D3FB               OUT  DDATA        ; AND TELL CONTROLLER
5C5F 3E16               MVI  A,14H+STPRAT       ;GET STEP COMMAND
5C61 D3F8               OUT  DCOM         ;SEND IT TO FLOPPY CONTROLLER

                        IF   NOT DMACNTL
5C63 DBFC               IN   WAIT         ;WAIT FOR INTRQ
                        ENDIF

                        IF   DMACNTL
                        CALL SLOOP        ;USE DMA CHECK PORT
                        ENDIF

5C65 DBF8       HDLD1:  IN   DSTAT        ;READ 1771 STATUS.
5C67 E620               ANI  20H          ;LOOK AT HL BIT.
5C69 3E04               MVI  A,4
5C6B C8                 RZ
5C6C 97                 SUB  A            ;HEAD IS LOADED
5C6D C9                 RET               ;RETURN FROM HDLD.
                        ;
                        IF   NOT DMACNTL
5C6E 32CA5D     DSKSET: STA  ERCNT        ;SAVE RETRY COCUNT
5C71 3ED0               MVI  A,0D0H       ;CAUSE INTRP
5C73 D3F8               OUT  DCOM
5C75 E3                 XTHL              ;SOME
5C76 E3                 XTHL              ;DELAY
                        ENDIF

                        IF   INTRP
                        DI
                        ENDIF

                        IF   NOT DMACNTL
5C77 2ABB5D             LHLD DMAADD       ;STARTING ADDRESS
5C7A 3ABA5D             LDA  SECT         ;GET SECTOR NUMBER
5C7D D3FA               OUT  SECTP        ;TELL CONTROLLER
5C7F CD505C             CALL HDLD         ;CHECK FOR HEAD LOADED
5C82 C9                 RET
                        ENDIF
                        ;
                        ; READ THE SECTOR AT SECT, FROM THE PRESENT TRACK.
                        ; USE STARTING ADDRESS AT DMAADD.
                        ;
5C83 3E0A       READ:   MVI  A,RTCNT      ;GET RETRY COUNT.
                RRETRY:
                        IF   DMACNTL
                        LXI  B,80D0H      ;FLOPPY READ, FORCE INTRP.
                        LXI  D,4080H      ;DMA READ, BYTE COUNT.
                        CALL DMARW        ;COMMON READ/WRITE ROUTINE.
                        ENDIF

                        IF  INTRP         ;IF INTERRUPTS ALLOWED,
                        DI                ;DISABLE THEM HERE.
                        ENDIF

                        IF   NOT DMACNTL
5C85 0680               MVI  B,80H        ;FLOPPY READ COMMAND.
5C87 CD6E5C             CALL DSKSET       ;SET UP DISK CONTROLLER
5C8A B0                 ORA  B            ;SET READ COMMAND
5C8B D3F8       READE:  OUT  DCOM         ;SEND COMMAND TO 1771.
5C8D DBFC       RLOOP:  IN   WAIT         ;WAIT FOR DRQ OR INTRQ.
5C8F B7                 ORA  A            ;SET FLAGS.
5C90 F29A5C             JP   RDDONE       ;DONE IF INTRQ.
5C93 DBFB               IN   DDATA        ;READ A DATA BYTE FROM DISK.
5C95 77                 MOV  M,A          ;PUT BYTE INTO MEMORY.
5C96 23                 INX  H            ;INCREMENT MEMORY POINTER.
5C97 C38D5C             JMP  RLOOP        ;KEEP READING.
5C9A DBF8       RDDONE: IN   DSTAT        ;READ DISK STATUS.
                        ENDIF

                        IF  INTRP         ;IF INTERRUPTS ALLOWED,
                        EI                ;ALLOW AGAIN HERE.
                        ENDIF
```

```
5C9C E69D            ANI  9DH          ;LOOK AT ERROR BITS.
5C9E C8              RZ                ;RETURN IF NONE.
5C9F CDCB5C          CALL ERCHK        ;CHECK FOR SEEK ERROR.

                     IF  TESTING
                     LXI  H,RECNT      ;GET RD ERR COUNT ADDR.
                     INR  M            ;ONE MORE ERROR.
                     MOV  A,M
                     CMA
                     OUT  0FFH
                     ENDIF

5CA2 3ACA5D          LDA  ERCNT        ;GET ERROR COUNT.
5CA5 3D              DCR  A            ;DECREMENT COUNT.
5CA6 C2855C          JNZ  RRETRY       ;TRY TO READ AGAIN.
5CA9 213B5D          LXI  H,RDMSG      ;PRINT "READ ".
5CAC CD275D    ERMSG:CALL PMSG         ;PRINT ORIGIN MESSAGE.
               ERMSG1:
                     IF NOT VIDEO      ;NEED MORE ROOM?
5CAF 7A              MOV  A,D          ;GET ERROR BITS.
5CB0 E610            ANI  10H          ;IF BIT 4 IS HIGH,
5CB2 21325D          LXI  H,RNMSG      ;PRINT "RECORD NOT FOUND"
5CB5 C4275D          CNZ PMSG
5CB8 7A              MOV  A,D          ;GET ERROR BITS.
5CB9 E608            ANI  8H           ;IF BIT 3 IS HIGH,
5CBB 21365D          LXI  H,CRCMSG     ;PRINT "CRC ERROR".
5CBE C4275D          CNZ PMSG
                     ENDIF

5CC1 21515D          LXI  H,ERRMSG     ;PRINT "ERROR."
5CC4 CD275D          CALL PMSG
5CC7 3E01            MVI  A,1          ;SET FOR PERM ERR MSG.
5CC9 B7              ORA  A            ;SET FLAGS.
5CCA C9              RET
               ;
               ; ERCHK - CHECK FOR RECORD NOT FOUND ERROR.
               ;
5CCB 57        ERCHK:MOV  D,A          ;SAVE ERROR BITS IN D.
5CCC E610            ANI  10H          ;IF RECORD NOT FOUND,
5CCE C8              RZ
               ;
               ;CHECK FOR SEEK TO CORRECT TRACK,
               ;AND CHANGE IF NECESSARY.
5CCF 3EC4      CHKSK:MVI  A,0C4H       ;SEND COMMAND TO 1771
5CD1 D3F8            OUT  DCOM         ;TO READ ADDRESS.

                     IF   NOT DMACNTL
5CD3 DBFC            IN   WAIT         ;WAIT FOR DRQ OR INTRQ.
                     ENDIF

                     IF   DMACNTL
                     CALL SLOOP
                     ENDIF

5CD5 DBFB            IN   DDATA        ;READ THE TRACK ADDRESS.
5CD7 47              MOV  B,A          ;SAVE IN REGISTER B.
5CD8 DBFC      CHKS2:IN   WAIT         ;WAIT FOR INTRQ.
5CDA B7              ORA  A            ;SET FLAGS.
5CDB F2E35C          JP   CHKS3        ;DONE WITH READ ADR OP.
5CDE DBFB            IN   DDATA        ;READ ANOTHER BYTE.
5CE0 C3D85C          JMP  CHKS2        ;DO IT AGAIN.
5CE3 DBF8      CHKS3:IN   DSTAT        ;READ DISK STATUS.
5CE5 B7              ORA  A            ;SET FLAGS.
5CE6 CAF25C          JZ   CHKS4        ;READ ADR OK IF 0.
5CE9 CDC35B          CALL HOME         ;OTHERWISE, HOME FIRST.
5CEC 3AB95D    CHKS5:LDA  TRK
5CEF C3085C          JMP  SEEK
5CF2 78        CHKS4:MOV  A,B          ;UPDATE TRACK REGISTER.
5CF3 D3F9            OUT  TRACK
5CF5 C3EC5C          JMP  CHKS5        ;RETURN FROM ERCHK.
               ;
               ; WRITE THE SECTOR AT SECT, ON THE PRESENT TRACK.
               ; USE STARTING ADDRESS AT DMAADD.
               ;
5CF8 3E0A      WRITE:MVI  A,RTCNT      ;GET RETRY COUNT.
               WRETRY:
                     IF   DMACNTL
                     LXI  B,0A0D0H     ;FLOPPY WRITE, FORCE INTRP
                     LXI  D,08080H     ;DMA WRITE, BYTE COUNT
```

```
                        CALL DMARW      ;COMMON ROUTINE
                        ENDIF

                        IF   NOT DMACNTL
5CFA 06A0               MVI  B,0A0H     ;WRITE COMMAND
5CFC CD6E5C             CALL DSKSET
5CFF B0                 ORA  B
5D00 D3F8       WRITE2: OUT  DCOM
5D02 DBFC       WLOOP1: IN   WAIT       ;WAIT FOR READY.
5D04 B7                 ORA  A          ;SET FLAGS.
5D05 F20F5D             JP   WDONE      ;HOP OUT WHEN DONE.
5D08 7E                 MOV  A,M        ;GET BYTE FROM MEM.
5D09 D3FB               OUT  DDATA      ;WRITE ONTO DISK.
5D0B 23                 INX  H          ;INCREMENT MEM PTR.
5D0C C3025D             JMP  WLOOP1     ;KEEP WRITING.
5D0F DBF8       WDONE:  IN   DSTAT
                        ENDIF

                        IF  INTRP       ;IF INTERRUPTS ALLOWED,
                        EI              ;ENABLE AGAIN HERE.
                        ENDIF

5D11 E6FD               ANI  0FDH       ;LOOK AT THESE BITS.
5D13 C8                 RZ              ;RETURN IF NO ERR.
5D14 CDCB5C             CALL ERCHK      ;CHECK/CORRECT SEEK ERR.

                        IF  TESTING     ;IF TESTING FOR ERRORS
                        LXI  H,WECNT    ;GET ADR OF WRITE ERR CTR.
                        INR  M          ;ONE MORE WRITE ERROR.
                        MOV  A,M
                        CMA
                        OUT  0FFH
                        ENDIF

5D17 3ACA5D             LDA  ERCNT      ;GET ERROR COUNT.
5D1A 3D                 DCR  A          ;DECREMENT COUNT.
5D1B C2FA5C             JNZ  WRETRY     ;TRY TO WRITE AGAIN.
5D1E 21435D             LXI  H,WTMSG    ;PRINT "WRITE ".

                        IF NOT VIDEO    ;NEED MORE ROOM?
5D21 CD275D             CALL PMSG
5D24 C3AF5C             JMP  ERMSG1     ;DO COMMON MESSAGES.
                        ENDIF

                        IF   VIDEO      ;WE NEED A RETURN.
                        JMP  ERMSG
                        ENDIF

                ;
                ; PRINT THE MESSAGE AT H&L UNTIL A ZERO.
                ;
5D27 7E         PMSG:    MOV  A,M       ;GET A CHARACTER.
5D28 B7                  ORA  A         ;IF IT'S ZERO,
5D29 C8                  RZ             ;RETURN.
5D2A 4F                  MOV  C,A       ;OTHERWISE,
5D2B CDFF5A              CALL CONOT     ;PRINT IT.
5D2E 23                  INX  H         ;INCREMENT H&L,
5D2F C3275D              JMP  PMSG      ;AND GET ANOTHER.
                ;
                ; CBIOS MESSAGES
                ;
                        IF NOT VIDEO    ;NEED MORE ROOM?
5D32 49442000   RNMSG:  DB   'ID ',0
5D36 435243200CRCMSG:   DB   'CRC ',0
                        ENDIF

5D3B 0D0A526561RDMSG:   DB   0DH,0AH,'Read ',0
5D43 0D0A577269WTMSG:   DB   0DH,0AH,'Write ',0
5D4C 426F6F7420BTMSG:   DB   'Boot '
5D51 4552524F52ERRMSG:  DB   'ERROR.',0
5D58 0D0A536565SKMSG:   DB   0DH,0AH,'Seek ',0
5D60 0D0A4D6F75MNTMSG:  DB   0DH,0AH,'Mount ',0
5D69 0D0A546172SMSG:    DB   0DH,0AH,'Tarbell '
5D73 3234               DB   MSIZE/10+'0',MSIZE MOD 10 + '0'
5D75 4B2043504D         DB   'K CPM V1.4 of 7-14-80'
5D8A 0D0A               DB   0DH,0AH

                        IF  TARBELL     ;IF USING TARBELL CPU.
                        DB  'Tarbell CPU, '
                        ENDIF
```

```
                        IF  STD             ;IF STANDARD I/O,
5D8C 5374616E64         DB   'Standard '
                        ENDIF

                        IF  MSIO2           ;IF MITS 2SIO,
                        DB   '2SIO '
                        ENDIF

                        IF  ISIO2           ;IF IMSAI SIO-2,
                        DB  'SIO-2 '
                        ENDIF

                        IF  TUART           ;IF TUART,
                        DB  'Tuart '
                        ENDIF

                        IF  SOLOS           ;IF PROC TECH SOLOS,
                        DB  'Solos '
                        ENDIF

                        IF   VDM            ;IF PROC TECH VDM,
                        DB   'VDM '
                        ENDIF

                        IF   FLASH          ;IF VG FLASHWRITER,
                        DB   'Flashwriter '
                        ENDIF

                        IF   VB1            ;IF SSM VB1-B,
                        DB   'VB1 '
                        ENDIF

                        IF  DUBSID          ;IF DOUBLE-SIDED,
                        DB   'Double-Sided '
                        ENDIF

                        IF  DUAL            ;IF DUAL DRIVE,
                        DB  'Dual '
                        ENDIF

                        IF  DMACNTL         ;IF USING DMA CONTROL
                        DB  'DMA '
                        ENDIF
5D95 5645522E           DB   'VER.'

5D99 0D0A486F77         DB   0DH,0AH,'How Many Disks? ',0

                ;
                ; WRITE A CHARACTER ON LISTING DEVICE.
                ;
                LIST:
                        IF LSTNUL           ;IF NULLS OR PAGING,
                        MVI  A,0DH          ;IF IT'S A CR,
                        CMP  C              ;THEN HOP OUT TO
                        JZ   LINUL          ;NULL ROUTINE.
                        ENDIF

                        IF LSTPAG           ;IF PAGING
                        MVI A,0AH           ;GET A LINEFEED
                        CMP C               ;DOES IT MATCH?
                        JZ LINUL3
                        ENDIF

5DAC DB02       LTBSY:  IN   LSTAT          ;READ LISTER STATUS.

                        IF   NOT TARDEL
5DAE E680               ANI  LRBIT          ;LOOK AT READY BIT.
                        ENDIF

                        IF   TARDEL
                        ANI  81H            ;MASK
                        XRI  81H
                        ENDIF

                        IF   TARDEL OR RDYLO    ;IF READY WHEN LOW,
5DB0 C2AC5D             JNZ  LTBSY          ;LOOP TILL LOW.
                        ENDIF

                        IF   NOT TARDEL AND RDYHI       ;IF READY WHEN HIGH,
                        JZ   LTBSY          ;LOOP TILL HIGH.
                        ENDIF
```

```
5DB3 79                 MOV  A,C        ;GET DATA BYTE.
5DB4 D303               OUT  LDATA      ;PRINT IT.
5DB6 C9                 RET             ;RETURN FROM LIST.

                        IF LSTNUL OR LSTPAG  ;IF NULLS OR PAGING,
                LINUL:  PUSH B          ;SAVE B&C.
                        MVI  B,(LNULL AND 0FFH)+1 ;GET NULL COUNT
                LINUL1: CALL LIBSY      ;PRINT (CR FIRST).
                        MVI  C,0        ;GET NULL CHAR.
                        DCR  B          ;DECREMENT COUNTER.
                        JNZ  LINUL1     ;DO NEXT NULL.
                        JMP  LINUL2     ;EXIT THE ROUTINE.
                        ENDIF

                        IF   LSTPAG     ;IF LIST DEV. PAGING,
                LINUL3: PUSH B          ;SAVE B,C PAIR
                        LDA  LFCNT      ;GET LINE-FEED COUNT.
                        INR  A          ;INCREMENT IT.
                        STA  LFCNT      ;SAVE IT BACK.
                        CPI  LINCNT-(LINCNT/11)  ;END OF PAGE?
                        MVI  B,1        ;SET UP FOR 1 LF.
                        JNZ  NOTEOP     ;HOP IF NOT END.
                        XRA  A          ;SET LF COUNT = 0.
                        STA  LFCNT
                        MVI  B,(LINCNT/11)+1  ;BETWEEN PAGES.
                NOTEOP: MVI  C,0AH      ;GET LINE-FEED CODE.
                LSTPA1: CALL LIBSY      ;PRINT LINE-FEED.
                        DCR  B          ;DECREMENT LF COUNTER.
                        JNZ  LSTPA1     ;DO NEXT LINE FEED?
                        ENDIF

                        IF LSTNUL OR LSTPAG  ;IF NULLS OR PAGING,
                LINUL2: POP  B          ;RESTORE B&C.
                        MOV  A,C        ;RESTORE A.
                        RET             ;RETURN FROM LIST.
                        ENDIF
                ;
                ; PUNCH PAPER TAPE.
                ;
5DB7 C9         PUNCH:  RET             ;RETURN FROM PUNCH.
                ;
                ;  NORMALLY USED TO READ PAPER TAPE.
                ;
5DB8 C9         READER: RET             ;RETURN FROM READER.
                ;
                ;NOTE:  AS THERE ARE ONLY NINE SECTORS
                ;AVAILABLE FOR CBIOS ON THE SECOND SYSTEM TRACK (1),
                ;THE LAST ADDRESS BEFORE THIS POINT SHOULD BE NO
                ;GREATER THAN THE CBIOS STARTING ADDRESS + 047F (HEX).
                ;THIS WILL NORMALLY BE XE7F (HEX).
                ;
                ; BIOS SCRATCH AREA.
                ;
5DB9            TRK:    DS   1          ;CURRENT TRACK NUMBER.
5DBA            SECT:   DS   1          ;CURRENT SECTOR NUMBER.
5DBB            DMAADD: DS   2          ;DISK TRANSFER ADDRESS.
                ;
                ; THE NEXT SEVERAL BYTES, BETWEEN STARTZ AND
                ; ENDZ, ARE SET TO ZERO AT COLD BOOT TIME.
                ;
                STARTZ:                 ;START OF ZEROED AREA.
5DBD            DISKNO: DS   1          ;DISK NUMBER (TO CP/M).

                        IF   TESTING
                ;
                ; ERROR COUNTS.  THESE LOCATIONS KEEP TRACK OF THE
                ; NUMBER OF ERRRS THAT OCCUR DURING READ, WRITE,
                ; OR SEEK OPERATIONS.  THEY ARE INITIALIZED ONLY
                ; WHEN A COLD-START IS PERFORMED BY THE BOOOTSTRAP.
                ;
                RECNT:  DS   1          ;READ ERROR COUNT.
                WECNT:  DS   1          ;WRITE ERROR COUNT.
                SECNT:  DS   1          ;SEEK ERROR COUNT.
                        ENDIF
                ;
                ; SPECIAL FLAGS.
                ;
5DBE            CONOTF: DS   1          ;NO-PRINT FLAG (WHEN FF).
5DBF            HLSF:   DS   1          ;HEAD-LOAD SELECT FLAG.
```

```
                        IF  LSTPAG
                LFCNT:  DS   1          ;PAGING LINE-FEED COUNT.
                        ENDIF
                ;
                ; TRTAB - DISK TRACK TABLE - PRESENT POSITION OF
                ;       HEADS FOR UP TO 4 DRIVES.
                ;
5DC0            TRTAB:  DS   4
5DC4            DRVFLG: DS   4          ;DRIVE FLAG BYTES FOR 4 DRIVES
5DC8            DENS:   DS   1          ;CURRENT DRIVE FLAG BYTE
                ;
                ; VDM SCRATCH AREA.
                ;
                ENDZ:                   ;END OF ZEROED AREA.
                        IF   VIDEO      ;IF VIDEO BOARD IN,
                VDMP:   DS   2          ;VIDEO CURSOR POSITION.
                        ENDIF

5DC9            NODSKS: DS   1          ;NUMBER OF DISKS.
5DCA            ERCNT:  DS   1          ;ERROR COUNT FOR RETRIES.
5DCB            SERCNT: DS   1          ;SEEK RETRY COUNTER.
5DCC            LATCH:  DS   1          ;NEW CODE FOR LATCH.
5DCD            CLATCH: DS   1          ;CURRENT CODE IN LATCH.
5DCE            DBUFF:  DS   128        ;DENSITY SELECT BUFFER
5E4E                     END

B>
```

```
                        ;
                        ; TARBELL ELECTRONICS CP/M COLDSTART LOADER
                        ; VERSION OF 7-3-80.
                        ;-----------------------------------------------------------
                        ; Modified for DMA Control - 1-5-80.
                        ; Modified for reading larger bios from TRK 1  - 6-5-80.
                        ; Modified for Tarbell CPU Card  -  7-3-80.
                        ; G.W.Mulchin
                        ; Tarbell Electronics
                        ;-----------------------------------------------------------
                        ; Copyright (c) 1980 Tarbell Electronics
                        ;
                        ;** NOTE **
                        ;       The equate for Double Density (DOUBDEN) must only be
                        ; set true for a disk which is formatted in double density only
                        ; and one which you wish to put an operating system on to.
                        ; Otherwise, leave it false if you are building an operating
                        ; system on to a single density formatted disk.
                        ;
                        ; THIS PROGRAM IS LOADED AT LOCATION ZERO BY THE BOOTSTRAP PROGRAM,
                        ; AND EXECUTED. ITS PURPOSE IS TO LOAD AND EXECUTE THE CP/M DISK
                        ; OPERATING SYSTEM AT THE TOP OF THE MEMORY IN USE.
                        ;
0000 =                  FALSE   EQU  0          ;DEFINE VALUE OF FALSE.
FFFF =                  TRUE    EQU  NOT FALSE  ;DEFINE VALUE OF TRUE.
                        ;
                        ;********* THIS IS THE AREA TO MAKE CHANGES IN ******************
                        ;********* FOR DIFFERENT SYSTEM CONFIGURATIONS ******************
                        ;                                                            **
0018 =                  MSIZE   EQU  24         ;MEMORY SIZE IN DECIMAL KB.          **
0000 =                  TARBELL EQU  FALSE      ;TRUE IF USING TARBELL CPU.          **
0000 =                  DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED SYSTEMS.      **
0000 =                  DELTA   EQU  FALSE      ;TRUE IF USING DELTA CPU CARD        **
0000 =                  DOUBDEN:EQU  FALSE      ;TRUE IF DOUB. DEN DISK.             **
0000 =                  DMACNTL EQU  FALSE      ;TRUE IF USING DMA CONTROL           **
0000 =                  BASE    EQU  0          ;TARBELL I/O PORTS (00 or 10 HEX)    **
001A =                  SPT     EQU  26         ;NUMBER OF SECTORS PER TRACK.        **
001A =                  DDS     EQU  26         ;sectors in trk 1 , (Range = 26 to 51)  **
00F8 =                  DISK    EQU  0F8H       ;DISK PORT BASE ADDRESS.             **
                        ;                                                            **
                        ;************************************************************
                        ;
                                IF   TARBELL
                        IO      EQU  BASE       ;i/o ports on tarbell cpu.
                        MMENB   EQU  IO+10      ;memory management enable port.
                        MEMMAG  EQU  BASE+32    ;memory management port.
                                ENDIF

00E0 =                  ADR0    EQU  0E0H       ;DMA ADDRESS PORT.
00E1 =                  WCT0    EQU  0E1H       ;DMA WORD COUNT PORT.
00E8 =                  CMND    EQU  0E8H       ;DMA COMMAND PORT.
00F8 =                  DCOM    EQU  DISK       ;COMMAND PORT.
00F8 =                  DSTAT   EQU  DISK       ;STATUS PORT.
00F9 =                  TRACK   EQU  DISK+1     ;TRACK PORT.
00FA =                  SECT    EQU  DISK+2     ;SECTOR PORT.
00FB =                  DATA    EQU  DISK+3     ;DATA PORT.
00FC =                  WAIT    EQU  DISK+4     ;WAIT PORT.
00FC =                  DCONT   EQU  DISK+4     ;CONTROL PORT.
00FD =                  DMACHK  EQU  DISK+5     ;DMA CHECK PORT.
00FF =                  PANEL   EQU  0FFH       ;front panel machines.
0019 =                  SDS     EQU  25         ;always 25 sectors to read in trk 1.
1C00 =                  CBASE   EQU  (MSIZE-17)*1024
4500 =                  CPMB    EQU  CBASE+2900H;START OF CP/M.
5A00 =                  BOOTE   EQU  CBASE+3E00H;COLD BOOT ENTRY POINT.
0033 =                  NSECTS  EQU  SDS + DDS  ;SECTORS OF CP/M.
000A =                  RTCNT   EQU  10         ;NUMBER OF RETRYS.
                        ;
0000                            ORG  0          ;START OF LOADER.
                        ;
                        BOOT:
                                IF   TARBELL    ;IF USING TARBELL CPU
                                OUT  MMENB      ;ENABLE MEMORY MANAGEMENT.
                                LXI  D,1000H    ;COUNT=16, DATA BYTE = 0
                                MVI  C,MEMMAG AND 0FFH
                        MLOOP:  MOV  A,E        ;GET ADDRESS VALUE
                                ORA  C          ;MAKE I/O PORT VALUE
                                STA  MOUT+1     ;MODIFY PORT ON THE FLY
```

```
                        MOV  A,E        ;GET INIT VALUE.
                        CMA             ;FLIP IT FOR RAM ON CPU
                MOUT:   OUT  BASE       ;PUT IT TO RAM ON CPU
                        INR  E          ;BUMP DATA VALUE
                        DCR  D          ;DECREASE COUNT
                        JNZ  MLOOP      ;LOOP 16 TIMES.
                        ENDIF
                ;
                        IF   DELTA      ;IF USING DELTA CPU.
                        MVI  A,1        ;GET A 1 IN REG A.
                        OUT  9          ; AND DISABLE CPU ROM SLOT.
                        ENDIF
                ;
0000 1E0A               MVI  E,RTCNT    ;GET RETRY COUNT.
0002 310001     BLOOP:  LXI  SP,100H    ;SET STACK POINTER.
0005 210045             LXI  H,CPMB     ;CP/M STARTS HERE.
0008 1633               MVI  D,NSECTS   ;NUMBER OF SECTORS TO READ.
000A 0E02               MVI  C,2        ;SECTOR NUMBER.
000C 0604       RNTRK:  MVI  B,4        ;FOR HEAD LOAD.
000E CD2900     RNSEC:  CALL READ       ;READ FIRST SECTOR.
0011 15                 DCR  D          ;IF DONE,
0012 CA005A             JZ   BOOTE      ;GO TO CP/M.
0015 0600               MVI  B,0        ;FOR NO HEAD LOAD.
0017 0C                 INR  C          ;INCREMENT SECTOR COUNT.
0018 79                 MOV  A,C        ;DONE WITH
0019 FE1B       SECCMP: CPI  SPT+1      ;THIS TRACK?
001B DA0E00             JC   RNSEC      ;IF NOT, READ NEXT SECTOR.
                ;
                        IF   DOUBDEN AND NOT DUBSID
                        MVI  A,DDS + 1  ;number of sectors to read on trk 2.
                        STA  SECCMP+1   ;modify sector compare value.
                        MVI  A,8        ;GET SET DOUBLE DENSITY CODE
                        OUT  WAIT       ;SET LATCH FOR D.DENSITY
                        ENDIF
                ;
                        IF   NOT DUBSID
001E 3E5B               MVI  A,5BH      ;STEP COMMAND.
0020 D3F8               OUT  DCOM       ;ISSUE IT.
0022 DBFC               IN   WAIT       ;WAIT UNTIL DONE.
                        ENDIF
                ;
                        IF   DUBSID     ;IF DOUBLE SIDED SYSTEM.
                        MVI  A,40H      ;SIDE SELECT COMMAND.
                        OUT  DCONT      ;ISSUE IT.
                        ENDIF
                ;
0024 0E01               MVI  C,1        ;SECTOR NUMBER.
0026 C30C00             JMP  RNTRK      ;READ NEXT TRACK.
                ;
                READ:
                        IF   DMACNTL    ;IF USING DMA CONTROL.
                        MVI  A,41H      ;SET UP FOR CHAN 0 REQ.
                        OUT  CMND
                        MVI  A,7FH      ;COUNT FOR 128 BYTES
                        OUT  WCT0
                        MVI  A,40H      ;READ COMMAND
                        OUT  WCT0
                        MOV  A,L        ;GET LOW ADDRESS BYTE
                        OUT  ADR0
                        MOV  A,H        ;HIGH ADDRESS BYTE
                        OUT  ADR0
                        ENDIF
                ;
0029 79                 MOV  A,C        ;SECTOR IN A.
002A D3FA               OUT  SECT       ;SET SECTOR REGISTER.
002C 3E88               MVI  A,88H      ;COMMAND FOR READ.
002E B0                 ORA  B          ;GET HEAD LOAD BIT.
002F D3F8               OUT  DCOM       ;ISSUE COMMAND.
                ;
                        IF   NOT DMACNTL;IF NOT USING DMA CONTROL.
0031 DBFC       RLOOP:  IN   WAIT       ;WAIT FOR DRQ.
0033 B7                 ORA  A          ;SET FLAGS.
0034 F23E00             JP   CHECK      ;JUMP IF DONE.
0037 DBFB               IN   DATA       ;READ DATA.
0039 77                 MOV  M,A        ;PUT IN MEMORY.
003A 23                 INX  H          ;INCREMENT POINTER.
003B C33100             JMP  RLOOP      ;LOOP UNTIL DONE.
                        ENDIF
                ;
                        IF  DMACNTL
```

4-21-81

2ABIOS24.PRN STD DD

```
                ;-----------------------------------------------
                ; CP/M BASIC INPUT/OUTPUT OPERATING SYSTEM (BIOS)
                ; TARBELL ELECTRONICS
                ; 2.X VERSION OF 11-4-80
                ; Copyright (c) 1980 Tarbell Electronics
                ;-----------------------------------------------
                ; This bios module is the CPM V2.X Auto Select Bios.
                ; This bios reads single or double density disk.
                ; The Double density disk contains 51 sectors/track,
                ; 77 tracks. Track 0 = single density, Tracks 1 - 76
                ; are double density at 51 sectors per track.
                ; Note:  If you leave DMACNTL false, you must have a CPU
                ; which runs at 4 MHz to run double density.
                ; This bios now supports double sided single/double density.
                ; THIS SECTION DEFINES THE I/O PORTS AND STATUS BITS.
                ; BY SETTING THE PROPER VALUES FOR THE EQU STATEMENTS,
                ; THE I/O MAY BE AUTOMATICALLY RECONFIGURED TO FIT MOST
                ; SITUATIONS. THE TRUE AND FALSE ONES CONTROL CONDITIONAL
                ; ASSEMBLIES OF DIFFERENT SECTIONS OF I/O ROUTINES TO FIT
                ; DIFFERENT INTERFACE REQUIREMENTS.
                ;
FFFF =          TRUE    EQU  0FFFFH             ;DEFINE VALUE OF TRUE.
0000 =          FALSE   EQU  NOT TRUE           ;DEFINE VALUE OF FALSE.
                ;
                ;*************************************************
                ;*** THIS BEGINS THE AREA WHICH REQUIRES CHANGES ***
                ;***      FOR DIFFERENT CONSOLE I/O SYSTEMS      ***
                ;*************************************************
                ;
0018 =          MSIZE   EQU  24         ;MEMORY SIZE IN KBYTES.
0000 =          INTRP   EQU  FALSE      ;TRUE IF INTERRUPTS ALLOWED.
0000 =          TARBELL EQU  FALSE      ;TRUE IF USING THE TARBELL Z-80 CPU.
0000 =          IOBASE  EQU  0          ;BASE IO ADDR FOR TARBELL CPU (0 or 10 hex).
0000 =          TIMER   EQU  FALSE      ;TRUE IF USING CPU TIMER (Tarbell CPU board).
FFFF =          STD     EQU  TRUE       ;TRUE IF STANDARD I/O.
0000 =          MSIO2   EQU  FALSE      ;TRUE IF MITS 2SIO.
0000 =          VDB8024 EQU  FALSE      ;TRUE IF USING VDB-8024 BOARD.
0000 =          DELTA   EQU  FALSE      ;TRUE IF USING DELTA PRODUCTS CPU.
0000 =          ISIO2   EQU  FALSE      ;TRUE IF IMSAI SIO-2.
0000 =          TUART   EQU  FALSE      ;TRUE IF CROMEMCO TUART.
0000 =          VIDEO   EQU  FALSE      ;TRUE IF USING A MEMORY MAPPED VIDEO.
0000 =          OTHER   EQU  FALSE      ;TRUE IF SOMETHING ELSE.
0000 =          SOLOS   EQU  FALSE      ;TRUE IF PROC TECH SOLOS.
0000 =          DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED DRIVES (1 logical drive).
0000 =          DMACNTL EQU  FALSE      ;TRUE IF USING DMA CONTROL.
0004 =          NDISK   EQU  4          ;DEFINES THE NUMBER DRIVES IN SYSTEM.
                ;
                        IF  VIDEO       ;IF USING A VIDEO BOARD
                OUTADDR EQU  0          ;PUT OUTPUT ADDRESS HERE
                        ENDIF
                ;
                        IF NOT SOLOS AND NOT TARDEL     ;IF NOT PROC TECH SOLOS,
0000 =          CSTAT   EQU  0          ;CONSOLE STATUS PORT.
0000 =          CCOM    EQU  0          ;CONSOLE COMMAND PORT.
0001 =          CDATA   EQU  1          ;CONSOLE DATA PORT.
0002 =          LSTAT   EQU  2          ;LIST STATUS PORT.
0002 =          LCOM    EQU  2          ;LIST COMMAND PORT.
0003 =          LDATA   EQU  3          ;LIST DATA PORT.
                        ENDIF
```

```
0000 =          CONUL   EQU  FALSE       ;CONSOLE NULLS?
0010 =          CNULL   EQU  16          ;CONSOLE NULL COUNT.
0000 =          LSTNUL  EQU  FALSE       ;LIST DEVICE NULLS?
0000 =          LNULL   EQU  0           ;LIST NULL COUNT.
0000 =          LSTPAG  EQU  FALSE       ;LIST DEVICE PAGING?
0042 =          LINCNT  EQU  66          ;LINES PER PAGE.
0000 =          HLAB    EQU  0           ;8 FOR HD LD AT BEG OF SEEK.
0002 =          STPRAT  EQU  2           ;RATE 0=3ms,1=6MS, 2=10MS, 3=20MS.
0000 =          DUAL    EQU  FALSE       ;TRUE IF DUAL HEADED (2 HEADS MOVING TOGETHER).
                ;
                ;*********************************************************
                ;*** THIS IS THE END OF THE AREA WHICH NORMALLY NEED ***
                ;***     BE CHANGED FOR MOST CONSOLE I/O SYSTEMS     ***
                ;*********************************************************
                ;
FFFF =          RDYLO   EQU  STD OR SOLOS OR OTHER  ;STATUS READY WHEN LOW.
0000 =          RDYHI   EQU  NOT RDYLO
0000 =          TARDEL  EQU  TARBELL OR DELTA   ;IF USING TARBELL OR DELTA CPU.
                ;
                        IF   TARDEL      ;IF USING TARBELL OR DELTA CPU
                CCOM    EQU  IOBASE+1    ;CONSOLE COMMAND PORT
                CSTAT   EQU  IOBASE+1    ;CONSOLE STATUS PORT ( CHAN A.)
                CDATA   EQU  IOBASE+0    ;CONSOLE DATA PORT
                LCOM    EQU  IOBASE+3    ;LIST COMMAND PORT
                LSTAT   EQU  IOBASE+3    ;LIST STATUS PORT (CHAN B.)
                LDATA   EQU  IOBASE+2    ;LIST DATA PORT
                        ENDIF
                ;
                        IF      TIMER AND TARBELL       ;MUST BE USING TARBELL CPU.
                ;
                ; TIMER EQUATES
                ;
                TCH0    EQU     IOBASE+4        ;TIMER CHAN 0 ADDRESS
                TCH1    EQU     IOBASE+5        ;TIMER CHAN 1 ADDRESS
                TCH2    EQU     IOBASE+6        ;TIMER CHAN 2 ADDRESS
                TCMND   EQU     IOBASE+7        ;TIMER COMMAND PORT
                IMASK   EQU     IOBASE+8        ;INTERRUPT MASKING PORT
                CNTR0   EQU     00000000B       ;counter 0
                CNTR1   EQU     01000000B       ;counter 1
                CNTR2   EQU     10000000B       ;counter 2
                RLWORD  EQU     00110000B       ;read/load lsb 1st, msb 2nd.
                RLHBYTE EQU     00100000B       ;read/load msb only.
                RLLBYTE EQU     00010000B       ;read/load lsb only.
                CNTRLT  EQU     00000000B       ;counter latching operation.
                BINARY  EQU     00000000B       ;select binary operation.
                BCD     EQU     00000001B       ;select BCD operation.
                MODE0   EQU     00000000B       ;interrupt on terminal count.
                MODE1   EQU     00000010B       ;programmable One-shot.
                MODE2   EQU     00000100B       ;rate generator.
                MODE3   EQU     00000110B       ;square wave rate generator.
                MODE4   EQU     00001000B       ;software triggered strobe.
                MODE5   EQU     00001010B       ;hardware triggered strobe.
                        ENDIF
                ;
                        IF   SOLOS               ;IF PROC TECH SOLOS,
                CSTAT   EQU  0FAH                ;CONSOLE STATUS PORT.
                KBD     EQU  0C02EH              ;SOLOS KEYBOARD.
                CLRSCR  EQU  0C0D5H              ;CLEAR SCREEN.
                SCRN    EQU  0C054H              ;SOLOS OUTPUT.
                        ENDIF
                ;
                        IF NOT SOLOS             ;IF NOT PROC TECH SOLOS,
```

```
00E0 =          DMAP    EQU  0E0H          ;DMA BASE ADDRESS.
00F8 =          DISK    EQU  0F8H          ;DISK BASE ADDRESS.
                ENDIF
                ;
                IF SOLOS                   ;IF PROC TECH SOLOS,
                DMAP    EQU  060H          ;DMA BASE ADDRESS.
                DISK    EQU  078H          ;DIFFERENT DISK PORTS.
                ENDIF
                ;
00E0 =          ADR0    EQU  DMAP+0        ;DMA ADDRESS REG PORT.
00E1 =          WCT0    EQU  DMAP+1        ;DMA WORD COUNT REG PORT.
00E8 =          CMND    EQU  DMAP+8        ;DMA COMMAND PORT.
00F8 =          DCOM    EQU  DISK          ;DISK COMMAND PORT.
00F8 =          DSTAT   EQU  DISK          ;DISK STATUS PORT.
00F9 =          TRACK   EQU  DISK+1        ;DISK TRACK PORT.
00FA =          SECTP   EQU  DISK+2        ;DISK SECTOR PORT.
00FB =          DDATA   EQU  DISK+3        ;DISK DATA PORT.
00FC =          WAIT    EQU  DISK+4        ;DISK WAIT PORT.
00FC =          DCONT   EQU  DISK+4        ;DISK CONTROL PORT.
00FD =          DMACHK  EQU  DISK+5        ;DMA CHECK PORT.
000A =          RTCNT   EQU  10            ;RETRY COUNT.
                ;
                IF   STD                   ;IF STANDARD I/O,
0001 =          CKBR    EQU  00000001B     ;KEYBOARD READY BIT.
0080 =          CPTR    EQU  10000000B     ;CONS OUTPUT RDY BIT.
                ENDIF
                ;
                IF   MSIO2                 ;IF MITS 2SIO,
                CKBR    EQU  00000001B     ;KEYBOARD READY BIT.
                CPTR    EQU  00000010B     ;PRINT READY BIT.
                ENDIF
                ;
                IF   VDB8024               ;IF VDB-8024 BOARD.
                CKBR    EQU  00000010B     ;KEYBOARD READY BIT.
                CPTR    EQU  00000100B     ;CONS OUTPUT RDY BIT.
                ENDIF
                ;
                IF   ISIO2
                CKBR    EQU  00000010B     ;KEYBOARD READY BIT.
                CPTR    EQU  00000001B     ;PRINT READY BIT.
                ENDIF
                ;
                IF   TARDEL
                CKBR    EQU  00000010B     ;KEYBOARD READY BIT.
                CPTR    EQU  00000001B     ;PRINT READY BIT.
                ENDIF
                ;
                IF   TUART                 ;IF CROMEMCO TUART,
                CKBR    EQU  01000000B     ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B     ;PRINT READY BIT.
                ENDIF
                ;
                IF   SOLOS                 ;IF PROC TECH SOLOS,
                CKBR    EQU  00000001B     ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B     ;DUMMY EQU.
                ENDIF
                ;
                IF   OTHER                 ;IF SOMETHING ELSE,
                CKBR    EQU  00000010B     ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B     ;PRINTER READY BIT.
                ENDIF
                ;
```

```
                        CALL CONIN              ;READ KEYBOARD.
                        CALL CONIN              ;READ KEYBOARD AGAIN.
                        CPI  0DH                ;IF NOT CARRIAGE-RETURN,
                        MVI  A,1                ;SLOW THE CLOCK.
                        JNZ  IT1                ;UNTIL A CARRIAGE-RETURN.
                        ENDIF
                ;
                        IF   SOLOS              ;IF PROC TECH SOLOS,
                        CALL CLRSCR             ;CLEAR SCREEN.
                        ENDIF
                ;
                        IF DMACNTL
                        LXI  H,RWDMA            ;POINT TO DMA ROUTINE
                        SHLD DMAENT+1           ;MODIFY BOOT JMP ADDRESS.
                        ENDIF
                ;
                        IF   TIMER AND TARBELL  ;IF USING TARBELL CPU
                        MVI  A,CNTR0+RLWORD+MODE2+BINARY        ;INIT 8253
                        OUT  TCMND              ;SEND IT TO COMMAND PORT
                        LXI  B,33333            ;TIME CONSTANT FOR 60 HZ
                        MOV  A,C
                        OUT  TCH0               ;LS BYTE OF COUNT
                        MOV  A,B
                        OUT  TCH0               ;MS BYTE OF COUNT
                        ENDIF
                ;
443D C3C059             JMP  BOOTF              ;FINISH BOOT
                ;
                        IF   TARDEL OR ISIO2
                IOINIT: DB   0AAH,040H,0CEH,037H
                        ENDIF
                ;
                        IF   TUART              ;IF CROMEMCO TUART,
                BAUDRS: DB   94H,0CEH,0A2H,92H,88H,84H,82H,1
                        ENDIF
                ;
59C0                    ORG  BIOS-64            ;HIDE REST OF BOOT HERE.
59C0 AF         BOOTF:  XRA  A                  ;CLEAR SCRATCH AREA.
59C1 320300             STA  IOBYTE             ;CLEAR I/O BYTE.
59C4 320400             STA  CDISK              ;SELECT DRIVE ZERO
59C7 0611               MVI  B,ENDZ-STARTZ      ;GET LENGTH OF ZERO AREA.
59C9 21195D             LXI  H,STARTZ           ;GET SCRATCH ADDRESS.
59CC 77         BOOTL:  MOV  M,A                ;PUT ZERO IN MEMORY.
59CD 23                 INX  H                  ;INCREMENT POINTER.
59CE 05                 DCR  B                  ;DECREMENT COUNTER.
59CF C2CC59             JNZ  BOOTL              ;LOOP TILL DONE.
59D2 DB01               IN   CDATA              ;CLEAR CONSOLE STATUS.
59D4 210844             LXI  H,SMSG             ;POINT TO SIGN ON.
59D7 7E         PMSG:   MOV  A,M                ;GET A BYTE OF THE MESSAGE
59D8 23                 INX  H                  ;BUMP MEMORY POINTER.
59D9 B7                 ORA  A                  ;IS IT A ZERO?
59DA CADF5A             JZ   GOCPM              ;YES, WE ARE DONE, JMP TO CPM.
59DD 4F                 MOV  C,A                ;NOPE, PRINT MORE MESSAGE.
59DE CD1E5B             CALL CONOT              ;USE THE CONOT ROUTINE.
59E1 C3D759             JMP  PMSG               ;AND LOOP TILL DONE.
                ;
5A00                    ORG  BIOS               ;START OF CBIOS STRUCTURE.
                ;
                ; I/O JUMP VECTOR
                ; THIS IS WHERE CPM CALLS WHENEVER IT NEEDS TO DO ANY INPUT/OUTPUT
                ; OPERATION. USER PROGRAMS MAY USE THESE ENTRY POINTS ALSO, BUT NOTE
                ; THAT THE LOCATION OF THIS VECTOR CHANGES WITH THE MEMORY SIZE.
```

```
                        ;
5A00 C33A44     DMAENT: JMP   BOOT              ;FROM SBOOT LOADER,CHANGED FOR DMA.
5A03 C3B15A     WBOOTE: JMP   WBOOT             ;FROM WARM BOOT.
5A06 C3065B             JMP   CONST             ;CHECK CONSOLE KB STATUS.
5A09 C3135B             JMP   CONIN             ;READ CONSOLE CHARACTER.
5A0C C31E5B             JMP   CONOT             ;WRITE CONSOLE CHARACTER.
5A0F C30B5D             JMP   LIST              ;WRITE LISTING CHAR.
5A12 C3055D             JMP   PUNCH             ;WRITE PUNCH CHAR.
5A15 C3055D             JMP   READER            ;READ READER CHAR.
5A18 C3B65B             JMP   HOME              ;MOVE DISK TO TRACK ZERO.
5A1B C3295B             JMP   SELDSK            ;SELECT DISK DRIVE.
5A1E C3BC5B             JMP   SETTRK            ;SEEK TO TRACK IN REG A.
5A21 C32B5C             JMP   SETSEC            ;SET SECTOR NUMBER.
5A24 C33B5C             JMP   SETDMA            ;SET DISK STARTING ADR.
5A27 C35F5C             JMP   READ              ;READ SELECTED SECTOR.
5A2A C3D65C             JMP   WRITE             ;WRITE SELECTED SECTOR.
5A2D C3FE5C             JMP   PRSTAT            ;LIST STATUS CHECK.
5A30 C3305C             JMP   SECTRAN           ;SECTOR TRANSLATE ROUTINE.
                        ;
                        ; THIS SECTION DEFINES THE THE DISK PARAMETERS
                        ;
5A33 =          DPBASE  EQU   $                 ;BASE OF DISK PARAMETER BLOCK
5A33 975A0000   DPE0:   DW    XLT0,0000H        ;TRANSLATE TABLE
5A37 00000000           DW    0000H,0000H       ;SCRATCH AREA
5A3B 2A5D765A           DW    DIRBUF,SDTAB+3    ;DIR BUFF, PARM BLOCK
5A3F C95DAA5D           DW    CSV0,ALV0         ;CHECK, ALLOC VECTORS
                        ;
5A43 975A0000   DPE1:   DW    XLT1,0000H
5A47 00000000           DW    0000H,0000H
5A4B 2A5D765A           DW    DIRBUF,DPB1
5A4F 005EE15D           DW    CSV1,ALV1
                        ;
5A53 975A0000   DPE2:   DW    XLT2,0000H
5A57 00000000           DW    0000H,0000H
5A5B 2A5D765A           DW    DIRBUF,DPB2
5A5F 375E185E           DW    CSV2,ALV2
                        ;
5A63 975A0000   DPE3:   DW    XLT3,0000H
5A67 00000000           DW    0000H,0000H
5A6B 2A5D765A           DW    DIRBUF,DPB3
5A6F 6E5E4F5E           DW    CSV3,ALV3
                        ;
                        ;THE FOLLOWING DESCRIBES THE DISK PHYSICAL NATURE, SUCH AS
                        ;SECTORS/TRACK,DIRECTORY SIZE, ETC...
                        ;THE FOLLOWING TABLE DEFINES A SINGLE DENSITY DRIVE.
                        ;
5A73 =          SDTAB:  EQU   $                 ;ONE OF 4 DISK PARM. BLOCKS
5A73 00                 DB    00H               ;LOG BYTE SINGLE DENSITY
5A74 975A               DW    XLT0              ;USE SINGLE DENSITY TRANSLATE TAB.
5A76 1A00               DW    26                ;SECTORS/TRACK
5A78 03                 DB    3                 ;BLOCK SHIFT
5A79 07                 DB    7                 ;BLOCK MASK
5A7A 00                 DB    0                 ;EXTNT MASK
5A7B F200               DW    242               ;DISK SIZE - 1
5A7D 3F00               DW    63                ;DIRECTORY MAX.
5A7F C0                 DB    192               ;ALLOC0
5A80 00                 DB    0                 ;ALLOC1
5A81 1000               DW    16                ;CHECK SIZE
5A83 0200               DW    2                 ;NUMBER OF SYSTEM TRACKS
                        ;
                        IF    DUBSID            ;is using double sided drives.
                        ;
```

```
                ; Defines a Single density/ Double sided disk
                ;
                        DB   02H                ;log byte doub sided
                        DW   XLT0
                        DW   26
                        DB   4
                        DB   15
                        DB   0
                        DW   242
                        DW   95                 ;allow 95 entrys for dir.
                        DB   192
                        DB   0
                        DW   24
                        DW   2
                        ENDIF
                ;
                ;THE FOLLOWING TABLE DEFINES A DOUBLE DENSITY DRIVE.
                ;
5A85 01         DDTAB:  DB   01H                ;log byte doub den/sing sided
5A86 0000               DW   0                  ;NO SECTOR TRANSLATE TABLE.
5A88 3300               DW   51                 ;51 SECTORS.
5A8A 04                 DB   4                  ;BLOCK SHIFT.
5A8B 0F                 DB   15                 ;BLOCK MASK.
5A8C 00                 DB   0                  ;EXTENT MASK.
5A8D ED00               DW   237                ;DISK SIZE -1
5A8F 5F00               DW   95                 ;DIRECTORY MAX.
5A91 C0                 DB   192                ;ALLOC0
5A92 00                 DB   0                  ;ALLOC1
5A93 1800               DW   24                 ;CHECK SIZE
5A95 0200               DW   2                  ;NUMBER OF SYSTEM TRACKS.
                ;
                        IF   DUBSID             ;if using double sided drives.
                ;
                ; Defines a Double density/Doub sided drive
                ;
                        DB   03H                ;log byte and dub sided
                        DW   0
                        DW   51
                        DB   5
                        DB   31
                        DB   0
                        DW   237
                        DW   95
                        DB   192
                        DB   0
                        DW   24
                        DW   2
                        ENDIF
                ;
                ;SECTOR TRANSLATION TABLE
                ;
5A97 =          XLT0    EQU  $                  ;START OF TRANS. TABLE
5A97 01070D1319         DB   1,7,13,19,25
5A9C 050B111703         DB   5,11,17,23,3
5AA1 090F150208         DB   9,15,21,2,8
5AA6 0E141A060C         DB   14,20,26,6,12
5AAB 1218040A10         DB   18,24,4,10,16,22
                ;
5A76 =          DPB1    EQU  SDTAB+3            ;EQUIVALENT PARAMETERS
5A97 =          XLT1    EQU  XLT0               ;SAME TRANSLATE TABLE
                ;
5A76 =          DPB2    EQU  SDTAB+3
```

```
                        MOV  D,A                ;NOW SET FOR H,L ADJUST.
                        DAD  D                  ;ADD IT TO H,L.
                        POP  D                  ;RESTORE D,E
                ;
                ;GENERAL PURPOSE WAIT ROUTINE.
                ;
                        MVI  A,20H              ;COUNT VALUE.
                CNTLOOP:DCR  A
                        JNZ  CNTLOOP            ;LOOP TILL = ZERO.
                SLOOP:  IN   DMACHK             ;CHECK FOR OPERATION DONE.
                        RLC                     ;BY LOOKING AT BIT 7.
                        JC   SLOOP              ;LOOP TILL BIT 7 = 0.
                        IN   DSTAT              ;CHECK AND RETURN DISK STATUS.
                        RET                     ;RETURN TO CALLER.
                        ENDIF
                ;
                ;Warm-boot - Read the CCP back into memory. BDOS and BIOS
                ;assumed still in memory. If they are not, a cold start will
                ;have to be done to bring them back into memory.
                ;
5AB1 318000     WBOOT:  LXI  SP,80H             ;SET STACK POINTER.
                ;
                        IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
                        EI                      ;ALLOW THEM HERE.
                        ENDIF
                ;
                        IF   LSTPAG             ;IF LIST DEVICE PAGING,
                        XRA  A                  ;RESET LINE-FEED COUNT.
                        STA  LPCNT
                        ENDIF
                ;
5AB4 0E00               MVI  C,0                ;SELECT DISK 0.
5AB6 CD295B             CALL SELDSK
5AB9 CDB65B             CALL HOME               ;MOVE TO TRACK ZERO.
5ABC 210000             LXI  H,0                ;clear h,l
5ABF 22215D             SHLD DRVFLG             ;clear drive flags
5AC2 22235D             SHLD DRVFLG+2
5AC5 0611               MVI  B,NSECTS           ;GET # SECTORS FOR CPM READ.
5AC7 0E02               MVI  C,2                ;TRACK (B)=0, SECTOR (C)=2.
                ;
                        IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
                        DI                      ;DISABLE THEM HERE.
                        ENDIF
                ;
5AC9 210044             LXI  H,CPMB             ;GET STARTING ADDRESS.
5ACC 22175D     RBLK1:  SHLD DMAADD             ;SET STARTING ADDRESS.
5ACF CD2B5C             CALL SETSEC             ;READ STARTING AT SECTOR IN C.
5AD2 C5                 PUSH B
5AD3 CD5F5C             CALL READ               ;READ A SECTOR BACK.
5AD6 C1                 POP  B
5AD7 C2005B             JNZ  RDERR              ;IF ERROR, PRINT MESSAGE.
5ADA 0C                 INR  C                  ;INCREMENT SECTOR NUMBER.
5ADB 05                 DCR  B                  ;DECREMENT SECTOR COUNT.
5ADC C2CC5A             JNZ  RBLK1              ;NOT ZERO, KEEP READING
                ;
                        IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
                        EI                      ;ALLOW THEM AGAIN HERE.
                        ENDIF
                ;
                ; SET UP JUMPS INTO CP/M IN LOWER MEMORY.
                ;
5ADF 3EC3       GOCPM:  MVI  A,0C3H             ;PUT JMP TO WBOOT
```

```
5A97 =          XLT2    EQU  XLT0
                ;
5A76 =          DPB3    EQU  SDTAB+3
5A97 =          XLT3    EQU  XLT0
                ;
                ;DISK SET UP ROUTINE. THIS ROUTINE IS COMMON TO BOTH THE
                ;READ AND WRITE ROUTINES FOR DMA OPERATION. THIS ROUTINE
                ;MAY BE USED STAND ALONE BY PASSING PARAMETERS TO IT AND
                ;JUMPING TO WBOOT-3 HEX. THIS JUMP VECTOR IS CHANGED WHEN
                ;CP/M IS BOOTED UP.
                ;
                ;ENTERY POINT = RWDMA:
                ;
                ;USER MUST SET UP DMAADD FOR MEMORY ADDRESS AND
                ;USER MUST SET UP DISK SECTOR WITH 'SETSEC' ENTRY
                ;THE TRACK TO READ OR WRITE MUST BE SET UP USING 'SETTRK'
                ;BEFORE USING RWDMA ROUTINE EXTERNALLY.
                ;
                ;ENTRY PARAMETERS:
                ;B    = FLOPPY DISK (1793) READ/WRITE COMMAND BYTE
                ;C    = FLOPPY DISK (1793) FORCE INTERRUPT COMMAND BYTE
                ;D    = DMA (8257) READ/WRITE COMMAND + HIGH BYTE COUNT
                ;E    = DMA (8257) LOW BYTE COUNT (80 HEX = 128 BYTES)
                ;
                ;EXIT VALUES
                ;B,C = FLOPPY COMMANDS
                ;D,E = DMA COMMAND + BYTE COUNT.
                ;H,L = (H,L + D,E)
                ;A    = FLOPPY DISK STATUS BYTE
                ;
                ;STACK USAGE IS 1 LEVEL DEEP.
                ;
                        IF DMACNTL              ;IF USING DMA CONTROL
                ;
                DMARW:  STA  ERCNT              ;SAVE ERROR COUNT.
                RWDMA:  LDA  SECT               ;GET SECTOR TO READ/WRITE
                        OUT  SECTP              ;AND SEND IT FLOPPY CHIP.
                        LHLD DMAADD             ;GET CPM DMA ADDRESS.
                DMARWE: XRA  A                  ;CLEAR ACCUM.
                        OUT  CMND               ;RESET DMA CHIP.
                        MOV  A,C                ;FORCE INTERRUPT COMMAND BYTE
                        OUT  DCOM               ;SEND IT TO CONTROLLER.
                        MOV  A,E                ;BYTE COUNT TO TRANSFER
                        DCR  A                  ;COUNT = COUNT - 1.
                        OUT  WCT0               ;SEND IT TO DMA CHIP.
                        MOV  A,D                ;GET READ/WRITE CODE.
                        OUT  WCT0               ;AND TELL DMA CHIP WHAT TO DO.
                        MOV  A,L                ;GET LOW ADDRESS BYTE
                        OUT  ADR0               ;AND SEND IT TO DMA CHIP.
                        MOV  A,H                ;GET HIGH ADDRESS BYTE
                        OUT  ADR0               ;AND SEND IT TO DMA CHIP.
                        MVI  A,41H              ;SET UP FOR REQUEST CH. 0
                        OUT  CMND               ;SEND IT TO CONTROLLER.
                        CALL HDLD               ;CHECK HEAD LOAD BIT.
                        ORA  B                  ;'OR' IN THE READ/WRITE BITS.
                        OUT  DCOM               ;TELL FLOPPY CHIP WHAT TO DO.
                ;
                ;ADJUST H,L FOR 128 BYTE INCREASE.
                ;
                        PUSH D                  ;SAVE D,E
                        MOV  A,D                ;GET DMA COMMAND BYTE
                        ANI  3FH                ;STRIP OFF COMMAND BITS 7 & 6
```

```
5AE1 320000             STA  0              ;ADR AT ZERO.
5AE4 21035A             LXI  H,WBOOTE       ;WARMBOOT ENTRY POINT
5AE7 220100             SHLD 1              ;SET IT.
5AEA 320500             STA  5              ;SET JUMP INSTRUCTION.
5AED 21064C             LXI  H,BDOS         ;PUT JUMP TO BDOS
5AF0 220600             SHLD 6              ;AT ADR 5,6,7.
5AF3 218000             LXI  H,80H          ;SET DEFAULT DMA ADR.
5AF6 22175D             SHLD DMAADD         ;SAVE IT.
5AF9 3A0400             LDA  CDISK          ;GET DISK NUMBER TO
5AFC 4F                 MOV  C,A            ;PASS TO CCP IN C.
5AFD C30044             JMP  CPMB           ;JUMP TO CCP.
                ;
5B00 CD875C     RDERR:  CALL RECOV          ;We have an error in booting.
5B03 C3B15A             JMP  WBOOT          ;DO A WARM BOOT.
                ;
                ; CHECK CONSOLE INPUT STATUS.
                ;
5B06 CD0E5B     CONST:  CALL STATCON        ;CHECK CONSOLE STATUS PORT.
5B09 3E00       CONST1: MVI  A,0            ;SET A=0 FOR RETURN.
                ;
                        IF   RDYLO          ;IF STATUS READY LOW,
5B0B C0                 RNZ                 ;NOT READY WHEN NOT 0.
                        ENDIF
                ;
                        IF   RDYHI          ;IF STATUS READY HIGH,
                        RZ                  ;NOT READY WHEN ZERO.
                        ENDIF
                ;
5B0C 2F                 CMA                 ;IF READY A=FF.
5B0D C9                 RET                 ;RETURN FROM CONST.
                ;
                ; STATCON - CHECK KEYBOARD STATUS
                ;
                        IF   NOT SOLOS
5B0E DB00       STATCON:IN   CSTAT          ;IN STATUS PORT
5B10 E601               ANI  CKBR           ;MASK READY BIT.
5B12 C9                 RET
                        ENDIF
                ;
                ; READ A CHARACTER FROM CONSOLE.
                ;
                CONIN:
                        IF   NOT SOLOS      ;IF NOT PROC TECH SOLOS,
5B13 CD0E5B             CALL STATCON        ;READ CONSOLE STATUS.
                        ENDIF
                ;
                        IF   SOLOS          ;IF PROC TECH SOLOS,
                        CALL KBD            ;READ SOL KEYBOARD.
                        JZ   CONIN          ;READY WHEN NOT ZERO.
                        ENDIF
                ;
                        IF   RDYLO AND NOT SOLOS
5B16 C2135B             JNZ  CONIN          ;LOOP UNTIL LOW.
                        ENDIF
                ;
                        IF   RDYHI          ;IF READY WHEN HIGH,
                        JZ   CONIN          ;LOOP UNTIL HIGH.
                        ENDIF
                ;
                        IF   NOT SOLOS      ;IF NOT PROC TECH SOLOS,
5B19 DB01               IN   CDATA          ;READ A CHARACTER.
                        ENDIF
```

```
                ;
5B1B E67F               ANI  7FH                ;MAKE MOST SIG. BIT = 0.
5B1D C9                 RET                     ;RETURN FROM CONIN.
                ;
                ; WRITE A CHARACTER TO THE CONSOLE DEVICE.
                ;
                CONOT:
                        IF CONUL                ;IF NULLS REQUIRED,
                        MVI  A,0AH              ;IF IT'S A LF,
                        CMP  C                  ;THEN HOP OUT
                        JZ   CONULL             ;TO NULL ROUTINE.
                        ENDIF
                CONOT1:
                        IF NOT SOLOS AND NOT VIDEO
5B1E DB00               IN   CSTAT              ;READ CONSOLE STATUS.
5B20 E680               ANI  CPTR               ;IF NOT READY,
                        ENDIF
                ;
                        IF RDYLO AND NOT SOLOS AND NOT VIDEO
5B22 C21E5B             JNZ  CONOT1             ;LOOP UNTIL LOW.
                        ENDIF
                ;
                        IF RDYHI                ;IF READY WHEN HIGH,
                        JZ   CONOT1             ;LOOP UNTIL HIGH.
                        ENDIF
                ;
                        IF NOT SOLOS AND NOT VIDEO
5B25 79                 MOV  A,C                ;GET CHARACTER.
5B26 D301               OUT  CDATA              ;PRINT IT.
5B28 C9                 RET                     ;RETURN.
                        ENDIF
                ;
                ;THIS ROUTINE CALLES YOUR VIDEO DRIVER ROUTINE WHICH MUST
                ;BE IN ROM. ALL REGISTERS MUST BE SAVED AND RESTORED BY YOUR
                ;VIDEO DRIVER IN ORDER TO BE COMPATIABLE WITH CPM. CPM PASSES
                ;THE CHAR. TO BE OUTPUT IN THE  C REGISTER. MAKE ANY CHANGES
                ;IN THIS ROUTINE TO PASS THE CHAR FROM REG C TO THE REGISTER
                ;YOUR VIDEO DRIVER EXPECTS IT TO BE IN.
                ;
                        IF  VIDEO               ;IF USING A VIDEO DRIVER IN ROM.
                        MOV  A,C                ;GET THE CPM CHAR INTO REG A
                        CALL CUTADDR            ;CALL YOUR VIDEO DRIVER.
                        RET                     ;RETURN TO CPM.
                        ENDIF
                ;
                        IF CONUL
                CONULL: PUSH B                  ;SAVE B&C.
                        MVI  B,CNULL+1          ;GET NULL COUNT.
                CONUL1: CALL CONOT1             ;PRINT CR.
                        MVI  C,0                ;GET NULL CHAR.
                        DCR  B                  ;DECREMENT COUNTER.
                        JNZ  CONUL1             ;DO NEXT NULL.
                        POP  B                  ;RESTORE B&C.
                        MOV  A,C                ;RESTORE A.
                        RET                     ;RETURN.
                        ENDIF
                ;
                        IF   SOLOS              ;IF PROC TECH SOLOS,
                        PUSH B                  ;SAVE B&C.
                        MOV  B,C                ;PUT CHAR IN B REG.
                        CALL SCRN               ;OUTPUT CHAR TO SOLOS.
                        POP  B                  ;RESTORE B&C.
```

```
                        MOV  A,C                ;PUT CHAR IN A.
                        RET                     ;RETURN FROM CONOT.
                        ENDIF
                ;
                ; SELECT DISK NUMBER ACCORDING TO REGISTER C.
                ;
5B29 210000     SELDSK: LXI  H,0                ;SET UP FOR ERROR CODE
5B2C 79                 MOV  A,C                ;GET NEW DRIVE.
5B2D FE04               CPI  NDISK              ;CALLING UNDEFINED DRIVE ?
5B2F D0                 RNC                     ;IF NO CY, H,L TELLS CPM YES.
5B30 21195D             LXI  H,DISKNO           ;GET OLD DRIVE NUMBER.
5B33 7E                 MOV  A,M                ;GET OLD DISK NUMBER.
                ;
                        IF   DUAL               ;IF DUAL DRIVE,
                        ANI  0FEH               ;CLEAR OUT BIT 0.
                        ENDIF
                ;
5B34 5F                 MOV  E,A                ;PUT OLD DISK NO. IN D&E.
5B35 1600               MVI  D,0
5B37 211D5D             LXI  H,TRTAB            ;GET ADDRESS OF TRACK TABLE.
5B3A E5                 PUSH H                  ;SAVE ADDRESS OF TRTAB.
5B3B 19                 DAD  D                  ;ADD DISK NO. TO ADDRESS.
5B3C DBF9               IN   TRACK              ;READ 1771 TRACK REGISTER.
5B3E 77                 MOV  M,A                ;PUT INTO TABLE.
5B3F 79                 MOV  A,C                ;GET NEW DISK NUMBER.
                ;
                        IF   DUAL               ;IF A DUAL DRIVE,
                        ANI  0FEH               ;CLEAR BIT 0.
                        ENDIF
                ;
5B40 5F                 MOV  E,A                ;PUT NEW DISK NO. IN D&E.
5B41 E1                 POP  H                  ;RESTORE ADDRESS OF TRTAB.
5B42 19                 DAD  D                  ;ADD DISK NO. TO ADDRESS.
5B43 7E                 MOV  A,M                ;GET NEW TRACK NUMBER.
5B44 D3F9               OUT  TRACK              ;PUT INTO 1771 TRACK REG.
5B46 79                 MOV  A,C                ;UPDATE OLD DISK NUMBER.
5B47 32195D             STA  DISKNO
5B4A 87                 ADD  A                  ;PUT BITS 1&2 AT 4&5.
5B4B 87                 ADD  A
5B4C 87                 ADD  A
5B4D 87                 ADD  A
5B4E 32285D             STA  LATCH              ;SAVE NEW LATCH CODE.
5B51 21215D     DENSITY:LXI  H,DRVFLG           ;POINT TO DRIVE DEN. FLAG
5B54 0600               MVI  B,0                ;CLEAR REG B.
5B56 09                 DAD  B                  ;INDEX INTO DRIVE FLAG LOC.
5B57 7E                 MOV  A,M                ;GET THE FLAG BYTE
5B58 B7                 ORA  A                  ;LOGGED IN?
5B59 FA825B             JM   LOGED              ;YES, IT'S LOGGED.
5B5C E5                 PUSH H                  ;NO, SAVE FLAG ADDRESS.
5B5D 3A285D             LDA  LATCH              ;GET LATCH CODE
5B60 D3FC               OUT  DCONT              ;CHANGE LATCH OR DENSITY
                ;
                ;READ TRACK 0 SECTOR 1 FOR DENSITY BYTE AT 7E HEX.
                ;
5B62 3E01               MVI  A,1                ;SECTOR 1.
5B64 32165D             STA  SECT               ;SAVE THE SECTOR VALUE.
5B67 CDB65B             CALL HOME               ;HOME THE DRIVE.
5B6A 2A175D             LHLD DMAADD             ;GET CP/M DMA ADDRESS VALUE
5B6D E5                 PUSH H                  ;SAVE IT ON THE STACK.
5B6E 21865E             LXI  H,DBUFF            ;POINT TO THE DMA BUFFER.
5B71 22175D             SHLD DMAADD             ;SET UP READ DMA ADDRESS.
                ;
```

```
                ;READ THE DATA USING READ ROUTINE.
                ;
5B74 CD5F5C             CALL READ               ;CBIOS READ ROUTINE.
                ;
                ;GET DENSITY BYTE VALUE AND DETERMINE DRIVE STATUS.
                ;
5B77 E1                 POP  H                  ;RESTORE DMA ADDRESS FROM THE STACK.
5B78 22175D             SHLD DMAADD             ; AND RESTORE THE CP/M DMA ADDRESS.
5B7B E1                 POP  H                  ;RESTORE DENSITY FLAG ADDRESS.
5B7C 3A045F             LDA  DBUFF+7EH          ;INDEX INTO DBUFF TO LOCATION DBUFF+7E.
5B7F F680               ORI  80H                ;set logged in bit
5B81 77                 MOV  M,A                ;place it in flag table.
5B82 011200     LOGED:  LXI  B,18               ;index value through drive table.
5B85 E612               ANI  12H                ;MASK DENSITY AND SIDE BITS OUT.
5B87 B7                 ORA  A                  ;SINGLE DENSITY?
5B88 21735A             LXI  H,SDTAB            ;point to start of tables
5B8B CA8F5B             JZ   DENSIT1            ;yes, overlay param. block
                ;
                        IF   DUBSID
                        DAD  B                  ;no, add offset to next table
                        CPI  2                  ;single den doub sided?
                        JZ   DENSIT1            ;yes
                        DAD  B                  ;no
                        CPI  10H                ;doub den single sided?
                        JZ   DENSIT1            ;yes
                        ENDIF
                ;
5B8E 09                 DAD  B                  ;no, must be doub den , doub sided
5B8F EB         DENSIT1:XCHG                    ;drive table pointer --> d,e
5B90 1A                 LDAX D                  ;get log and drive type byte.
5B91 13                 INX  D                  ;bump pointer
5B92 32255D             STA  DENS               ;set current drive density.
5B95 D5                 PUSH D                  ;save drive table pointer.
5B96 CDA95B             CALL PARINDX            ;compute parameter overlay area.
5B99 D1                 POP  D                  ;restore drive table pointer.
5B9A 010802             LXI  B,0208H            ;B = 2, C = 8 (count values).
5B9D 1A         MOVE:   LDAX D                  ;GET XLT0 BYTE.
5B9E 77                 MOV  M,A                ;AND PUT IT INTO DW TABLE FOR DRIVE.
5B9F 13                 INX  D                  ;BUMP
5BA0 23                 INX  H                  ; POINTERS
5BA1 05                 DCR  B                  ;DECREASE COUNT.
5BA2 C29D5B             JNZ  MOVE               ; AND LOOP TILL ZERO.
5BA5 09                 DAD  B                  ;NOW ADD INDEX INTO DPB0 AREA.
5BA6 73                 MOV  M,E                ;GET LOW POINTER BYTE.
5BA7 23                 INX  H                  ;BUMP POINTER.
5BA8 72                 MOV  M,D                ;GET HIGH POINTER BYTE.
                ;
                ;SELECT DRIVE AS A FUNCTION OF H,L
                ;
5BA9 2A195D     PARINDX:LHLD DISKNO             ;LOAD DISK NUMBER AND ZERO BYTE
5BAC 11335A             LXI  D,DPBASE           ;POINT TO DISK PARM START.
5BAF 29                 DAD  H                  ;*2
5BB0 29                 DAD  H                  ;*4
5BB1 29                 DAD  H                  ;*8
5BB2 29                 DAD  H                  ;*16
5BB3 19                 DAD  D                  ;COMPUTE INDEX FOR THE DRIVE
5BB4 AF                 XRA  A                  ;SET A = 0.
5BB5 C9                 RET                     ;RETURN FROM SELDSK.
                ;
                ; MOVE DISK TO TRACK ZERO.
                ;
5BB6 0E00       HOME:   MVI  C,0                ;SEEK TO TRACK ZERO.
```

```
5BB8 3E02           MVI  A,STPRAT          ;RESTORE COMMAND
5BBA D3F8           OUT  DCOM              ;TELL CONTROLLER.
                    ;
                    ; SET TRACK NUMBER TO WHATEVER IS IN REGISTER C.
                    ; ALSO PERFORM MOVE TO THE CORRECT TRACK (SEEK).
                    ;
                    SETTRK:
5BBC 2A285D         LHLD LATCH             ;get new and old latch.
5BBF 7C             MOV  A,H               ;get latch value.
5BC0 E6B7           ANI  0B7H              ;strip density and side bits.
5BC2 67             MOV  H,A               ;restore it.
                    ;
                    IF   DUBSID            ;if using double sided drive.
                    LDA  DENS              ;check if double sided.
                    RRC
                    RRC                    ;look at bit 1.
                    JNC  NOTSID            ;if bit 1 = 0, it's single sided.
                    MOV  A,C               ;it's doub sided, so get track number.
                    RRC                    ;divide by 2.
                    MOV  B,A               ;save it in reg b.
                    MOV  A,L               ;get old latch value.
                    JC   SIDE2             ;change side if odd track.
                    ANI  0BFH              ;clear side bit from latch.
                    JMP  SETLAT            ;go set the latch.
                    ;
                    SIDE2:  ORI  40H       ;turn on side select bit.
                    SETLAT: STA  CLATCH    ;save it for later.
                    ANI  0BFH              ;clear side bit.
                    CALL OLDLAT            ;check for drive change.
                    MOV  A,B               ;restore doub sided trk number.
                    ANI  7FH               ;clear bit 7.
                    MOV  C,A               ;trk number now in reg c.
                    JMP  TRKSET            ;check for density of track going to.
                    ENDIF
                    ;
                    IF   NOT DUBSID        ;if not using double sided drive
5BC3 C3D15B         JMP  NOTSID            ;jump around subroutine.
                    ENDIF
                    ;
5BC6 BC             OLDLAT: CMP  H         ;new = old?
5BC7 3EFF           MVI  A,0FFH            ;if not, set = ff
5BC9 C2CD5B         JNZ  SFLAG
5BCC 2F             CMA                    ;new = old, set = 0.
5BCD 321B5D         SFLAG:  STA  HLSF      ;save head load select flag.
5BD0 C9             RET
                    ;
5BD1 7D             NOTSID: MOV  A,L       ;get latch value.
5BD2 32295D         STA  CLATCH            ;save it
5BD5 CDC65B         CALL OLDLAT            ;check for drive change.
5BD8 3A255D         TRKSET: LDA  DENS      ;CHECK DRIVE DENSITY FLAG.
5BDB 0F             RRC                    ;IS BIT 0 = 0?
5BDC D2ED5B         JNC  TRKSD             ;YES, WE ARE SINGLE DENSITY.
5BDF 79             MOV  A,C               ;NO, RESTORE TRACK NUMBER.
5BE0 FE01           CPI  1                 ;IS IT TRACK 1?
5BE2 DAED5B         JC   TRKSD             ;IF LESS THAN, SET SINGLE DENSITY.
5BE5 3A295D         LDA  CLATCH            ;GET CURRENT LATCH VALUE.
5BE8 F608           ORI  8                 ;SET FOR DOUBLE DENSITY.
5BEA C3F25B         JMP  TRKDD
                    ;
5BED 3A295D         TRKSD:  LDA  CLATCH    ;GET CURRENT LATCH VALUE.
5BF0 E6F7           ANI  0F7H              ;TURN OFF BIT 4 (SINGLE DENSITY).
5BF2 32295D         TRKDD:  STA  CLATCH    ;SAVE NEW LATCH VALUE.
```

```
5BF5 D3FC               OUT  DCONT              ;SELECT DISK AND MAKE DENSITY CHANGE.
5BF7 79                 MOV  A,C                ;RESTORE TRACK VALUE
5BF8 32155D             STA  TRK                ;UPDATE OLD WITH NEW.
                ;
                ; MOVE THE HEAD TO THE TRACK IN REGISTER A.
                ;
5BFB C5         SEEK:   PUSH B                  ;SAVE B&C.
5BFC 47                 MOV  B,A                ;SAVE DESTINATION TRACK.
5BFD 3E0A               MVI  A,RTCNT            ;GET RETRY COUNT.
5BFF 32275D     SRETRY: STA  SERCNT             ;STORE IN ERROR COUNTER.
5C02 DBF9               IN   TRACK              ;READ PRESENT TRACK NO.
5C04 B8                 CMP  B                  ;SAME AS NEW TRACK NO.?
5C05 C20A5C             JNZ  NOTHR              ;JUMP IF NOT THERE.
5C08 C1         THERE:  POP  B                  ;RESTORE B&C.
5C09 C9                 RET                     ;RETURN FROM SEEK.
5C0A 78         NOTHR:  MOV  A,B                ;RESTORE A FROM B.
5C0B D3FB               OUT  DDATA              ;TRACK TO DATA REGISTER.
5C0D 3E16               MVI  A,14H+STPRAT+HLAB  ;GET STEP RATE, DO
5C0F D3F8               OUT  DCOM               ;SEEK WITH VERIFY.
                ;
                        IF NOT DMACNTL
5C11 DBFC               IN   WAIT               ;WAIT FOR INTRQ.
5C13 DBF8               IN   DSTAT              ;READ STATUS.
                        ENDIF
                ;
                        IF DMACNTL
                        CALL SLOOP              ;NO WAIT STATUS CHECK.
                        ENDIF
                ;
5C15 E691               ANI  91H                ;LOOK AT BITS.
5C17 CA085C             JZ   THERE              ;OK IF ZERO.
5C1A 3A275D             LDA  SERCNT             ;GET ERROR COUNT.
5C1D 3D                 DCR  A                  ;DECREMENT COUNT.
5C1E C2FF5B             JNZ  SRETRY             ;RETRY SEEK.
5C21 C1                 POP  B                  ;RESTORE B&C.
5C22 C5                 PUSH B                  ;SAVE
5C23 CD875C             CALL RECOV              ;IF SEEK RETRY = 10 CHECK
5C26 C1                 POP  B
5C27 79                 MOV  A,C                ;RECOVER TRACK NUMBER.
5C28 C3FB5B             JMP  SEEK               ; FOR CNTL-C FOR ABORT.
                ;
                ; SET DISK SECTOR NUMBER.
                ;
5C2B 79         SETSEC:  MOV  A,C               ;GET SECTOR NUMBER.
5C2C 32165D              STA  SECT              ;PUT AT SECT # ADDRESS.
5C2F C9                  RET                    ;RETURN FROM SETSEC.
                ;
                ;TRANSLATE THE SECTOR GIVEN B,C USING
                ;THE TRANSLATE TABLE;GIVEN BY D,E
                ;
                SECTRAN:
5C30 69                 MOV  L,C                ;GET PHYSICAL SECTOR NUMBER
5C31 2C                 INR  L                  ;BUMP IT BY ONE.
5C32 7A                 MOV  A,D                ;ARE WE USING NO XLAT TABLE?
5C33 B3                 ORA  E                  ;IT WILL BE ZERO IF NOT.
5C34 C8                 RZ                      ;RETURN IF IT IS ZERO.
5C35 EB                 XCHG                    ;H,L = TRANS
5C36 09                 DAD  B                  ;H,L = TRANS (SECTOR)
5C37 6E                 MOV  L,M                ;L   = TRANS (SECTOR)
5C38 2600               MVI  H,0                ;CLEAR REG H
5C3A C9                 RET                     ;H,L = TRANSLATED SECTOR
                ;
```

```
                ; SET DISK DMA ADDRESS.
                ;
5C3B 60         SETDMA: MOV  H,B                ;MOVE B&C TO H&L.
5C3C 69                 MOV  L,C
5C3D 22175D             SHLD DMAADD             ;PUT AT DMA ADR ADDRESS.
5C40 C9                 RET                     ;RETURN FROM SETDMA.
                ;
                ; HDLD - GET HEAD-LOAD BIT IF REQUIRED.
                ;
5C41 3A1B5D     HDLD:   LDA  HLSF               ;GET HEAD-LOAD FLAG.
5C44 B7                 ORA  A                  ;IS A = ZERO?
5C45 CA565C             JZ   HDLD1              ;HOP IF SO.
5C48 2F                 CMA                     ;SET A = 0.
5C49 321B5D             STA  HLSF               ;SET FLAG = 0 IF NOT.
                ;
                ;IF CHANGING TO A NEW DRIVE, PERFORM A SEEK TO
                ;THE SAME TRACK TO UNLOAD THE HEAD ON NEW DRIVE.
                ;
5C4C DBF9               IN   TRACK              ;GET PRESENT TRACK
5C4E D3FB               OUT  DDATA              ;TELL CONTROLLER.
5C50 3E16               MVI  A,14H+STPRAT
5C52 D3F8               OUT  DCOM
                ;
                        IF   NOT DMACNTL
5C54 DBFC               IN   WAIT               ;WAIT FOR INTRQ.
                        ENDIF
                ;
                        IF   DMACNTL
                        CALL SLOOP              ;CHECK DMA STATUS PORT.
                        ENDIF
                ;
5C56 DBF8       HDLD1:  IN   DSTAT              ;READ 1771 STATUS.
5C58 E620               ANI  20H                ;LOOK AT HL BIT.
5C5A 3E04               MVI  A,4
5C5C C8                 RZ                      ;RETURN IF HEAD IS NOT LOADED.
5C5D 97                 SUB  A                  ;HEAD IS ALREADY LOADED.
5C5E C9                 RET                     ;RETURN FROM HDLD.
                ;
                ; READ THE SECTOR AT SECT, FROM THE PRESENT TRACK,
                ; USE STARTING ADDRESS AT DMAADD.
                ;
5C5F 3E0A       READ:   MVI  A,RTCNT            ;GET RETRY COUNT.
                RRETRY:
                        IF DMACNTL
                        LXI  B,80D0H            ;FLOPPY READ, FORCE INTERRUPT.
                        LXI  D,4080H            ;DMA READ, DMA COUNT BYTE
                        CALL DMARW              ;ENTER COMMON READ/WRITE ROUTINE.
                        ENDIF
                ;
                        IF NOT DMACNTL
5C61 0680               MVI  B,80H              ;FLOPPY READ COMMAND BYTE.
5C63 CDC15C             CALL DSKSET             ;SET UP DISK CONTROLLER.
5C66 B0                 ORA  B                  ;'OR' IN THE READ COMMAND.
5C67 D3F8       READE:  OUT  DCOM               ;SEND COMMAND TO 1771.
5C69 DBFC       RLOOP:  IN   WAIT               ;WAIT FOR DRQ OR INTRQ.
5C6B B7                 ORA  A                  ;SET FLAGS.
5C6C F2765C             JP   RDDONE             ;DONE IF INTRQ.
5C6F DBFB               IN   DDATA              ;READ A DATA BYTE FROM DISK.
5C71 77                 MOV  M,A                ;PUT BYTE INTO MEMORY.
5C72 23                 INX  H                  ;INCREMENT MEMORY POINTER.
5C73 C3695C             JMP  RLOOP              ;KEEP READING.
5C76 DBF8       RDDONE: IN   DSTAT              ;READ DISK STATUS.
```

```
                        ENDIF
                ;
                        IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
                        EI                      ;ALLOW AGAIN HERE.
                        ENDIF
                ;
5C78 E69D               ANI  9DH                ;LOOK AT ERROR BITS.
5C7A C8                 RZ                      ;RETURN IF NONE.
5C7B CD955C             CALL ERCHK              ;CHECK FOR SEEK ERROR.
5C7E C2615C             JNZ  RRETRY             ;TRY TO READ AGAIN.
5C81 CD875C             CALL RECOV              ;CHECK FOR ABORT OR CONTINUE
5C84 C35F5C             JMP  READ               ;IF NOT CNTL-C, TRY TO READ AGAIN.
                ;
                ;RECOV
                ;THIS ROUTINE IS CALLED BY ANY READ,WRITE,SEEK ROUTINE IF THE RETRY
                ;COUNT GOES TO 10. IF IT DOES,THIS ROUTINE CALLS CONIN FOR A KEY TO
                ;BE PUSHED. IF THE KEY IS A CNTL-C, THEN A WARMBOOT IS EXECUTED. IF
                ;ANY OTHER KEY IS PUSHED, THEN A RETURN IS MADE BACK TO THE CALLER
                ;AND THAT ROUTINE IS RETRIED FOR 10 MORE TIMES.
                ;
                RECOV:
5C87 0E65               MVI  C,'e'              ;ERROR CODE
5C89 CD1E5B             CALL CONOT              ;PRINT IT
5C8C CD135B             CALL CONIN              ;CHECK FOR PUSHED KEY.
5C8F FE03               CPI  03H                ;IS IT A CNTL-C ?
5C91 C0                 RNZ                     ;RETURN TO CALLER IF NOT.
5C92 C3B15A             JMP  WBOOT              ;YES, DO WARMBOOT.
                ;
                ; ERCHK - CHECK FOR RECORD NOT FOUND ERROR.
                ;
5C95 E610       ERCHK:  ANI  10H                ;IF RECORD NOT FOUND,
5C97 C29F5C             JNZ  CHKSK              ;DO A CHECK ON SEEK.
5C9A 3A265D     CHKOK:  LDA  ERCNT              ;GET RETRYS ALLOWED
5C9D 3D                 DCR  A                  ;DECREASE IT,
5C9E C9                 RET                     ;AND RETURN WITH NUMBER.
                ;
                ;CHECK FOR SEEK TO CORRECT TRACK,
                ;AND CHANGE IF NECESSARY.
                ;
                        IF NOT DMACNTL
5C9F 3EC4       CHKSK:  MVI  A,0C4H             ;SEND COMMAND TO 1771
5CA1 D3F8               OUT  DCOM               ;TO READ ADDRESS.
5CA3 DBFC               IN   WAIT               ;WAIT FOR DRQ OR INTRQ.
5CA5 DBFB               IN   DDATA              ;READ THE TRACK ADDRESS.
5CA7 F5                 PUSH PSW                ;SAVE IT ON THE STACK.
5CA8 DBFD       CHKS2:  IN   DMACHK             ;WAIT FOR INTRQ.
5CAA B7                 ORA  A                  ;SET FLAGS.
5CAB F2B35C             JP   CHKS3              ;DONE WITH READ ADR OP.
5CAE DBFB               IN   DDATA              ;READ ANOTHER BYTE.
5CB0 C3A85C             JMP  CHKS2              ;DO IT AGAIN.
5CB3 DBF8       CHKS3:  IN   DSTAT              ;READ DISK STATUS.
                        ENDIF
                ;
                        IF   DMACNTL
                CHKSK:  LXI  H,BIOS-7           ;POINT TO UNUSED SPACE
                        LXI  B,0C4D0H           ;READ ADDRESS, FORCE INTERRUPT CMNDS.
                        LXI  D,04006H           ;DMA READ, COUNT BYTE
                        CALL DMARWE             ;READ THE ID USING DMA CONTROL.
                        ORA  A                  ;SET FLAGS.
                        JZ   CHKS4              ;READ ADR OK IF 0.
                        CALL HOME               ;OTHERWISE, HOME FIRST.
                        JMP  CHKS5
```

```
                        ENDIF
                ;
                        IF   NOT DMACNTL
5CB5 F1         CHKS4:  POP  PSW                ;UPDATE TRACK REGISTER.
                        ENDIF
                ;
                        IF   DMACNTL
                CHKS4:  IN   SECTP              ;GET THE TRACK BYTE
                        ENDIF
                ;
5CB6 D3F9               OUT  TRACK
5CB8 3A155D     CHKS5:  LDA  TRK                ;GET REQUIRED TRACK NO.
5CBB CDFB5B             CALL SEEK               ;MOVE THE HEAD TO IT.
5CBE C39A5C             JMP  CHKOK              ;EXIT FROM ERROR CHECK.
                ;
                        IF   NOT DMACNTL
5CC1 32265D     DSKSET: STA  ERCNT              ;STORE IN ERROR CTR.
5CC4 3ED0               MVI  A,0D0H             ;CAUSE INTERRUPT.
5CC6 D3F8               OUT  DCOM
5CC8 E3                 XTHL                    ;SOME
5CC9 E3                 XTHL                    ; DELAY
                        ENDIF
                ;
                        IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
                        DI                      ;DISABLE THEM HERE.
                        ENDIF
                ;
                        IF   NOT DMACNTL
5CCA 2A175D             LHLD DMAADD             ;GET STARTING ADDR.
5CCD 3A165D             LDA  SECT               ;GET SECTOR NUMBER.
5CD0 D3FA               OUT  SECTP              ;SET SECTOR INTO 1771.
5CD2 CD415C             CALL HDLD               ;GET HEAD-LOAD BIT?
5CD5 C9                 RET                     ;RETURN TO CALLER
                        ENDIF
                ;
                ; WRITE THE SECTOR AT SECT, ON THE PRESENT TRACK,
                ; USE STARTING ADDRESS AT DMAADD.
                ;
5CD6 3E0A       WRITE:  MVI  A,RTCNT            ;GET RETRY COUNT.
                WRETRY:
                        IF   DMACNTL
                        LXI  B,0A0D0H           ;FLOPPY WRITE, FORCE INTERRUPT.
                        LXI  D,08080H           ;DMA WRITE, DMA COUNT BYTE.
                        CALL DMARW              ;ENTER COMMON READ/WRITE ROUTINE.
                        ENDIF
                ;
                        IF   NOT DMACNTL
5CD8 06A0               MVI  B,0A0H             ;FLOPPY WRITE COMMAND BYTE.
5CDA CDC15C             CALL DSKSET             ;SET UP FLOPPY CONTROLLER.
5CDD B0                 ORA  B                  ;'OR' IN WRITE COMMAND.
5CDE D3F8       WRITE2: OUT  DCOM
5CE0 DBFC       WLOOP:  IN   WAIT               ;WAIT FOR READY.
5CE2 B7                 ORA  A                  ;SET FLAGS.
5CE3 F2ED5C             JP   WDONE              ;HOP OUT WHEN DONE.
5CE6 7E                 MOV  A,M                ;GET BYTE FROM MEM.
5CE7 D3FB               OUT  DDATA              ;WRITE ONTO DISK.
5CE9 23                 INX  H                  ;INCREMENT MEM PTR.
5CEA C3E05C             JMP  WLOOP              ;KEEP WRITING.
5CED DBF8       WDONE:  IN   DSTAT              ;READ DISK STATUS.
                        ENDIF
                ;
                        IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
```

```
                        EI                      ;ENABLE AGAIN HERE.
                        ENDIF
                ;
5CEF E6FD               ANI  0FDH               ;LOOK AT THESE BITS.
5CF1 C8                 RZ                      ;RETURN IF NO ERR.
5CF2 CD955C             CALL ERCHK              ;CHECK/CORRECT SEEK ERR.
5CF5 C2D85C             JNZ  WRETRY             ;TRY TO WRITE AGAIN.
5CF8 CD875C             CALL RECOV              ;CHECK FOR ABORT
5CFB C3D65C             JMP  WRITE              ;RETRY WRITE AGAIN.
                ;
                ;LIST STATUS CHECK ROUTINE
                ;
5CFE CD065D     PRSTAT: CALL PSTAT              ;CHECK PRINTER STATUS PORT.
                ;
5D01 3E00               MVI  A,0                ;RETURN STATUS ACTIVITY.
                ;
                        IF   TARDEL OR RDYLO
5D03 C0                 RNZ
                        ENDIF
                ;
                        IF   RDYHI
                        RZ
                        ENDIF
                ;
5D04 2F                 CMA                     ;INVERT IT
                ;
                ; PUNCH AND READER ARE NOT SUPPORTED.
                ;
                PUNCH:
5D05 C9         READER: RET
                ;
                ;PSTAT - PRINTER STATUS CHECK ROUTINE.
                ;
5D06 DB02       PSTAT:  IN   LSTAT              ;READ PRINTER STATUS PORT.
                ;
                        IF   NOT TARDEL
5D08 E680               ANI  LRBIT
                        ENDIF
                ;
                        IF   TARDEL
                        ANI  81H                ;MASK READY BITS
                        XRI  81H
                        ENDIF
                ;
5D0A C9                 RET                     ;RETURN TO CALLER
                ;
                ; WRITE A CHARACTER ON LISTING DEVICE.
                ;
                LIST:
                        IF   LSTNUL             ;IF NULLS OR PAGING,
                        MVI  A,0DH              ;IF IT'S A CR,
                        CMP  C                  ;THEN HOP OUT TO
                        JZ   LINUL              ;NULL ROUTINE.
                        ENDIF
                ;
                        IF   LSTPAG             ;IF PAGING
                        MVI  A,0AH              ;GET A LINEFEED
                        CMP  C                  ;DOES IT MATCH?
                        JZ   LINUL3
                        MOV  A,C
                        CPI  0CH
                        RZ
```

```
                        ENDIF
                ;
5D0B CD065D     LTBSY:  CALL PSTAT              ;READ LISTER STATUS.
                ;
                        IF   TARDEL OR RDYLO
5D0E C20B5D             JNZ  LTBSY              ;LOOP TILL LOW.
                        ENDIF
                ;
                        IF   NOT TARDEL AND RDYHI
                        JZ   LTBSY              ;LOOP TILL HIGH.
                        ENDIF
                ;
5D11 79                 MOV  A,C                ;GET DATA BYTE.
5D12 D303               OUT  LDATA              ;PRINT IT.
5D14 C9                 RET                     ;RETURN FROM LIST.
                ;
                        IF   LSTNUL             ;IF LIST NULLS
                LINUL:  PUSH B                  ;SAVE B&C.
                        MVI  B,(LNULL AND 0FFH)+1 ;GET NULL COUNT
                LINUL1: CALL LTBSY              ;PRINT (CR FIRST).
                        MVI  C,0                ;GET NULL CHAR.
                        DCR  B                  ;DECREMENT COUNTER.
                        JNZ  LINUL1             ;DO NEXT NULL.
                        JMP  LINUL2             ;EXIT THE ROUTINE.
                        ENDIF
                ;
                        IF   LSTPAG             ;IF LIST DEV. PAGING,
                LINUL3: PUSH B                  ;SAVE B,C PAIR
                        LDA  LFCNT              ;GET LINE-FEED COUNT.
                        INR  A                  ;INCREMENT IT.
                        STA  LFCNT              ;SAVE IT BACK.
                        CPI  LINCNT-(LINCNT/11) ;END OF PAGE?
                        MVI  B,1                ;SET UP FOR 1 LF.
                        JNZ  NOTEOP             ;HOP IF NOT END.
                        XRA  A                  ;SET LF COUNT = 0.
                        STA  LFCNT
                        MVI  B,(LINCNT/11)+1    ;BETWEEN PAGES.
                NOTEOP: MVI  C,0AH              ;GET LINE-FEED CODE.
                LSTPA1: CALL LTBSY              ;PRINT LINE-FEED.
                        DCR  B                  ;DECREMENT LF COUNTER.
                        JNZ  LSTPA1             ;DO NEXT LINE FEED?
                        ENDIF
                ;
                        IF   LSTNUL OR LSTPAG   ;IF NULLS OR PAGING,
                LINUL2: POP  B                  ;RESTORE B&C.
                        MOV  A,C                ;RESTORE A.
                        RET                     ;RETURN FROM LIST.
                        ENDIF
                ;
5D14 =          ENDPROG EQU     $-1             ;ENDING ADDRESS.
                ;
                ;NOTE:  AS THERE ARE ONLY SIX (6) SECTORS AVAILABLE FOR CBIOS ON
                ;THE SECOND SYSTEM TRACK (1), THE LAST ADDRESS BEFORE THIS POINT
                ;SHOULD BE NO GREATER THAN THE CBIOS STARTING ADDRESS + 037F (HEX).
                ;THIS WILL NORMALLY BE XD7F (HEX).
                ;
                ; BIOS SCRATCH AREA.
                ;
5D15            TRK:    DS   1                  ;CURRENT TRACK NUMBER.
5D16            SECT:   DS   1                  ;CURRENT SECTOR NUMBER.
5D17            DMAADD: DS   2                  ;DISK TRANSFER ADDRESS.
                ;
```

```
                ; THE NEXT SEVERAL BYTES, BETWEEN STARTZ AND
                ; ENDZ, ARE SET TO ZERO AT COLD BOOT TIME.
                ;
                STARTZ:                         ;START OF ZEROED AREA.
                ;
5D19            DISKNO: DS   2                  ;DISK NUMBER
                ;
                ; SPECIAL FLAGS.
                ;
5D1B            HLSF:   DS   1                  ;HEAD-LOAD SELECT FLAG.
5D1C            LFCNT:  DS   1                  ;PAGING LINE-FEED COUNT.
                ;
                ; TRTAB - DISK TRACK TABLE - PRESENT POSITION OF
                ;       HEADS FOR UP TO 4 DRIVES.
                ;
5D1D            TRTAB:  DS   4
5D21            DRVFLG: DS   4                  ;DRIVE DENSITY FLAGS.
5D25            DENS:   DS   1                  ;CURRENT DRIVE DENSITY VALUE.
5D26            ERCNT:  DS   1                  ;ERROR COUNT FOR RETRIES.
5D27            SERCNT: DS   1                  ;SEEK RETRY COUNTER.
5D28            LATCH:  DS   1                  ;NEW CODE FOR LATCH.
5D29            CLATCH: DS   1                  ;CURRENT CODE IN LATCH.
                ;
5D2A =          ENDZ    EQU  $
                ;
5D2A =          BEGDAT  EQU  $
5D2A            DIRBUF: DS   128                ;DIRECTORY BUFFER
5DAA            ALV0:   DS   31
5DC9            CSV0:   DS   24
5DE1            ALV1:   DS   31
5E00            CSV1:   DS   24
5E18            ALV2:   DS   31
5E37            CSV2:   DS   24
5E4F            ALV3:   DS   31
5E6E            CSV3:   DS   24
5E86 =          ENDDAT  EQU  $
015C =          DATSIZ  EQU  $-BEGDAT           ;TOTAL SIZE OF DISK PARM STORAGE.
                ;
5E86            DBUFF:  DS   128                ;128 BYTE DENSITY SELECT BUFFER.
                ;
5D14                    ORG  ENDPROG            ;SHOW ACTUAL ENDING ADDRESS OF BIOS
5D14                    END
```

```
                ;
                ; TARBELL ELECTRONICS CP/M COLDSTART LOADER
                ; VERSION OF 7-31-80.
                ;-----------------------------------------------------------
                ; Modified for DMA Control - 1-5-80.
                ; Modified for reading larger bios from TRK 1  - 6-5-80.
                ; Modified for Tarbell CPU Card  -  7-3-80.
                ; G.W.Mulchin
                ; Tarbell Electronics
                ;-----------------------------------------------------------
                ; Copyright (c) 1980 Tarbell Electronics
                ;
                ;
                ;*****************************************************************
                ;*                                                               *
                ;*                        ** NOTE **                             *
                ;*                        ============                           *
                ;*                                                               *
                ;*      The equate for Double Density (DOUBDEN) must only be     *
                ;* set true for a disk which is formatted in double density only *
                ;* and one which you wish to put an operating system on to.      *
                ;* Otherwise, leave it false if you are building an operating    *
                ;* system on to a single density formatted disk.                 *
                ;*                                                               *
                ;*****************************************************************
                ;
                ;
                ;
                ; THIS PROGRAM IS LOADED AT LOCATION ZERO BY THE BOOTSTRAP PROGRAM,
                ; AND EXECUTED. ITS PURPOSE IS TO LOAD AND EXECUTE THE CP/M DISK
                ; OPERATING SYSTEM AT THE TOP OF THE MEMORY IN USE.
                ;
0000 =          FALSE   EQU  0          ;DEFINE VALUE OF FALSE.
FFFF =          TRUE    EQU  NOT FALSE  ;DEFINE VALUE OF TRUE.
                ;
                ;********* THIS IS THE AREA TO MAKE CHANGES IN ******************
                ;********* FOR DIFFERENT SYSTEM CONFIGURATIONS ******************
                ;                                                              **
0018 =          MSIZE   EQU  24         ;MEMORY SIZE IN DECIMAL KB.           **
0000 =          TARBELL EQU  FALSE      ;TRUE IF USING TARBBELL CPU.          **
0000 =          DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED SYSTEMS.       **
0000 =          DELTA   EQU  FALSE      ;TRUE IF USING DELTA CPU CARD         **
0000 =          DOUBDEN EQU  FALSE      ;TRUE IF DOUB. DEN DISK.              **
0000 =          DMACNTL EQU  FALSE      ;TRUE IF USING DMA CONTROL            **
0000 =          BASE    EQU  0          ;TARBELL I/O PORTS (00 or 10 HEX)     **
001A =          SPT     EQU  26         ;NUMBER OF SECTORS PER TRACK.         **
001A =          DDS     EQU  26         ;sectors in trk 1 , (Range = 26 to 51)**
00F8 =          DISK    EQU  0F8H       ;DISK PORT BASE ADDRESS.              **
                ;                                                              **
                ;*****************************************************************
                ;
                        IF   TARBELL
                IO      EQU  BASE       ;i/o ports on tarbell cpu.
                MMENB   EQU  IO+10      ;memory management enable port.
                MEMMAG  EQU  BASE+32    ;memory management port.
                        ENDIF

00E0 =          ADR0    EQU  0E0H       ;DMA ADDRESS PORT.
```

```
00E1 =          WCT0    EQU  0E1H       ;DMA WORD COUNT PORT.
00E8 =          CMND    EQU  0E8H       ;DMA COMMAND PORT.
00F8 =          DCOM    EQU  DISK       ;COMMAND PORT.
00F8 =          DSTAT   EQU  DISK       ;STATUS PORT.
00F9 =          TRACK   EQU  DISK+1     ;TRACK PORT.
00FA =          SECT    EQU  DISK+2     ;SECTOR PORT.
00FB =          DATA    EQU  DISK+3     ;DATA PORT.
00FC =          WAIT    EQU  DISK+4     ;WAIT PORT.
00FC =          DCONT   EQU  DISK+4     ;CONTROL PORT.
00FD =          DMACHK  EQU  DISK+5     ;DMA CHECK PORT.
00FF =          PANEL   EQU  0FFH       ;front panel machines.
1000 =          CBASE   EQU  (MSIZE-20)*1024
4400 =          CPMB    EQU  CBASE+3400H;START OF CP/M.
5A00 =          BOOTE   EQU  CPMB+1600H ;COLD BOOT ENTRY POINT.
0019 =          SDS     EQU  25         ;always 25 sectors to read in trk 1.
0033 =          NSECTS  EQU  SDS + DDS  ;SECTORS OF CP/M.
000A =          RTCNT   EQU  10         ;NUMBER OF RETRYS.
                ;
0000                    ORG  0          ;START OF LOADER.
                ;
                BOOT:
                        IF   TARBELL    ;IF USING TARBELL CPU
                        LXI  D,1000H    ;COUNT=16, DATA BYTE = 0
                        MVI  C,MEMMAG AND 0FFH
                MLOOP:  MOV  A,E        ;GET ADDRESS VALUE
                        ORA  C          ;MAKE I/O PORT VALUE
                        STA  MOUT+1     ;MODIFY PORT ON THE FLY
                        MOV  A,E        ;GET INIT VALUE.
                        CMA             ;FLIP IT FOR RAM ON CPU
                MOUT:   OUT  BASE       ;PUT IT TO RAM ON CPU
                        INR  E          ;BUMP DATA VALUE
                        DCR  D          ;DECREASE COUNT
                        JNZ  MLOOP      ;LOOP 16 TIMES.
                        OUT  MMENB      ;ENABLE MEMORY MANAGEMENT.
                        ENDIF
                ;
                        IF   DELTA      ;IF USING DELTA CPU.
                        MVI  A,1        ;GET A 1 IN REG A.
                        OUT  9          ; AND DISABLE CPU ROM SLOT.
                        ENDIF
                ;
0000 1E0A               MVI  E,RTCNT    ;GET RETRY COUNT.
0002 310001     BLOOP:  LXI  SP,100H    ;SET STACK POINTER.
0005 210044             LXI  H,CPMB     ;CP/M STARTS HERE.
0008 1633               MVI  D,NSECTS   ;NUMBER OF SECTORS TO READ.
000A 0E02               MVI  C,2        ;SECTOR NUMBER.
000C 0604       RNTRK:  MVI  B,4        ;FOR HEAD LOAD.
000E CD2900     RNSEC:  CALL READ       ;READ FIRST SECTOR.
0011 15                 DCR  D          ;IF DONE,
0012 CA005A             JZ   BOOTE      ;GO TO CP/M.
0015 0600               MVI  B,0        ;FOR NO HEAD LOAD.
0017 0C                 INR  C          ;INCREMENT SECTOR COUNT.
0018 79                 MOV  A,C        ;DONE WITH
0019 FE1B       SECCMP: CPI  SPT+1      ;THIS TRACK?
001B DA0E00             JC   RNSEC      ;IF NOT, READ NEXT SECTOR.
                ;
                        IF   DOUBDEN AND NOT DUBSID
                        MVI  A,DDS + 1  ;number of sectors to read on trk 2.
                        STA  SECCMP+1   ;modify sector compare value.
                        MVI  A,8        ;GET SET DOUBLE DENSITY CODE
                        OUT  WAIT       ;SET LATCH FOR D.DENSITY
                        ENDIF
```

```
                        ;
                                IF   NOT DUBSID
001E 3E5B                       MVI  A,5BH      ;STEP COMMAND.
0020 D3F8                       OUT  DCOM       ;ISSUE IT.
0022 DBFC                       IN   WAIT       ;WAIT UNTIL DONE.
                                ENDIF
                        ;
                                IF   DUBSID     ;IF DOUBLE SIDED SYSTEM.
                                MVI  A,40H      ;SIDE SELECT COMMAND.
                                OUT  DCONT      ;ISSUE IT.
                                ENDIF
                        ;
0024 0E01                       MVI  C,1        ;SECTOR NUMBER.
0026 C30C00                     JMP  RNTRK      ;READ NEXT TRACK.
                        ;
                        READ:
                                IF   DMACNTL    ;IF USING DMA CONTROL.
                                MVI  A,41H      ;SET UP FOR CHAN 0 REQ.
                                OUT  CMND
                                MVI  A,7FH      ;COUNT FOR 128 BYTES
                                OUT  WCT0
                                MVI  A,40H      ;READ COMMAND
                                OUT  WCT0
                                MOV  A,L        ;GET LOW ADDRESS BYTE
                                OUT  ADR0
                                MOV  A,H        ;HIGH ADDRESS BYTE
                                OUT  ADR0
                                ENDIF
                        ;
0029 79                         MOV  A,C        ;SECTOR IN A.
002A D3FA                       OUT  SECT       ;SET SECTOR REGISTER.
002C 3E88                       MVI  A,88H      ;COMMAND FOR READ.
002E B0                         ORA  B          ;GET HEAD LOAD BIT.
002F D3F8                       OUT  DCOM       ;ISSUE COMMAND.
                        ;
                                IF   NOT DMACNTL;IF NOT USING DMA CONTROL.
0031 DBFC               RLOOP:  IN   WAIT       ;WAIT FOR DRQ.
0033 B7                         ORA  A          ;SET FLAGS.
0034 F23E00                     JP   CHECK      ;JUMP IF DONE.
0037 DBFB                       IN   DATA       ;READ DATA.
0039 77                         MOV  M,A        ;PUT IN MEMORY.
003A 23                         INX  H          ;INCREMENT POINTER.
003B C33100                     JMP  RLOOP      ;LOOP UNTIL DONE.
                                ENDIF
                        ;
                                IF  DMACNTL
                        SLOPP:  IN   DMACHK     ;CHECK DMA STATUS
                                RLC             ; BIT 7
                                JC   SLOPP      ;LOOP IF CARRY
                                PUSH B          ;SAVE B,C PAIR
                                LXI  B,128      ;COUNT SET FOR 128 BYTES
                                DAD  B          ;ADJUST H,L BY 128 BYTES
                                POP  B          ;RESTORE BC
                                ENDIF
                        ;
003E DBF8               CHECK:  IN   DSTAT      ;READ STATUS.
0040 E69D                       ANI  9DH        ;LOOK AT ERROR BITS.
0042 C8                         RZ              ;OK IF ZERO.
0043 1D                         DCR  E          ;DECREMENT RETRY COUNT.
0044 C20200                     JNZ  BLOOP      ;TRY AGAIN IF NOT ZERO.
0047 2F                         CMA             ;flip for front panel
0048 D3FF                       OUT  PANEL      ;show error code from floppy.
```

```
004A C34A00     HERE:   JMP  HERE       ;LOOP.
                ;
007D                    ORG  7DH        ;PUT JUMP HERE.
                ;
                        IF   DOUBDEN AND NOT DUBSID     ;IF RUNNING DOUBLE DENSITY
                        RST  0          ;DO RESTART 0
                        DB   0DDH       ;THIS BYTE MUST BE HERE IF DOUB DEN.
                        DB   0          ;THIS BYTE UNUSED
                        ENDIF
                ;
                        IF   DOUBDEN AND DUBSID
                        RST  0
                        DB   0DFH
                        DB   0
                        ENDIF
                ;
                        IF   NOT DOUBDEN AND NOT DUBSID
007D C7                 RST  0          ;DO WARM BOOT WITH RST INST.
007E E5                 DB   0E5H
007F 00                 DB   0
                        ENDIF
                ;
                        IF   NOT DOUBDEN AND DUBSID
                        RST  0
                        DB   0E7H
                        DB   0
                        ENDIF
                ;
0080                    END   BOOT      ;END OF BOOT.
```

```
                SLOPP:  IN    DMACHK      ;CHECK DMA STATUS
                        RLC               ; BIT 7
                        JC    SLOPP       ;LOOP IF CARRY
                        PUSH  B           ;SAVE B,C PAIR
                        LXI   B,128       ;COUNT SET FOR 128 BYTES
                        DAD   B           ;ADJUST H,L BY 128 BYTES
                        POP   B           ;RESTORE BC
                        ENDIF
                ;
003E DBF8       CHECK:  IN    DSTAT       ;READ STATUS.
0040 E69D               ANI   9DH         ;LOOK AT ERROR BITS.
0042 C8                 RZ                ;OK IF ZERO.
0043 1D                 DCR   E           ;DECREMENT RETRY COUNT.
0044 C20200             JNZ   BLOOP       ;TRY AGAIN IF NOT ZERO.
0047 2F                 CMA               ;flip for front panel
0048 D3FF               OUT   PANEL       ;show error code from floppy.
004A C34A00     HERE:   JMP   HERE        ;LOOP.
                ;
007D                    ORG   7DH         ;PUT JUMP HERE.
007D C7                 RST   0           ;USE RESTART
                ;
                        IF    DOUBDEN
                        DB    0DDH        ;THIS BYTE MUST BE HERE IF DOUB DEN.
                        DB    0           ;THIS BYTE UNUSED
                        ENDIF
                ;
                        IF    NOT DOUBDEN
007E E5                 DB    0E5H
007F 00                 DB    0
                        ENDIF
                ;
0080                    END   BOOT        ;END OF BOOT.

B>
```

```
                ; CP/M BASIC INPUT/OUTPUT OPERATING SYSTEM (BIOS)
                ; TARBELL ELECTRONICS CPM 2.2 VERSION OF 7-16-80.
                ; Copyright (c) 1980 Tarbell Electronics.
                ;
                ; THIS BIOS NOW SUPPORTS THE NEW TARBELL Z-80 CPU BOARD.
                ; THIS MODULE CONTAINS ALL THE INPUT/OUTPUT ROUTINES
                ; FOR THE CP/M SYSTEM, INCLUDING THE DISK ROUTINES.
                ;
                ; THIS SECTION DEFINES THE I/O PORTS AND STATUS BITS.
                ; BY SETTING THE PROPER VALUES FOR THE EQU STATEMENTS,
                ; THE I/O MAY BE AUTOMATICALLY RECONFIGURED TO FIT MOST
                ; SITUATIONS. THE TRUE AND FALSE ONES CONTROL CONDITIONAL
                ; ASSEMBLIES OF DIFFERENT SECTIONS OF I/O ROUTINES TO FIT
                ; DIFFERENT INTERFACE REQUIREMENTS.

FFFF =          TRUE    EQU  0FFFFH     ;DEFINE VALUE OF TRUE.
0000 =          FALSE   EQU  NOT TRUE   ;DEFINE VALUE OF FALSE.

                ;*****************************************************
                ;*** THIS BEGINS THE AREA WHICH REQUIRES CHANGES ***
                ;***      FOR DIFFERENT CONSOLE I/O SYSTEMS        ***
                ;*****************************************************

0018 =          MSIZE   EQU  24         ;MEMORY SIZE IN KBYTES.
0000 =          INTRP   EQU  FALSE      ;TRUE IF INTERRUPTS ALLOWED.
0000 =          TARBELL EQU  FALSE      ;TRUE IF USING THE TARBELL Z-80 CPU.
0000 =          IOBASE  EQU  0          ;BASE IO ADDR FOR TARBELL CPU (0 or 10 hex).
0000 =          TIMER   EQU  FALSE      ;TRUE IF USING CPU TIMER (Tarbell CPU board).
FFFF =          STD     EQU  TRUE       ;TRUE IF STANDARD I/O.
0000 =          MSIO2   EQU  FALSE      ;TRUE IF MITS 2SIO.
0000 =          VDB8024 EQU  FALSE      ;TRUE IF USING VDB-8024 BOARD.
0000 =          DELTA   EQU  FALSE      ;TRUE IF USING DELTA PRODUCTS CPU.
0000 =          ISIO2   EQU  FALSE      ;TRUE IF IMSAI SIO-2.
0000 =          TUART   EQU  FALSE      ;TRUE IF CROMEMCO TUART.
0000 =          VIDEO   EQU  FALSE      ;TRUE IF USING A MEMORY MAPPED VIDEO.
0000 =          OTHER   EQU  FALSE      ;TRUE IF SOMETHING ELSE.
0000 =          SOLOS   EQU  FALSE      ;TRUE IF PROC TECH SOLOS.
0000 =          DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED DRIVES.
0000 =          DUAL    EQU  FALSE      ;TRUE IF DUAL DRIVE.
0000 =          PERSCI  EQU  FALSE      ;TRUE IF FAST SEEK (PERSCI).
0004 =          NDISK   EQU  4          ;DEFINES THE NUMBER DRIVES IN SYSTEM.

                        IF  VIDEO       ;IF USING A VIDEO BOARD
                OUTADDR EQU  0          ;PUT OUTPUT ADDRESS HERE
                        ENDIF

                        IF NOT SOLOS AND NOT TARDEL     ;IF NOT PROC TECH SOLOS,
0000 =          CSTAT   EQU  0          ;CONSOLE STATUS PORT.
0000 =          CCOM    EQU  0          ;CONSOLE COMMAND PORT.
0001 =          CDATA   EQU  1          ;CONSOLE DATA PORT.
0002 =          LSTAT   EQU  2          ;LIST STATUS PORT.
0002 =          LCOM    EQU  2          ;LIST COMMAND PORT.
0003 =          LDATA   EQU  3          ;LIST DATA PORT.
                        ENDIF

0000 =          CONUL   EQU  FALSE      ;CONSOLE NULLS?
0000 =          CNULL   EQU  0          ;CONSOLE NULL COUNT.
0000 =          LSTNUL  EQU  FALSE      ;LIST DEVICE NULLS?
0000 =          LNULL   EQU  0          ;LIST NULL COUNT.
0000 =          LSTPAG  EQU  FALSE      ;LIST DEVICE PAGING?
0042 =          LINCNT  EQU  66         ;LINES PER PAGE.
0000 =          HLAB    EQU  0          ;8 FOR HD LD AT BEG OF SEEK.
0002 =          STPRAT  EQU  2          ;RATE 1=6MS, 2=10MS, 3=20MS.

                ;*****************************************************
                ;*** THIS IS THE END OF THE AREA WHICH NORMALLY NEED ***
                ;***      BE CHANGED FOR MOST CONSOLE I/O SYSTEMS     ***
                ;*****************************************************

FFFF =          RDYLO   EQU  STD OR SOLOS OR OTHER  ;STATUS READY WHEN LOW.
0000 =          RDYHI   EQU  NOT RDYLO
0000 =          TARDEL  EQU  TARBELL OR DELTA   ;IF USING TARBELL OR DELTA CPU.
                ;
                        IF   TARDEL     ;IF USING TARBELL OR DELTA CPU
                CCOM    EQU  IOBASE+1   ;CONSOLE COMMAND PORT
                CSTAT   EQU  IOBASE+1   ;CONSOLE STATUS PORT ( CHAN A.)
                CDATA   EQU  IOBASE+0   ;CONSOLE DATA PORT
```

```
                LCOM    EQU  IOBASE+3    ;LIST COMMAND PORT
                LSTAT   EQU  IOBASE+3    ;LIST STATUS PORT (CHAN B.)
                LDATA   EQU  IOBASE+2    ;LIST DATA PORT
                        ENDIF
                ;
                        IF      TIMER AND TARBELL       ;MUST BE USING TARBELL CPU.
                ;
                ; TIMER EQUATES
                ;
                TCH0    EQU     IOBASE+4        ;TIMER CHAN 0 ADDRESS
                TCH1    EQU     IOBASE+5        ;TIMER CHAN 1 ADDRESS
                TCH2    EQU     IOBASE+6        ;TIMER CHAN 2 ADDRESS
                TCMND   EQU     IOBASE+7        ;TIMER COMMAND PORT
                IMASK   EQU     IOBASE+8        ;INTERRUPT MASKING PORT
                CNTR0   EQU     00000000B       ;counter 0
                CNTR1   EQU     01000000B       ;counter 1
                CNTR2   EQU     10000000B       ;counter 2
                RLWORD  EQU     00110000B       ;read/load lsb 1st, msb 2nd.
                RLHBYTE EQU     00100000B       ;read/load msb only.
                RLLBYTE EQU     00010000B       ;read/load lsb only.
                CNTRLT  EQU     00000000B       ;counter latching operation.
                BINARY  EQU     00000000B       ;select binary operation.
                BCD     EQU     00000001B       ;select BCD operation.
                MODE0   EQU     00000000B       ;interrupt on terminal count.
                MODE1   EQU     00000010B       ;programmable One-shot.
                MODE2   EQU     00000100B       ;rate generator.
                MODE3   EQU     00000110B       ;square wave rate generator.
                MODE4   EQU     00001000B       ;software triggered strobe.
                MODE5   EQU     00001010B       ;hardware triggered strobe.
                        ENDIF

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                CSTAT   EQU  0FAH       ;CONSOLE STATUS PORT.
                KBD     EQU  0C02EH     ;SOLOS KEYBOARD.
                CLRSCR  EQU  0C0D5H     ;CLEAR SCREEN.
                SCRN    EQU  0C054H     ;SOLOS OUTPUT.
                        ENDIF

                        IF NOT SOLOS    ;IF NOT PROC TECH SOLOS,
00F8 =          DISK    EQU  0F8H       ;DISK BASE ADDRESS.
                        ENDIF

                        IF SOLOS        ;IF PROC TECH SOLOS,
                DISK    EQU  0E8H       ;DIFFERENT DISK PORTS.
                        ENDIF

00F8 =          DCOM    EQU  DISK       ;DISK COMMAND PORT.
00F8 =          DSTAT   EQU  DISK       ;DISK STATUS PORT.
00F9 =          TRACK   EQU  DISK+1     ;DISK TRACK PORT.
00FA =          SECTP   EQU  DISK+2     ;DISK SECTOR PORT.
00FB =          DDATA   EQU  DISK+3     ;DISK DATA PORT.
00FC =          WAIT    EQU  DISK+4     ;DISK WAIT PORT.
00FC =          DCONT   EQU  DISK+4     ;DISK CONTROL PORT.
000A =          RTCNT   EQU  10         ;RETRY COUNT.

                        IF   STD        ;IF STANDARD I/O,
0001 =          CKBR    EQU  00000001B  ;KEYBOARD READY BIT.
0080 =          CPTR    EQU  10000000B  ;CONS OUTPUT RDY BIT.
                        ENDIF

                        IF   MSIO2      ;IF MITS 2SIO,
                CKBR    EQU  00000001B  ;KEYBOARD READY BIT.
                CPTR    EQU  00000010B  ;PRINT READY BIT.
                        ENDIF

                        IF   VDB8024            ;IF VDB-8024 BOARD.
                CKBR    EQU  00000010B          ;KEYBOARD READY BIT.
                CPTR    EQU  00000100B          ;CONS OUTPUT RDY BIT.
                        ENDIF
                ;
                        IF   ISIO2
                CKBR    EQU  00000010B          ;KEYBOARD READY BIT.
                CPTR    EQU  00000001B          ;PRINT READY BIT.
                        ENDIF
                ;
                        IF   TARDEL
                CKBR    EQU  00000010B          ;KEYBOARD READY BIT.
                CPTR    EQU  00000001B          ;PRINT READY BIT.
                        ENDIF
```

```
                        IF   TUART      ;IF CROMEMCO TUART,
                CKBR    EQU  01000000B  ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B  ;PRINT READY BIT.
                        ENDIF

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                CKBR    EQU  00000001B  ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B  ;DUMMY EQU.
                        ENDIF

                        IF OTHER        ;IF SOMETHING ELSE,
                CKBR    EQU  00000010B  ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B  ;PRINTER READY BIT.
                        ENDIF

0080 =          LRBIT   EQU  CPTR       ;LISTER READY BIT.
0003 =          IOBYTE  EQU  3                  ;ADDRESS OF I/O BYTE.
1000 =          CBASE   EQU  (MSIZE-20)*1024    ;BIAS FOR LARGER THAN 20K.
4400 =          CPMB    EQU  CBASE+3400H        ;START OF CPM 2.0
4C06 =          BDOS    EQU  CPMB+806H          ;START OF BDOS 2.0.
5A00 =          BIOS    EQU  CPMB+1600H         ;START OF CBIOS IO.
0004 =          CDISK   EQU  4                  ;LOCATION 4 IS CURRENT DISK.
0011 =          NSECTS  EQU  17                 ;NUMBER OF SECTORS IN IT.

5A00                    ORG  BIOS               ;START OF CBIOS STRUCTURE.
                ;
                ; I/O JUMP VECTOR
                ; THIS IS WHERE CPM CALLS WHENEVER IT NEEDS
                ; TO DO ANY INPUT/OUTPUT OPERATION.
                ; USER PROGRAMS MAY USE THESE ENTRY POINTS
                ; ALSO, BUT NOTE THAT THE LOCATION OF THIS
                ; VECTOR CHANGES WITH THE MEMORY SIZE.
                ;
5A00 C39C5A             JMP  BOOT               ;FROM COLD START LOADER.
5A03 C3EF5A     WBOOTE: JMP  WBOOT              ;FROM WARM BOOT.
5A06 C3235B             JMP  CONST              ;CHECK CONSOLE KB STATUS.
5A09 C32C5B             JMP  CONIN              ;READ CONSOLE CHARACTER.
5A0C C3385B             JMP  CONOT              ;WRITE CONSOLE CHARACTER.
5A0F C3E45C             JMP  LIST               ;WRITE LISTING CHAR.
5A12 C3DE5C             JMP  PUNCH              ;WRITE PUNCH CHAR.
5A15 C3DE5C             JMP  READER             ;READ READER CHAR.
5A18 C37B5B             JMP  HOME               ;MOVE DISK TO TRACK ZERO.
5A1B C3435B             JMP  SELDSK             ;SELECT DISK DRIVE.
5A1E C37D5B             JMP  SETTRK             ;SEEK TO TRACK IN REG A.
5A21 C3CB5B             JMP  SETSEC             ;SET SECTOR NUMBER.
5A24 C3D65B             JMP  SETDMA             ;SET DISK STARTING ADR.
5A27 C3FD5B             JMP  READ               ;READ SELECTED SECTOR.
5A2A C37D5C             JMP  WRITE              ;WRITE SELECTED SECTOR.
5A2D C3D75C             JMP  PRSTAT             ;LIST STATUS CHECK.
5A30 C3D05B             JMP  SECTRAN            ;SECTOR TRANSLATE ROUTINE.
                ;
                ; THIS SECTION DEFINES THE THE DISK PARAMETERS
                ; NOTE:
                ; IF YOU HAVE THE MACRO ASSEMBLER (MAC) FROM
                ; DIGITAL RESEARCH, YOU MAY ELIMINATE THIS SECTION
                ; STARTING AT **AA** TO **BB** AND USE THE MACRO
                ; FILE "DISKDEF" TO CUSTOM TAILOR YOUR SYSTEM TO
                ; ALLOW DIFFERENT TYPES OF DRIVES OR MORE THAN 4
                ; DRIVES.
                ;
                ;**AA**
                ;
5A33 =          DPBASE  EQU  $                  ;BASE OF DISK PARAMETER BLOCK
5A33 825A0000   DPE0:   DW   XLT0,0000H         ;TRANSLATE TABLE
5A37 00000000           DW   0000H,0000H        ;SCRATCH AREA
5A3B FE5C735A           DW   DIRBUF,DPB0        ;DIR BUFF, PARM BLOCK
5A3F 9D5D7E5D           DW   CSV0,ALV0          ;CHECK, ALLOC VECTORS
                ;
5A43 825A0000   DPE1:   DW   XLT1,0000H
5A47 00000000           DW   0000H,0000H
5A4B FE5C735A           DW   DIRBUF,DPB1
5A4F CC5DAD5D           DW   CSV1,ALV1
                ;
5A53 825A0000   DPE2:   DW   XLT2,0000H
5A57 00000000           DW   0000H,0000H
5A5B FE5C735A           DW   DIRBUF,DPB2
5A5F FB5DDC5D           DW   CSV2,ALV2
                ;
5A63 825A0000   DPE3:   DW   XLT3,0000H
5A67 00000000           DW   0000H,0000H
```

```
5A6B FE5C735A            DW   DIRBUF,DPB3
5A6F 2A5E0B5E            DW   CSV3,ALV3
                         ;
                         ;THE FOLLOWING DESCRIBES THE DISK PHYSICAL
                         ;NATURE, SUCH AS SECTORS/TRACK,DIRECTORY SIZE.
                         ;
5A73 =           DPB0    EQU  $                  ;ONE OF 4 DISK PARM. BLOCKS
5A73 1A00                DW   26                 ;SECTORS/TRACK
5A75 03                  DB   3                  ;BLOCK SHIFT
5A76 07                  DB   7                  ;BLOCK MASK
5A77 00                  DB   0                  ;EXTNT MASK
5A78 F200                DW   242                ;DISK SIZE - 1
5A7A 3F00                DW   63                 ;DIRECTORY MAX.
5A7C C0                  DB   192                ;ALLOC0
5A7D 00                  DB   0                  ;ALLOC1
5A7E 1000                DW   16                 ;CHECK SIZE
5A80 0200                DW   2                  ;NUMBER OF SYSTEM TRACKS
                         ;
                         ;SECTOR TRANSLATION TABLE
                         ;
5A82 =           XLT0    EQU  $                  ;START OF TRANS. TABLE
5A82 01070D1319          DB   1,7,13,19,25
5A87 050B111703          DB   5,11,17,23,3
5A8C 090F150208          DB   9,15,21,2,8
5A91 0E141A060C          DB   14,20,26,6,12
5A96 1218040A10          DB   18,24,4,10,16,22
                         ;
5A73 =           DPB1    EQU  DPB0               ;EQUIVALENT PARAMETERS
5A82 =           XLT1    EQU  XLT0               ;SAME TRANSLATE TABLE
                         ;
5A73 =           DPB2    EQU  DPB0
5A82 =           XLT2    EQU  XLT0
                         ;
5A73 =           DPB3    EQU  DPB0
5A82 =           XLT3    EQU  XLT0
                         ;
                         ;**BB**
                         ;
                         ; BOOT
                         ; THIS SECTION IS EXECUTED WHENEVER RESET AND RUN
                         ; IS PUSHED, AFTER THE COLDSTART LOADER READS IN
                         ; THE CPM SYSTEM.
                         ;
5A9C 318000      BOOT:   LXI  SP,80H             ;SET STACK POINTER.

                         IF  INTRP               ;IF INTERRUPTS ALLOWED,
                         EI                      ;ENABLE THEM HERE.
                         ENDIF

                         IF   STD                ;IF STANDARD I/O,
5A9F 00000000            DW   0,0                ;LEAVE SPACE FOR INIT.
5AA3 00000000            DW   0,0
5AA7 00000000            DW   0,0
5AAB 00000000            DW   0,0
                         ENDIF

                         IF   MSIO2              ;IF MITS 2SIO,
                         MVI  A,3                ;INITIALIZE 2SIO.
                         OUT  CCOM
                         OUT  LCOM
                         MVI  A,11H
                         OUT  CCOM
                         OUT  LCOM
                         ENDIF

                         IF TARDEL OR ISIO2
                         LXI  H,IOINIT           ;POINT TO 8251 BYTE TABLE.
                         MVI  B,4                ;THERE ARE 4 OF THEM.
                INITIO:  MOV  A,M                ;GET A BYTE FROM TABLE.
                         OUT  CCOM               ;OUT CONSOLE COMMAND PORT.
                         OUT  LCOM               ;OUT LIST COMMAND PORT.
                         INX  H                  ;POINT TO NEXT BYTE.
                         DCR  B                  ;DECREASE COUNT.
                         JNZ  INITIO             ;LOOP TILL DONE.
                         ENDIF

                         IF   TIMER AND TARBELL  ;IF USING TARBELL CPU
                         MVI  A,CNTR0+RLWORD+MODE2+BINARY        ;INIT 8253
                         OUT  TCMND              ;SEND IT TO COMMAND PORT
                         LXI  B,33333            ;TIME CONSTANT FOR 60 HZ
```

```
                        MOV  A,C
                        OUT  TCH0               ;LS BYTE OF COUNT
                        MOV  A,B
                        OUT  TCH0               ;MS BYTE OF COUNT
                        ENDIF

                        IF   TUART              ;IF CROMEMCO TUART,
                        MVI  A,1                ;SET A = 1.
                        OUT  54H                ;SELECT DEVICE A.
                        OUT  52H                ;RESET DEVICE B.
                        LXI  H,BAUDRS           ;GET ADR OF BAUD RATE TABLE.
                        MVI  A,11H              ;OCTUPLE THE CLOCK.
                IT1:    OUT  02H                ;& RESET CURRENT DEV.
                        MOV  A,M                ;GET BAUD RATE FROM TABLE.
                        INX  H                  ;INCREMENT POINTER.
                        OUT  0                  ;SET BAUD RATE.
                        CALL CONIN              ;READ KEYBOARD.
                        CALL CONIN              ;READ KEYBOARD AGAIN.
                        CPI  0DH                ;IF NOT CARRIAGE-RETURN,
                        MVI  A,1                ;SLOW THE CLOCK.
                        JNZ  IT1                ;UNTIL A CARRIAGE-RETURN.
                        ENDIF

                        IF   SOLOS              ;IF PROC TECH SOLOS,
                        CALL CLRSCR             ;CLEAR SCREEN.
                        ENDIF

5AAF AF                 XRA  A                  ;CLEAR SCRATCH AREA.
5AB0 320300             STA  IOBYTE             ;CLEAR I/O BYTE.
5AB3 320400             STA  CDISK              ;SELECT DRIVE ZERO
5AB6 0E08               MVI  C,ENDZ-STARTZ      ;GET LENGTH OF ZERO AREA.
5AB8 21F25C             LXI  H,STARTZ           ;GET SCRATCH ADDRESS.
5ABB 77         BOOTL:  MOV  M,A                ;PUT ZERO IN MEMORY.
5ABC 23                 INX  H                  ;INCREMENT POINTER.
5ABD 0D                 DCR  C                  ;DECREMENT COUNTER.
5ABE C2BB5A             JNZ  BOOTL              ;LOOP TILL DONE.
5AC1 3EF2               MVI  A,0F2H             ;SET LATCH CODE = F2.
5AC3 32FC5C             STA  LATCH
5AC6 DB01               IN   CDATA              ;CLEAR CONSOLE STATUS.
5AC8 21B65C             LXI  H,SMSG             ;PRINT OPENING MESSAGE.
5ACB CDA45C             CALL PMSG
                ;
                ; SET UP JUMPS INTO CP/M IN LOWER MEMORY.
                ;
5ACE 3EC3       SETUP:  MVI  A,0C3H             ;PUT JMP TO WBOOT
5AD0 320000             STA  0                  ;ADR AT ZERO.
5AD3 21035A             LXI  H,WBOOTE
5AD6 220100             SHLD 1
5AD9 320500             STA  5
5ADC 21064C             LXI  H,BDOS             ;PUT JUMP TO BDOS
5ADF 220600             SHLD 6                  ;AT ADR 5,6,7.
5AE2 218000             LXI  H,80H              ;SET DEFAULT DMA ADR.
5AE5 22F05C             SHLD DMAADD
5AE8 3A0400             LDA  CDISK              ;GET CURRENT DISK
5AEB 4F                 MOV  C,A                ;PASS IT TO CCP IN REG C.
5AEC C30044             JMP  CPMB               ;JUMP TO CCP.

                        IF   TARDEL OR ISIO2
                IOINIT: DB   0AAH,040H,0CEH,037H
                        ENDIF

                        IF   TUART              ;IF CROMEMCO TUART,
                BAUDRS: DB   94H,0CEH,0A2H,92H,88H,84H,82H,1
                        ENDIF
                ;
                ; WARM-BOOT:    READ CCP BACK INTO MEMORY.
                ; BDOS AND BIOS ASSUMED STILL IN MEMORY.
                ;
5AEF 318000     WBOOT:  LXI  SP,80H             ;SET STACK POINTER.

                        IF  INTRP               ;IF INTERRUPTS ALLOWED,
                        EI                      ;ALLOW THEM HERE.
                        ENDIF

                        IF   LSTPAG             ;IF LIST DEVICE PAGING,
                        XRA  A                  ;RESET LINE-FEED COUNT.
                        STA  LFCNT
                        ENDIF

5AF2 0E00               MVI  C,0                ;SELECT DISK ZERO.
```

```
5AF4 CD435B             CALL SELDSK
5AF7 CD7B5B             CALL HOME       ;MOVE TO TRACK ZERO.
5AFA C2185B             JNZ  RDERR      ;IF ERROR, PRINT MESSAGE.
5AFD 0611               MVI  B,NSECTS   ;GET # SECTORS FOR CPM READ.
5AFF 0E02               MVI  C,2        ;TRACK (B)=0, SECTOR (C)=2.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        DI              ;DISABLE THEM HERE.
                        ENDIF

5B01 210044             LXI  H,CPMB     ;GET STARTING ADDRESS.
5B04 22F05C     RBLK1:  SHLD DMAADD     ;SET STARTING ADDRESS.
5B07 CDCB5B             CALL SETSEC     ;READ STARTING AT SECTOR IN C.
5B0A CDFD5B             CALL READ       ;READ SYSTEM SECTORS.
5B0D C2185B             JNZ  RDERR      ;IF ERROR, PRINT MESSAGE.
5B10 0C                 INR  C          ;INCREMENT SECTOR NUMBER.
5B11 05                 DCR  B          ;DECREMENT SECTOR COUNT.
5B12 C2045B             JNZ  RBLK1      ;NOT ZERO, KEEP READING.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        EI              ;ALLOW THEM AGAIN HERE.
                        ENDIF

5B15 C3CE5A             JMP  SETUP      ;SET UP JUMPS.
                        ;
5B18 0E42       RDERR:  MVI  C,'B'      ;GET BOOT ERROR CODE.
5B1A CD385B             CALL CONOT      ;PRINT IT.
5B1D CD2C5B             CALL CONIN      ;READ A CHAR FROM CONSOLE.
5B20 C3EF5A             JMP  WBOOT      ;DO A WARM BOOT.
                        ;
                        ; CHECK CONSOLE INPUT STATUS.
                        ;
5B23 DB00       CONST:  IN   CSTAT      ;READ CONSOLE STATUS.
5B25 E601               ANI  CKBR       ;LOOK AT KB READY BIT.
5B27 3E00       CONST1: MVI  A,0        ;SET A=0 FOR RETURN.

                        IF   RDYLO      ;IF STATUS READY LOW,
5B29 C0                 RNZ             ;NOT READY WHEN NOT 0.
                        ENDIF

                        IF   RDYHI      ;IF STATUS READY HIGH,
                        RZ              ;NOT READY WHEN ZERO.
                        ENDIF

5B2A 2F                 CMA             ;IF READY A=FF.
5B2B C9                 RET             ;RETURN FROM CONST.
                        ;
                        ; READ A CHARACTER FROM CONSOLE.
                        ;
                        CONIN:
                        IF NOT SOLOS    ;IF NOT PROC TECH SOLOS,
5B2C DB00               IN   CSTAT      ;READ CONSOLE STATUS.
5B2E E601               ANI  CKBR       ;IF NOT READY,
                        ENDIF

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                        CALL KBD        ;READ SOL KEYBOARD.
                        JZ   CONIN      ;READY WHEN NOT ZERO.
                        ENDIF

                        IF RDYLO AND NOT SOLOS
5B30 C22C5B             JNZ  CONIN      ;LOOP UNTIL LOW.
                        ENDIF

                        IF   RDYHI      ;IF READY WHEN HIGH,
                        JZ   CONIN      ;LOOP UNTIL HIGH.
                        ENDIF

                        IF NOT SOLOS    ;IF NOT PROC TECH SOLOS,
5B33 DB01               IN   CDATA      ;READ A CHARACTER.
                        ENDIF

5B35 E67F               ANI  7FH                ;MAKE MOST SIG. BIT = 0.
5B37 C9                 RET                     ;RETURN FROM CONIN.
                        ;
                        ; WRITE A CHARACTER TO THE CONSOLE DEVICE.
                        ;
                        CONOT:
                        IF CONUL                ;IF NULLS REQUIRED,
                        MVI  A,0DH              ;IF IT'S A CR,
```

```
                        CMP  C                  ;THEN HOP OUT
                        JZ   CONULL             ;TO NULL ROUTINE.
                        ENDIF
                CONOT1:
                        IF NOT SOLOS AND NOT VIDEO
5B38 DB00               IN   CSTAT              ;READ CONSOLE STATUS.
5B3A E680               ANI  CPTR               ;IF NOT READY,
                        ENDIF

                        IF RDYLO AND NOT SOLOS AND NOT VIDEO
5B3C C2385B             JNZ  CONOT1             ;LOOP UNTIL LOW.
                        ENDIF

                        IF RDYHI                ;IF READY WHEN HIGH,
                        JZ   CONOT1             ;LOOP UNTIL HIGH.
                        ENDIF

                        IF NOT SOLOS AND NOT VIDEO
5B3F 79                 MOV  A,C                ;GET CHARACTER.
5B40 D301               OUT  CDATA              ;PRINT IT.
5B42 C9                 RET                     ;RETURN.
                        ENDIF
                ;
                ;THIS ROUTINE CALLES YOUR VIDEO DRIVER ROUTINE WHICH
                ;MUST BE IN ROM. ALL REGISTERS MUST BE SAVED AND RESTORED
                ;BY YOUR VIDEO DRIVER IN ORDER TO BE COMPATIABLE WITH CPM.
                ;CPM PASSES THE CHAR. TO BE OUTPUT IN THE  C REGISTER.
                ;MAKE ANY CHANGES IN THIS ROUTINE TO PASS THE CHAR FROM
                ;REG C TO THE REGISTER YOUR VIDEO DRIVER EXPECTS IT TO BE IN.
                ;
                        IF  VIDEO               ;IF USING A VIDEO DRIVER IN ROM.
                        MOV  A,C                ;GET THE CPM CHAR INTO REG A
                        CALL OUTADDR            ;CALL YOUR VIDEO DRIVER.
                        RET                     ;RETURN TO CPM.
                        ENDIF

                        IF CONUL
                CONULL: PUSH B                  ;SAVE B&C.
                        MVI  B,CNULL+1          ;GET NULL COUNT.
                CONUL1: CALL CONOT1             ;PRINT CR.
                        MVI  C,0                ;GET NULL CHAR.
                        DCR  B                  ;DECREMENT COUNTER.
                        JNZ  CONUL1             ;DO NEXT NULL.
                        POP  B                  ;RESTORE B&C.
                        MOV  A,C                ;RESTORE A.
                        RET                     ;RETURN.
                        ENDIF

                        IF   SOLOS              ;IF PROC TECH SOLOS,
                        PUSH B                  ;SAVE B&C.
                        MOV  B,C                ;PUT CHAR IN B REG.
                        CALL SCRN               ;OUTPUT CHAR TO SOLOS.
                        POP  B                  ;RESTORE B&C.
                        MOV  A,C                ;PUT CHAR IN A.
                        RET                     ;RETURN FROM CONOT.
                        ENDIF
                ;
                ; SELECT DISK NUMBER ACCORDING TO REGISTER C.
                ;
5B43 210000     SELDSK: LXI  H,0                ;SET UP FOR ERROR CODE
5B46 79                 MOV  A,C                ;GET NEW DRIVE.
5B47 FE04               CPI  NDISK              ;CALLING UNDEFINED DRIVE ?
5B49 D0                 RNC                     ;IF NO CY, H,L TELLS CPM YES.
5B4A 21F25C             LXI  H,DISKNO           ;GET OLD DRIVE NUMBER.
5B4D 7E                 MOV  A,M        ;GET OLD DISK NUMBER.

                        IF   DUAL       ;IF DUAL DRIVE,
                        ANI  0FEH       ;CLEAR OUT BIT 0.
                        ENDIF

5B4E 5F                 MOV  E,A        ;PUT OLD DISK NO. IN D&E.
5B4F 1600               MVI  D,0
5B51 21F65C             LXI  H,TRTAB    ;GET ADDRESS OF TRACK TABLE.
5B54 E5                 PUSH H          ;SAVE ADDRESS OF TRTAB.
5B55 19                 DAD  D          ;ADD DISK NO. TO ADDRESS.
5B56 DBF9               IN   TRACK      ;READ 1771 TRACK REGISTER.
5B58 77                 MOV  M,A        ;PUT INTO TABLE.
5B59 79                 MOV  A,C        ;GET NEW DISK NUMBER.

                        IF   DUAL       ;IF A DUAL DRIVE,
```

```
                        ANI  0FEH        ;CLEAR BIT 0.
                        ENDIF

5B5A 5F                 MOV  E,A         ;PUT NEW DISK NO. IN D&E.
5B5B E1                 POP  H           ;RESTORE ADDRESS OF TRTAB.
5B5C 19                 DAD  D           ;ADD DISK NO. TO ADDRESS.
5B5D 7E                 MOV  A,M         ;GET NEW TRACK NUMBER.
5B5E D3F9               OUT  TRACK       ;PUT INTO 1771 TRACK REG.
5B60 79                 MOV  A,C         ;UPDATE OLD DISK NUMBER.
5B61 32F25C             STA  DISKNO
5B64 2F                 CMA              ;BITS INVERTED INTO LATCH.
5B65 87                 ADD  A           ;PUT BITS 1&2 AT 4&5.
5B66 87                 ADD  A
5B67 87                 ADD  A
5B68 87                 ADD  A
5B69 F602               ORI  2           ;MAKE LATCH COMMAND.
5B6B 32FC5C             STA  LATCH       ;SAVE NEW LATCH CODE.
                ;
                ;SELECT DRIVE AS A FUNCTION OF H,L
                ;
5B6E 2AF25C             LHLD DISKNO              ;LOAD DISK NUMBER AND ZERO BYTE
5B71 11335A             LXI  D,DPBASE            ;POINT TO DISK PARM START.
5B74 29                 DAD  H                   ;*2
5B75 29                 DAD  H                   ; *4
5B76 29                 DAD  H                   ;  *8
5B77 29                 DAD  H                   ;   *16
5B78 19                 DAD  D                   ;COMPUTE INDEX FOR THE DRIVE
5B79 AF                 XRA  A           ;SET A = 0.
5B7A C9                 RET              ;RETURN FROM SELDSK.
                ;
                ; MOVE DISK TO TRACK ZERO.
                ;
5B7B 0E00       HOME:   MVI  C,0         ;SEEK TO TRACK ZERO.
                ;
                ; SET TRACK NUMBER TO WHATEVER IS IN REGISTER C.
                ; ALSO PERFORM MOVE TO THE CORRECT TRACK (SEEK).
                ;
                SETTRK:
                        IF NOT DUBSID    ;IF NOT DOUBLE-SIDED,
5B7D E5                 PUSH H           ;SAVE H&L.
5B7E 2AFC5C             LHLD LATCH       ;GET NEW & OLD LATCH.
5B81 7D                 MOV  A,L         ;GET NEW LATCH.
5B82 D3FC               OUT  DCONT       ;SELECT DRIVE NOW.
5B84 32FD5C             STA  CLATCH      ;REMEMBER CURRENT LATCH.
5B87 BC                 CMP  H           ;IS NEW SAME AS OLD?
5B88 3EFF               MVI  A,0FFH      ;IF NOT, SET FLAG = FF.
5B8A C28E5B             JNZ  SFLAG
5B8D 2F                 CMA              ;IF NEW = OLD, FLAG = 0.
5B8E 32F45C     SFLAG:  STA  HLSF        ;SET HEAD-LOAD/SELECT FLAG.
5B91 E1                 POP  H           ;RESTORE H&L.
                        ENDIF

5B92 79                 MOV  A,C         ;GET NEW TRACK NUMBER.

                        IF   DUBSID      ;IF DOUBLE-SIDED DISK,
                        RRC              ;SHIFT RIGHT ONCE.
                        PUSH PSW         ;SAVE REVISED TRACK NUMBER.
                        PUSH H           ;SAVE H&L.
                        LHLD LATCH       ;GET NEW & OLD LATCH.
                        MOV  A,L         ;PUT NEW LATCH IN A.
                        JC   SIDE2       ;IF TRACK # ODD, SIDE #2.
                        ORI  40H         ;CLEAR LATCH BIT FOR SIDE 1.
                        JMP  SETLAT      ;GO AHEAD AND SET LATCH.
                SIDE2:  ANI  0B2H        ;SET LATCH BIT FOR SIDE 2.
                SETLAT: OUT  DCONT       ;OUTPUT TO THE LATCH.
                        STA  CLATCH      ;SET CURRENT LATCH CODE.
                        XRA  H           ;COMPARE OLD WITH NEW,
                        ANI  0BFH        ;IGNORE SIDE BIT.
                        MVI  A,0FFH      ;IF OLD NOT = NEW,
                        JNZ  SETFL       ;  SET FLAG = FF.
                        CMA  A           ;IF OLD = NEW, FLAG = 0.
                SETFL:  STA  HLSF        ;SET FLAG ZERO IF SAME.
                        POP  H           ;RESTORE H&L.
                        POP  PSW         ;RESTORE THE TRACK NUMBER.
                        ANI  7FH         ;CLEAR MSB.
                        MOV  C,A         ;SET C=TRACK NUMBER.
                        ENDIF

5B93 32EE5C             STA  TRK         ;UPDATE OLD WITH NEW.
                ;
```

```
                        ; MOVE THE HEAD TO THE TRACK IN REGISTER A.
                        ;
5B96 C5         SEEK:   PUSH B          ;SAVE B&C.
5B97 47                 MOV  B,A        ;SAVE DESTINATION TRACK.
5B98 3E0A               MVI  A,RTCNT    ;GET RETRY COUNT.
5B9A 32FB5C     SRETRY: STA  SERCNT     ;STORE IN ERROR COUNTER.
5B9D DBF9               IN   TRACK      ;READ PRESENT TRACK NO.
5B9F B8                 CMP  B          ;SAME AS NEW TRACK NO.?
5BA0 C2A55B             JNZ  NOTHR      ;JUMP IF NOT THERE.
5BA3 C1         THERE:  POP  B          ;RESTORE B&C.
5BA4 C9                 RET             ;RETURN FROM SEEK.
                        ;
                        ;THIS ROUTINE IS TO ALLOW TIME FOR THE DRIVE
                        ;TUNNEL ERASE TO TERMINATE BEFORE MOVING THE
                        ;HEAD.  THE DELAY IS APPROX. 700 MICRO-SEC. @
                        ;4 MHZ CPU TIME, AND DOUBLE THIS FOR 2 MHZ CPU'S.
                        ;
5BA5 F5         NOTHR:  PUSH PSW        ;SAVE ACCUM
5BA6 3ED0               MVI   A,0D0H    ;DELAY COUNT = 208
5BA8 3D         BUSY1:  DCR   A         ;DECREASE COUNT
5BA9 C2A85B             JNZ   BUSY1     ;LOOP TILL DONE
5BAC F1                 POP   PSW       ;RESTORE ACCUM

                        IF NOT PERSCI   ;IF NOT PERSCI DRIVE,
5BAD 78                 MOV  A,B        ;RESTORE A FROM B.
5BAE D3FB               OUT  DDATA      ;TRACK TO DATA REGISTER.
5BB0 3E16               MVI  A,14H+STPRAT+HLAB  ;GET STEP RATE, DO
5BB2 D3F8               OUT  DCOM       ;SEEK WITH VERIFY.
5BB4 DBFC               IN   WAIT       ;WAIT FOR INTRQ.
5BB6 DBF8               IN   DSTAT      ;READ STATUS.
5BB8 E691               ANI  91H        ;LOOK AT BITS.
5BBA CAA35B             JZ  THERE       ;OK IF ZERO.
                        ENDIF

                        IF  PERSCI      ;IF PERSCI DRIVE,
                        MVI  A,40H+HLAB  ;IF CARRY = 1,
                        JC  SDIR        ;STEP IN.
                        MVI  A,60H+HLAB  ;OTHERWISE, OUT.
                SDIR:   OUT  DCOM       ;ISSUE STEP DIRECTION.
                        MVI  A,20       ;DELAY LOOP COUNT.
                DLOOP:  DCR  A          ;DECREMENT COUNTER.
                        JNZ  DLOOP
                        MOV  A,C        ;GET PRESENT TRACK.
                        SUB  B          ;FIGURE TRACKS TO STEP.
                        JP   STEP       ;IF NEGATIVE,
                        CMA             ;FIGURE THE
                        INR  A          ;TWO'S COMPLEMENT.
                STEP:   MOV  C,A        ;GET DIFFERENCE.
                        MVI  A,1        ;PERSCI STEP COMMAND.
                STEP1:  OUT  DCONT      ;STEP PERSCI (E-14).
                        DCR  C          ;COUNT THE STEP.
                        JNZ  STEP1      ;STEP UNTIL C = 0.
                        IN   WAIT       ;CLEAR 1771.
                        IN   DSTAT
                        MOV  A,B        ;GET DEST. TRACK.
                        OUT  TRACK      ;UPDATE TRACK REG.
                        LDA  CLATCH     ;GET LATCH CODE.
                        ANI  72H        ;MAKE COMMAND TO
                        OUT  DCONT      ;SWITCH WAIT FOR
                        IN   WAIT       ;SEEK COMPLETE.
                        LDA  CLATCH     ;RESTORE LATCH TO
                        OUT  DCONT      ;OLD CODE.
                        XRA  A          ;MAKE GOOD RETURN.
                        POP  B          ;RESTORE  B&C.
                        RET
                        ENDIF

                        IF NOT PERSCI   ;IF NOT PERSCI DRIVE,
5BBD 3AFB5C             LDA  SERCNT     ;GET ERROR COUNT.
5BC0 3D                 DCR  A          ;DECREMENT COUNT.
5BC1 C29A5B             JNZ  SRETRY     ;RETRY SEEK.
5BC4 C1                 POP  B          ;RESTORE B&C.
5BC5 CD245C             CALL RECOV      ;IF SEEK RETRY = 10 CHECK
5BC8 C3965B             JMP  SEEK       ; FOR CNTL-C FOR ABORT.
                        ENDIF
                        ;
                        ; SET DISK SECTOR NUMBER.
                        ;
5BCB 79         SETSEC:  MOV  A,C       ;GET SECTOR NUMBER.
5BCC 32EF5C              STA  SECT      ;PUT AT SECT # ADDRESS.
```

```
5BCF C9                 RET             ;RETURN FROM SETSEC.
                        ;
                        ;TRANSLATE THE SECTOR GIVEN B,C USING THE
                        ;TRANSLATE TABLE GIVEN BY D,E
                        ;
5BD0 EB         SECTRAN:XCHG                    ;H,L = TRANS
5BD1 09                 DAD  B                  ;H,L = TRANS (SECTOR)
5BD2 6E                 MOV  L,M                ;L   = TRANS (SECTOR)
5BD3 2600               MVI  H,0                ;CLEAR REG H
5BD5 C9                 RET                     ;H,L = TRANSLATED SECTOR
                        ;
                        ; SET DISK DMA ADDRESS.
                        ;
5BD6 60         SETDMA: MOV  H,B        ;MOVE B&C TO H&L.
5BD7 69                 MOV  L,C
5BD8 22F05C             SHLD DMAADD     ;PUT AT DMA ADR ADDRESS.
5BDB C9                 RET             ;RETURN FROM SETDMA.
                        ;
                        ; HDLD - GET HEAD-LOAD BIT IF REQUIRED.
                        ;
5BDC 3AF45C     HDLD:   LDA  HLSF       ;GET HEAD-LOAD FLAG.
5BDF B7                 ORA  A          ;IS A = ZERO?
5BE0 CAF45B             JZ   HDLD1      ;HOP IF SO.
5BE3 2F                 CMA             ;SET A = 0.
5BE4 32F45C             STA  HLSF       ;SET FLAG = 0 IF NOT.
                        ;
                        ;IF CHANGING TO A NEW DRIVE, PERFORM A SEEK TO
                        ;THE SAME TRACK TO ALLOW THE HEAD TO UNLOAD.
                        ;
5BE7 DBF9               IN   TRACK      ;GET PRESENT TRACK
5BE9 D3FB               OUT  DDATA      ;AND TELL 1771 ABOUT IT.
5BEB 3E16               MVI  A,14H+STPRAT+HLAB ;GET THE STEP RATE.
5BED D3F8               OUT  DCOM       ;SEND IT TO FLOPPY CONTROLLER.
5BEF DBFC               IN   WAIT       ;WAIT FOR INTRQ.
5BF1 3E04       HDLDY:  MVI  A,4        ;SET BIT TO LOAD HEAD.
5BF3 C9                 RET             ;RETURN FROM HDLD.
5BF4 DBF8       HDLD1:  IN   DSTAT      ;READ 1771 STATUS.
5BF6 E620               ANI  20H        ;LOOK AT HL BIT.
5BF8 CAF15B             JZ   HDLDY      ;LOAD IF NOT LOADED.
5BFB AF                 XRA  A          ;OTHERWISE, A=0.
5BFC C9                 RET             ;RETURN FROM HDLD.
                        ;
                        ; READ THE SECTOR AT SECT, FROM THE PRESENT TRACK.
                        ; USE STARTING ADDRESS AT DMAADD.
                        ;
5BFD 3E0A       READ:   MVI  A,RTCNT    ;GET RETRY COUNT.
5BFF CD695C     RRETRY: CALL DSKSET     ;SET UP DISK.
5C02 C688               ADI  88H        ;ADD CODE FOR READ SECT.
5C04 D3F8       READE:  OUT  DCOM       ;SEND COMMAND TO 1771.
5C06 DBFC       RLOOP:  IN   WAIT       ;WAIT FOR DRQ OR INTRQ.
5C08 B7                 ORA  A          ;SET FLAGS.
5C09 F2135C             JP   RDDONE     ;DONE IF INTRQ.
5C0C DBFB               IN   DDATA      ;READ A DATA BYTE FROM DISK.
5C0E 77                 MOV  M,A        ;PUT BYTE INTO MEMORY.
5C0F 23                 INX  H          ;INCREMENT MEMORY POINTER.
5C10 C3065C             JMP  RLOOP      ;KEEP READING.
5C13 DBF8       RDDONE: IN   DSTAT      ;READ DISK STATUS.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        EI              ;ALLOW AGAIN HERE.
                        ENDIF

5C15 E69D               ANI  9DH        ;LOOK AT ERROR BITS.
5C17 C8                 RZ              ;RETURN IF NONE.
5C18 CD335C             CALL ERCHK      ;CHECK FOR SEEK ERROR.
5C1B C2FF5B             JNZ  RRETRY     ;TRY TO READ AGAIN.
5C1E CD245C             CALL RECOV      ;CHECK FOR ABORT OR CONTINUE
5C21 C3FD5B             JMP  READ       ;IF NOT CNTL-C, TRY TO READ AGAIN.
                        ;
                        ;RECOV
                        ;THIS ROUTINE IS CALLED BY ANY READ,WRITE,SEEK ROUTINE IF THE
                        ;RETRY COUNT GOES TO 10. IF IT DOES,THIS ROUTINE CALLS CONIN
                        ;FOR A KEY TO BE PUSHED. IF THE KEY IS A CNTL-C, THEN A WARMBOOT
                        ;IS EXECUTED. IF ANY OTHER KEY IS PUSHED, THEN A RETURN IS MADE
                        ;BACK TO THE CALLER AND THAT ROUTINE IS RETRIED FOR 10 MORE TIMES.
                        ;
5C24 21AF5C     RECOV:  LXI  H,ERRMSG   ;POINT TO "ERROR" MESSAGE.
5C27 CDA45C             CALL PMSG       ;PRINT IT ON CONSOL.
5C2A CD2C5B             CALL CONIN      ;CHECK FOR PUSHED KEY.
5C2D FE03               CPI  03H        ;IS IT A CNTL-C ?
```

```
5C2F C0                 RNZ             ;RETURN TO CALLER IF NOT.
5C30 C3EF5A             JMP  WBOOT      ;YES, DO WARMBOOT.
                        ;
                        ; ERCHK - CHECK FOR RECORD NOT FOUND ERROR.
                        ;
5C33 E610       ERCHK:  ANI  10H        ;IF RECORD NOT FOUND,
5C35 C23D5C             JNZ  CHKSK      ;DO A CHECK ON SEEK.
5C38 3AFA5C     CHKOK:  LDA  ERCNT      ;GET ALLOWED ERROR COUNT
5C3B 3D                 DCR  A
5C3C C9                 RET             ;AND RETURN.
                        ;
                        ;CHECK FOR SEEK TO CORRECT TRACK, AND CHANGE IF NECESSARY.
                        ;
5C3D 3EC4       CHKSK:  MVI  A,0C4H     ;SEND COMMAND TO 1771
5C3F D3F8               OUT  DCOM       ;TO READ ADDRESS.
5C41 DBFC               IN   WAIT       ;WAIT FOR DRQ OR INTRQ.
5C43 DBFB               IN   DDATA      ;READ THE TRACK ADDRESS.
5C45 47                 MOV  B,A        ;SAVE IN REGISTER B.
5C46 DBFC       CHKS2:  IN   WAIT       ;WAIT FOR INTRQ.
5C48 B7                 ORA  A          ;SET FLAGS.
5C49 F2515C             JP   CHKS3      ;DONE WITH READ ADR OP.
5C4C DBFB               IN   DDATA      ;READ ANOTHER BYTE.
5C4E C3465C             JMP  CHKS2      ;DO IT AGAIN.
5C51 DBF8       CHKS3:  IN  DSTAT       ;READ DISK STATUS.
5C53 B7                 ORA  A          ;SET FLAGS.
5C54 CA5D5C             JZ   CHKS4      ;READ ADR OK IF 0.
5C57 CD7B5B             CALL HOME       ;OTHERWISE, HOME FIRST.
5C5A C3605C             JMP  CHKS5
5C5D 78         CHKS4:  MOV  A,B        ;UPDATE TRACK REGISTER.
5C5E D3F9               OUT  TRACK
5C60 3AEE5C     CHKS5:  LDA  TRK        ;GET REQUIRED TRACK NO.
5C63 CD965B             CALL SEEK       ;MOVE THE HEAD TO IT.
5C66 C3385C             JMP  CHKOK
                        ;
                        ;DISK SET UP ROUTINE. THIS ROUTINE IS COMMON TO BOTH THE
                        ;READ AND WRITE ROUTINES.
                        ;
5C69 32FA5C     DSKSET: STA  ERCNT      ;STORE IN ERROR CTR.
5C6C 3ED0               MVI  A,0D0H     ;CAUSE INTERRUPT.
5C6E D3F8               OUT  DCOM
5C70 E3                 XTHL            ;SOME
5C71 E3                 XTHL            ; DELAY

                        IF  INTRP       ;IF INTERRUPTS ALLOWED,
                        DI              ;DISABLE THEM HERE.
                        ENDIF

5C72 2AF05C             LHLD DMAADD     ;GET STARTING ADDR.
5C75 3AEF5C             LDA  SECT       ;GET SECTOR NUMBER.
5C78 D3FA               OUT  SECTP      ;SET SECTOR INTO 1771.
5C7A C3DC5B             JMP  HDLD       ;GET HEAD-LOAD BIT?
                        ;
                        ; WRITE THE SECTOR AT SECT, ON THE PRESENT TRACK.
                        ; USE STARTING ADDRESS AT DMAADD.
                        ;
5C7D 3E0A       WRITE:  MVI  A,RTCNT    ;GET RETRY COUNT.
5C7F CD695C     WRETRY: CALL DSKSET     ;SET UP DISK.
5C82 C6A8               ADI  0A8H       ;ADD CODE FOR WRITE.
5C84 D3F8       WRITE2: OUT  DCOM
5C86 DBFC       WLOOP:  IN   WAIT       ;WAIT FOR READY.
5C88 B7                 ORA  A          ;SET FLAGS.
5C89 F2935C             JP   WDONE      ;HOP OUT WHEN DONE.
5C8C 7E                 MOV  A,M        ;GET BYTE FROM MEM.
5C8D D3FB               OUT  DDATA      ;WRITE ONTO DISK.
5C8F 23                 INX  H          ;INCREMENT MEM PTR.
5C90 C3865C             JMP  WLOOP      ;KEEP WRITING.
5C93 DBF8       WDONE:  IN   DSTAT      ;READ DISK STATUS.

                        IF  INTRP       ;IF INTERRUPTS ALLOWED,
                        EI              ;ENABLE AGAIN HERE.
                        ENDIF

5C95 E6FD               ANI  0FDH       ;LOOK AT THESE BITS.
5C97 C8                 RZ              ;RETURN IF NO ERR.
5C98 CD335C             CALL ERCHK      ;CHECK/CORRECT SEEK ERR.
5C9B C27F5C             JNZ  WRETRY     ;TRY TO WRITE AGAIN.
5C9E CD245C             CALL RECOV      ;CHECK FOR ABORT
5CA1 C37D5C             JMP  WRITE      ;RETRY WRITE AGAIN.
                        ;
                        ; PRINT THE MESSAGE AT H&L UNTIL A ZERO.
```

```
                        ;
5CA4 7E         PMSG:   MOV  A,M        ;GET A CHARACTER.
5CA5 B7                 ORA  A          ;IF IT'S ZERO,
5CA6 C8                 RZ              ;RETURN.
5CA7 4F                 MOV  C,A        ;OTHERWISE,
5CA8 CD385B             CALL CONOT      ;PRINT IT.
5CAB 23                 INX  H          ;INCREMENT H&L,
5CAC C3A45C             JMP  PMSG       ;AND GET ANOTHER.
                        ;
                        ; CBIOS MESSAGES
                        ;
5CAF 4552524F52ERRMSG: DB   'ERROR.',0
5CB6 0D0A546172SMSG:   DB   0DH,0AH,'Tarbell '
5CC0 3234              DB   MSIZE/10+'0',MSIZE MOD 10 + '0'
5CC2 4B2043504D        DB   'K CPM 2.2 of 7-16-80',0
                        ;
                        ;LIST STATUS CHECK ROUTINE
                        ;
5CD7 CDDF5C     PRSTAT: CALL PSTAT              ;CHECK PRINTER STATUS PORT.
5CDA 3E00               MVI  A,0                ;RETURN STATUS ACTIVITY.
                        ;
                        IF   TARDEL OR RDYLO
5CDC C0                 RNZ
                        ENDIF
                        ;
                        IF   NOT TARDEL AND RDYHI
                        RZ
                        ENDIF
                        ;
5CDD 2F                 CMA                     ;INVERT IT
                        ;
                        ; PUNCH AND READER ARE NOT SUPPORTED.
                        ;
                        PUNCH:
5CDE C9         READER: RET
                        ;
                        ;PSTAT - PRINTER STATUS CHECK ROUTINE.
                        ;
5CDF DB02       PSTAT:  IN   LSTAT              ;READ PRINTER STATUS PORT.
                        ;
                        IF   NOT TARDEL
5CE1 E680               ANI  LRBIT
                        ENDIF
                        ;
                        IF   TARDEL
                        ANI  81H                ;MASK READY BITS
                        XRI  81H
                        ENDIF
                        ;
5CE3 C9                 RET                     ;RETURN TO CALLER
                        ;
                        ; WRITE A CHARACTER ON LISTING DEVICE.
                        ;
                        LIST:
                        IF   LSTNUL             ;IF NULLS OR PAGING,
                        MVI  A,0DH              ;IF IT'S A CR,
                        CMP  C                  ;THEN HOP OUT TO
                        JZ   LINUL              ;NULL ROUTINE.
                        ENDIF
                        ;
                        IF   LSTPAG             ;IF PAGING
                        MVI  A,0AH              ;GET A LINEFEED
                        CMP  C                  ;DOES IT MATCH?
                        JZ   LINUL3
                        MOV  A,C
                        CPI  0CH
                        RZ
                        ENDIF
                        ;
5CE4 CDDF5C     LTBSY:  CALL PSTAT              ;READ LISTER STATUS.
                        ;
                        IF   TARDEL OR RDYLO
5CE7 C2E45C             JNZ  LTBSY              ;LOOP TILL LOW.
                        ENDIF
                        ;
                        IF   NOT TARDEL AND RDYHI
                        JZ   LTBSY              ;LOOP TILL HIGH.
                        ENDIF
                        ;
5CEA 79                 MOV  A,C                ;GET DATA BYTE.
```

```
                        ;
                        ; TARBELL ELECTRONICS
                        ; CP/M COLDSTART LOADER
                        ; VERSION OF 7-16-80.
                        ;
                        ; THIS BOOTSTRAP NOW SUPPORTS THE NEW TARBELL Z-80 CPU BOARD.
                        ; THIS PROGRAM IS LOADED AT LOCATION ZERO BY THE BOOTSTRAP PROGRAM,
                        ; AND EXECUTED. ITS PURPOSE IS TO LOAD AND EXECUTE THE CP/M DISK
                        ; OPERATING SYSTEM AT THE TOP OF THE MEMORY IN USE.
                        ;
0000 =          FALSE   EQU  0          ;DEFINE VALUE OF FALSE.
FFFF =          TRUE    EQU  NOT FALSE  ;DEFINE VALUE OF TRUE.
                        ;
                        ;********* THIS IS THE AREA TO MAKE CHANGES IN ******************
                        ;********* FOR DIFFERENT SYSTEM CONFIGURATIONS ******************
                        ;                                                             **
0018 =          MSIZE   EQU  24         ;MEMORY SIZE IN DECIMAL KB.                   **
0000 =          TARBELL EQU  FALSE      ;TRUE IF USING TARBELL CPU.                   **
0000 =          DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED SYSTEMS.               **
0000 =          PERSCI  EQU  FALSE      ;TRUE FOR PERSCI DRIVES                       **
0000 =          DELTA   EQU  FALSE      ;TRUE IF USING DELTA CPU CARD                 **
0000 =          BASE    EQU  0          ;TARBELL I/O PORTS (00 or 10 HEX)             **
001A =          SPT     EQU  26         ;NUMBER OF SECTORS PER TRACK.                 **
00F8 =          DISK    EQU  0F8H       ;DISK PORT BASE ADDRESS.                      **
                        ;                                                             **
                        ;*************************************************************
                        ;
                                IF   TARBELL
                        IO      EQU  BASE       ;i/o ports on tarbell cpu.
                        MMENB   EQU  IO+10      ;memory management enable port.
                        MEMMAG  EQU  BASE+32    ;memory management port.
                                ENDIF

00F8 =          DCOM    EQU  DISK       ;COMMAND PORT.
00F8 =          DSTAT   EQU  DISK       ;STATUS PORT.
00F9 =          TRACK   EQU  DISK+1     ;TRACK PORT.
00FA =          SECT    EQU  DISK+2     ;SECTOR PORT.
00FB =          DATA    EQU  DISK+3     ;DATA PORT.
00FC =          WAIT    EQU  DISK+4     ;WAIT PORT.
00FC =          DCONT   EQU  DISK+4     ;CONTROL PORT.
1000 =          CBASE   EQU  (MSIZE-20)*1024
4400 =          CPMB    EQU  CBASE+3400H;START OF CP/M.
5A00 =          BOOTE   EQU  CPMB+1600H ;COLD BOOT ENTRY POINT.
0033 =          NSECTS  EQU  51         ;SECTORS OF CP/M.
000A =          RTCNT   EQU  10         ;NUMBER OF RETRYS.

0000            BOOT:   ORG  0          ;START OF LOADER.

                        IF   TARBELL    ;IF USING TARBELL CPU
                        OUT  MMENB      ;ENABLE MEMORY MANAGEMENT.
                        LXI  D,1000H    ;COUNT=16, DATA BYTE = 0
                        MVI  C,MEMMAG AND 0FFH
                MLOOP:  MOV  A,E        ;GET ADDRESS VALUE
                        ORA  C          ;MAKE I/O PORT VALUE
                        STA  MOUT+1     ;MODIFY PORT ON THE FLY
                        MOV  A,E        ;GET INIT VALUE.
                        CMA             ;FLIP IT FOR RAM ON CPU
                MOUT:   OUT  BASE       ;PUT IT TO RAM ON CPU
                        INR  E          ;BUMP DATA VALUE
                        DCR  D          ;DECREASE COUNT
                        JNZ  MLOOP      ;LOOP 16 TIMES.
                        ENDIF

                        IF  DELTA       ;IF USING DELTA PRODUCTS CPU.
                        MVI  A,1        ;CODE TO KICK OUT ROM
                        OUT  9
                        ENDIF

0000 1E0A               MVI  E,RTCNT    ;GET RETRY COUNT.
0002 310001     BLOOP:  LXI  SP,100H    ;SET STACK POINTER.
0005 210044             LXI  H,CPMB     ;CP/M STARTS HERE.
0008 1633               MVI  D,NSECTS   ;NUMBER OF SECTORS TO READ.
000A 0E02               MVI  C,2        ;SECTOR NUMBER.
000C 0604       RNTRK:  MVI  B,4        ;FOR HEAD LOAD.
000E CD2900     RNSEC:  CALL READ       ;READ FIRST SECTOR.
0011 15                 DCR  D          ;IF DONE,
0012 CA005A             JZ   BOOTE      ;GO TO CP/M.
```

```
5CEB D303               OUT  LDATA              ;PRINT IT.
5CED C9                 RET                     ;RETURN FROM LIST.

                        IF LSTNUL       ;IF LIST NULLS
                LINUL:  PUSH B          ;SAVE B&C.
                        MVI  B,(LNULL AND 0FFH)+1 ;GET NULL COUNT
                LINUL1: CALL LTBSY      ;PRINT (CR FIRST).
                        MVI  C,0        ;GET NULL CHAR.
                        DCR  B          ;DECREMENT COUNTER.
                        JNZ  LINUL1     ;DO NEXT NULL.
                        JMP  LINUL2     ;EXIT THE ROUTINE.
                        ENDIF

                        IF   LSTPAG     ;IF LIST DEV. PAGING,
                LINUL3: PUSH B          ;SAVE B,C PAIR
                        LDA  LFCNT      ;GET LINE-FEED COUNT.
                        INR  A          ;INCREMENT IT.
                        STA  LFCNT      ;SAVE IT BACK.
                        CPI  LINCNT-(LINCNT/11)  ;END OF PAGE?
                        MVI  B,1        ;SET UP FOR 1 LF.
                        JNZ  NOTEOP     ;HOP IF NOT END.
                        XRA  A          ;SET LF COUNT = 0.
                        STA  LFCNT
                        MVI  B,(LINCNT/11)+1  ;BETWEEN PAGES.
                NOTEOP: MVI  C,0AH      ;GET LINE-FEED CODE.
                LSTPA1: CALL LTBSY      ;PRINT LINE-FEED.
                        DCR  B          ;DECREMENT LF COUNTER.
                        JNZ  LSTPA1     ;DO NEXT LINE FEED?
                        ENDIF

                        IF LSTNUL OR LSTPAG  ;IF NULLS OR PAGING,
                LINUL2: POP  B          ;RESTORE B&C.
                        MOV  A,C        ;RESTORE A.
                        RET             ;RETURN FROM LIST.
                        ENDIF
                ;
                ;NOTE:  AS THERE ARE ONLY SIX (6) SECTORS
                ;AVAILABLE FOR CBIOS ON THE SECOND SYSTEM TRACK (1),
                ;THE LAST ADDRESS BEFORE THIS POINT SHOULD BE NO
                ;GREATER THAN THE CBIOS STARTING ADDRESS + 037F (HEX).
                ;THIS WILL NORMALLY BE XD7F (HEX).
                ;
                ; BIOS SCRATCH AREA.
                ;
5CEE            TRK:    DS   1          ;CURRENT TRACK NUMBER.
5CEF            SECT:   DS   1          ;CURRENT SECTOR NUMBER.
5CF0            DMAADD: DS   2          ;DISK TRANSFER ADDRESS.
                ;
                ; THE NEXT SEVERAL BYTES, BETWEEN STARTZ AND
                ; ENDZ, ARE SET TO ZERO AT COLD BOOT TIME.
                ;
                STARTZ:                 ;START OF ZEROED AREA.
5CF2            DISKNO: DS   2          ;DISK NUMBER
                ;
                ; SPECIAL FLAGS.
5CF4            HLSF:   DS   1          ;HEAD-LOAD SELECT FLAG.
5CF5            LFCNT:  DS   1          ;PAGING LINE-FEED COUNT.
                ;
                ; TRTAB - DISK TRACK TABLE - PRESENT POSITION OF
                ;       HEADS FOR UP TO 4 DRIVES.
                ;
5CF6            TRTAB:  DS   4
                ENDZ:                   ;END OF ZEROED AREA.
5CFA            ERCNT:  DS   1          ;ERROR COUNT FOR RETRIES.
5CFB            SERCNT: DS   1          ;SEEK RETRY COUNTER.
5CFC            LATCH:  DS   1          ;NEW CODE FOR LATCH.
5CFD            CLATCH: DS   1          ;CURRENT CODE IN LATCH.
5CFE =          BEGDAT  EQU  $
5CFE            DIRBUF: DS   128        ;DIRECTORY BUFFER
5D7E            ALV0:   DS   31
5D9D            CSV0:   DS   16
5DAD            ALV1:   DS   31
5DCC            CSV1:   DS   16
5DDC            ALV2:   DS   31
5DFB            CSV2:   DS   16
5E0B            ALV3:   DS   31
5E2A            CSV3:   DS   16
5E3A =          ENDDAT  EQU  $
013C =          DATSIZ  EQU  $-BEGDAT           ;TOTAL SIZE OF DISK PARM STORAGE.
5E3A                     END
```

```
0015 0600               MVI  B,0          ;FOR NO HEAD LOAD.
0017 0C                 INR  C            ;INCREMENT TRACK COUNT.
0018 79                 MOV  A,C          ;DONE WITH
0019 FE1B               CPI  SPT+1        ;THIS TRACK?
001B DA0E00             JC   RNSEC        ;IF NOT, READ NEXT SECTOR.

                        IF   NOT PERSCI AND NOT DUBSID
001E 3E53               MVI  A,53H        ;STEP COMMAND.
0020 D3F8               OUT  DCOM         ;ISSUE IT.
0022 DBFC               IN   WAIT         ;WAIT UNTIL DONE.
                        ENDIF

                        IF   PERSCI AND NOT DUBSID
                        MVI  A,50H        ;INCREMENT TRACK
                        OUT  DCOM         ;REGISTER.
                        IN   WAIT         ;WAIT FOR INTRQ.
                        MVI  A,1          ;STEP
                        OUT  DCONT        ;PERSCI.
                        MVI  A,72H        ;SWITCH WAIT FOR
                        OUT  DCONT        ;SEEK COMPLETE.
                        IN   WAIT         ;WAIT.
                        MVI  A,0F2H       ;SWITCH WAIT
                        OUT  DCONT        ;BACK.
                        ENDIF

                        IF   DUBSID       ;IF DOUBLE SIDED SYSTEM.
                        MVI  A,0B2H       ;SIDE SELECT COMMAND.
                        OUT  DCONT        ;ISSUE IT.
                        ENDIF

0024 0E01               MVI  C,1          ;SECTOR NUMBER.
0026 C30C00             JMP  RNTRK        ;READ NEXT TRACK.

0029 79         READ:   MOV  A,C          ;GET SECTOR NUMBER.
002A D3FA               OUT  SECT         ;SET SECTOR REGISTER.
002C 3E88               MVI  A,88H        ;COMMAND FOR READ.
002E B0                 ORA  B            ;GET HEAD LOAD BIT.
002F D3F8               OUT  DCOM         ;ISSUE COMMAND.
0031 DBFC       RLOOP:  IN   WAIT         ;WAIT FOR DRQ.
0033 B7                 ORA  A            ;SET FLAGS.
0034 F23E00             JP   CHECK        ;JUMP IF DONE.
0037 DBFB               IN   DATA         ;READ DATA.
0039 77                 MOV  M,A          ;PUT IN MEMORY.
003A 23                 INX  H            ;INCREMENT POINTER.
003B C33100             JMP  RLOOP        ;LOOP UNTIL DONE.

003E DBF8       CHECK:  IN   DSTAT        ;READ STATUS.
0040 E69D               ANI  9DH          ;LOOK AT ERROR BITS.
0042 C8                 RZ                ;OK IF ZERO.
0043 1D                 DCR  E            ;DECREMENT RETRY COUNT.
0044 C20200             JNZ  BLOOP        ;TRY AGAIN IF NOT ZERO.
0047 2F                 CMA               ;INVERT AND SEND
0048 D3FF               OUT  0FFH         ;TO FRONT PANEL.
004A C34A00     HERE:   JMP  HERE         ;LOOP.

007D                    ORG  7DH          ;PUT JUMP HERE.
007D C30000             JMP  BOOT         ;JUMP INTO BOOT.
0080                    END  BOOT         ;END OF BOOT.
```

B>

ABIOS24.PRN STD DD

```
                ; CP/M BASIC INPUT/OUTPUT OPERATING SYSTEM (BIOS)
                ; TARBELL ELECTRONICS CPM 1.4 VERSION OF 10-1-80.
                ; Copyright (c) 1980 Tarbell Electronics.
                ;
                ; THIS MODULE CONTAINS ALL THE INPUT/OUTPUT ROUTINES FOR
                ; THE CP/M SYSTEM, INCLUDING THE DISK ROUTINES FOR AUTOMATIC
                ; SELECTION OF THE DISK DENSITY AND THE NECESSARY DMA ROUTINES.
                ; THIS SECTION ALSO DEFINES THE I/O PORTS AND STATUS BITS. BY
                ; SETTING THE PROPER VALUES FOR THE EQU STATEMENTS, THE I/O MAY BE
                ; AUTOMATICALLY RECONFIGURED TO FIT MOST SITUATIONS. THE TRUE AND
                ; FALSE ONES CONTROL CONDITIONAL ASSEMBLIES OF DIFFERENT SECTIONS
                ; OF I/O ROUTINES TO FIT DIFFERENT INTERFACE REQUIREMENTS.

FFFF =          TRUE    EQU  0FFFFH     ;DEFINE VALUE OF TRUE.
0000 =          FALSE   EQU  NOT TRUE   ;DEFINE VALUE OF FALSE.

                ************************************************
                *** THIS BEGINS THE AREA WHICH REQUIRES CHANGES ***
                ***      FOR DIFFERENT CONSOLE I/O SYSTEMS      ***
                ************************************************

0018 =          MSIZE   EQU  24         ;MEMORY SIZE IN KBYTES.
0000 =          INTRP   EQU  FALSE      ;TRUE IF INTERRUPTS ALLOWED.
0000 =          TESTING EQU  FALSE      ;TRUE FOR TESTING ERRORS
0000 =          TARBELL EQU  FALSE      ;TRUE IF USING THE TARBELL Z-80 CPU.
0000 =          IOBASE  EQU  0          ;BASE IO ADDR FOR TARBELL CPU (0 or 10 hex).
0000 =          TIMER   EQU  FALSE      ;TRUE IF USING CPU TIMER (Tarbell CPU board).
FFFF =          STD     EQU  TRUE       ;TRUE IF STANDARD I/O.
0000 =          VDB8024 EQU  FALSE      ;TRUE IF USING SD SALES VDB-8024.
0000 =          DELTA   EQU  FALSE      ;TRUE IF USING DELTA CPU CARD.
0000 =          MSIO2   EQU  FALSE      ;TRUE IF MITS 2SIO.
0000 =          ISIO2   EQU  FALSE      ;TRUE IF IMSAI SIO-2.
0000 =          TUART   EQU  FALSE      ;TRUE IF CROMEMCO TUART.
0000 =          VDM     EQU  FALSE      ;TRUE IF PROC TECH VDM.
0000 =          FLASH   EQU  FALSE      ;TRUE IF VG FLASHWRITER.
0000 =          VB1     EQU  FALSE      ;TRUE IF SSM VB1-B.
0000 =          OTHER   EQU  FALSE      ;TRUE IF SOMETHING ELSE.
0000 =          SOLOS   EQU  FALSE      ;TRUE IF PROC TECH SOLOS.
FFFF =          BACKSP  EQU  TRUE       ;AUTO-BACKSPACE FOR CRT'S.
0000 =          DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED DRIVES.(1 LOGICAL DRIVE)
0000 =          DMACNTL EQU  FALSE      ;TRUE IF DMA CONTROLLER
0000 =          SPOOL   EQU  FALSE      ;TRUE IF USING KLH SPOOLER
                ;
                        IF NOT SOLOS AND NOT TARDEL     ;IF NOT PROC TECH SOLOS,
0000 =          CSTAT   EQU  0          ;CONSOLE STATUS PORT.
0000 =          CCOM    EQU  0          ;CONSOLE COMMAND PORT.
0001 =          CDATA   EQU  1          ;CONSOLE DATA PORT.
0002 =          LSTAT   EQU  2          ;LIST STATUS PORT.
0002 =          LCOM    EQU  2          ;LIST COMMAND PORT.
0003 =          LDATA   EQU  3          ;LIST DATA PORT.
                        ENDIF
                ;
0000 =          CONUL   EQU  FALSE      ;CONSOLE NULLS?
0010 =          CNULL   EQU  16         ;CONSOLE NULL COUNT.
0000 =          LSTNUL  EQU  FALSE      ;LIST DEVICE NULLS?
0000 =          LNULL   EQU  0          ;LIST NULL COUNT.
0000 =          LSTPAG  EQU  FALSE      ;LIST DEVICE PAGING?
0042 =          LINCNT  EQU  66         ;LINES PER PAGE.
0000 =          HF      EQU  0          ;8 FOR HD LD AT BEG OF SEEK.
0002 =          S       EQU  2          ;RATE 0=3ms,1=6MS, 2=10MS, 3=20MS.
```

```
0000 =          DUAL    EQU  FALSE       ;TRUE IF DUAL DRIVE.(2 HEADS MOVE TOGETHER)

                ********************************************************
                *** THIS IS THE END OF THE AREA WHICH NORMALLY NEED ***
                ***      BE CHANGED FOR MOST CONSOLE I/O SYSTEMS      ***
                ********************************************************
0000 =          VIDEO   EQU  VDM OR FLASH OR VB1     ;TRUE FOR ANY VIDEO.
FFFF =          RDYLO   EQU  STD OR SOLOS OR OTHER  ;STATUS READY WHEN LOW.
0000 =          RDYHI   EQU  NOT RDYLO
0000 =          TARDEL  EQU  TARBELL OR DELTA    ;IF USING TARBELL OR DELTA CPU.
                ;
                        IF   TARDEL      ;IF USING TARBELL OR DELTA CPU
                CCOM    EQU  IOBASE+1    ;CONSOLE COMMAND PORT
                CSTAT   EQU  IOBASE+1    ;CONSOLE STATUS PORT ( CHAN A.)
                CDATA   EQU  IOBASE+0    ;CONSOLE DATA PORT
                LCOM    EQU  IOBASE+3    ;LIST COMMAND PORT
                LSTAT   EQU  IOBASE+3    ;LIST STATUS PORT (CHAN B.)
                LDATA   EQU  IOBASE+2    ;LIST DATA PORT
                        ENDIF
                ;
                        IF      TIMER AND TARBELL        ;MUST BE USING TARBELL CPU.
                ;
                ; TIMER EQUATES
                ;
                TCH0    EQU     IOBASE+4         ;TIMER CHAN 0 ADDRESS
                TCH1    EQU     IOBASE+5         ;TIMER CHAN 1 ADDRESS
                TCH2    EQU     IOBASE+6         ;TIMER CHAN 2 ADDRESS
                TCMND   EQU     IOBASE+7         ;TIMER COMMAND PORT
                IMASK   EQU     IOBASE+8         ;INTERRUPT MASKING PORT
                CNTR0   EQU     00000000B        ;counter 0
                CNTR1   EQU     01000000B        ;counter 1
                CNTR2   EQU     10000000B        ;counter 2
                RLWORD  EQU     00110000B        ;read/load lsb 1st, msb 2nd.
                RLHBYTE EQU     00100000B        ;read/load msb only.
                RLLBYTE EQU     00010000B        ;read/load lsb only.
                CNTRLT  EQU     00000000B        ;counter latching operation.
                BINARY  EQU     00000000B        ;select binary operation.
                BCD     EQU     00000001B        ;select BCD operation.
                MODE0   EQU     00000000B        ;interrupt on terminal count.
                MODE1   EQU     00000010B        ;programmable One-shot.
                MODE2   EQU     00000100B        ;rate generator.
                MODE3   EQU     00000110B        ;square wave rate generator.
                MODE4   EQU     00001000B        ;software triggered strobe.
                MODE5   EQU     00001010B        ;hardware triggered strobe.
                        ENDIF

                        IF   VDM                 ;IF PROC TECH VDM1,
                SCREEN  EQU  0CC00H              ;SCREEN PLACE IN MEMORY.
                        ENDIF

                        IF  FLASH OR VB1
                SCREEN  EQU  0F800H              ;SCREEN PLACE IS DIFFERENT.
                        ENDIF

                        IF  FLASH                ;IF VG FLASHWRITTER
                ENDSCR  EQU (SCREEN+2048) SHR 8
                        ENDIF

                        IF  VB1 OR VDM                   ;IF SSM VB1-B BOARD
                ENDSCR  EQU (SCREEN+1024) SHR 8
                        ENDIF
                ;
```

```
                        IF  VIDEO
                LINES15 EQU SCREEN + 64          ;VIDEO LINES
                SCRLCNT EQU 960                  ;LINES TO SCROLL
                SCRNTOP EQU SCREEN + SCRLCNT     ;TOP OF SCREEN
                        ENDIF
                ;
0008 =          BKSP    EQU 08H                  ;BACKSPACE EQUATE
000A =          LF      EQU 0AH                  ;LINEFEED EQUATE
000D =          CR      EQU 0DH                  ;CARRIAGE RET EQUATE
000C =          FF      EQU 0CH                  ;FORM-FEED EQUATE.
                ;
0008 =          DDEN    EQU  08H         ;DOUBLE DENSITY ENABLE CODE
00F7 =          DDDSB   EQU  0F7H        ;DOUBL. DENSITY DISABLE CODE
                ;
                        IF   SOLOS               ;IF PROC TECH SOLOS,
                CSTAT   EQU  0FAH                ;CONSOLE STATUS PORT.
                KBD     EQU  0C02EH              ;SOLOS KEYBOARD.
                CLRSCR  EQU  0C0D5H              ;CLEAR SCREEN.
                SCRN    EQU  0C054H              ;SOLOS OUTPUT.
                        ENDIF

                        IF NOT SOLOS             ;IF NOT PROC TECH SOLOS,
00E0 =          DMAP    EQU  0E0H                ;DMA BASE ADDRESS.
00F8 =          DISK    EQU  0F8H                ;DISK BASE ADDRESS.
                        ENDIF
                ;
                        IF SOLOS                 ;IF PROC TECH SOLOS,
                DMAP    EQU  060H                ;DMA BASE ADDRESS.
                DISK    EQU  078H                ;DIFFERENT DISK PORTS.
                        ENDIF
                ;
00E0 =          ADR0    EQU  DMAP+0              ;DMA ADDRESS REG PORT.
00E1 =          WCT0    EQU  DMAP+1              ;DMA WORD COUNT REG PORT.
00E8 =          CMND    EQU  DMAP+8              ;DMA COMMAND PORT.
00F8 =          DCOM    EQU  DISK                ;DISK COMMAND PORT.
00F8 =          DSTAT   EQU  DISK                ;DISK STATUS PORT.
00F9 =          TRACK   EQU  DISK+1              ;DISK TRACK PORT.
00FA =          SECTP   EQU  DISK+2              ;DISK SECTOR PORT.
00FB =          DDATA   EQU  DISK+3              ;DISK DATA PORT.
00FC =          WAIT    EQU  DISK+4              ;DISK WAIT PORT.
00FC =          DCONT   EQU  DISK+4              ;DISK CONTROL PORT.
00FD =          DMACHK  EQU  DISK+5              ;DMA CHECK PORT.
000A =          RTCNT   EQU  10                  ;RETRY COUNT.

                        IF   STD                 ;IF STANDARD I/O,
0001 =          CKBR    EQU  00000001B           ;KEYBOARD READY BIT.
0080 =          CPTR    EQU  10000000B           ;CONS OUTPUT RDY BIT.
                        ENDIF

                        IF   VDB8024
                CKBR    EQU  00000010B           ;KEYBOARD RDY BIT.
                CPTR    EQU  00000100B           ;CON RDY BIT.
                        ENDIF

                        IF   MSIO2               ;IF MITS 2SIO,
                CKBR    EQU  00000001B           ;KEYBOARD READY BIT.
                CPTR    EQU  00000010B           ;PRINT READY BIT.
                        ENDIF

                        IF   TARDEL OR ISIO2             ;IF IMSAI SIO-2,
                CKBR    EQU  00000010B           ;KEYBOARD READY BIT.
                CPTR    EQU  00000001B           ;PRINT READY BIT.
```

```
                        ENDIF

                        IF   TUART              ;IF CROMEMCO TUART,
                CKBR    EQU  01000000B          ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B          ;PRINT READY BIT.
                        ENDIF

                        IF   SOLOS              ;IF PROC TECH SOLOS,
                CKBR    EQU  00000001B          ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B          ;DUMMY EQU.
                        ENDIF

                        IF OTHER                ;IF SOMETHING ELSE,
                CKBR    EQU  00000010B          ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B          ;PRINTER READY BIT.
                        ENDIF

0080 =          LRBIT   EQU  CPTR               ;LISTER READY BIT.
0003 =          IOBYTE  EQU  3                  ;ADDRESS OF I/O BYTE.
1C00 =          CBASE   EQU  (MSIZE-17)*1024    ;BIAS FOR LARGER THAN 17K.
4500 =          CPMB    EQU  CBASE+2900H        ;START OF CPM.
4D06 =          BDOS    EQU  CBASE+3106H        ;START OF BDOS 1.4.
1500 =          CPML    EQU  1500H              ;LENGTH OF CPM SYSTEM-BIOS.
0019 =          NSECTS  EQU  25                 ;NUMBER OF SECTORS TO READ.
                ;
5572                    ORG  CPMB+1072H         ;CP/M PATCH.
5572 CDF44F             CALL CPMB+0AF4H
5575 000000             NOP!NOP!NOP
5578 79                 MOV  A,C
5579 21F859             LXI  H,CPMB+14F8H
                ;
5A00                    ORG  CPMB+1500H         ;START OF BIOS.
                ;
                ; I/O JUMP VECTOR
                ; THIS IS WHERE CPM CALLS WHENEVER IT NEEDS
                ; TO DO ANY INPUT/OUTPUT OPERATION.
                ; USER PROGRAMS MAY USE THESE ENTRY POINTS
                ; ALSO, BUT NOTE THAT THE LOCATION OF THIS
                ; VECTOR CHANGES WITH THE MEMORY SIZE.
                ;
5A00 C33B5A             JMP  BOOT               ;FROM COLD START LOADER.
5A03 C39E5A     WBOOTE: JMP  WBOOT              ;FROM WARM BOOT.
5A06 C3E35A             JMP  CONST              ;CHECK CONSOLE KB STATUS.
5A09 C3EC5A             JMP  CONIN              ;READ CONSOLE CHARACTER.
5A0C C3025B             JMP  CONOT              ;WRITE CONSOLE CHARACTER.
5A0F C3AF5D             JMP  LIST               ;WRITE LISTING CHAR.
5A12 C3BA5D             JMP  PUNCH              ;WRITE PUNCH CHAR.
5A15 C3BB5D             JMP  READER             ;READ READER CHAR.
5A18 C3C65B             JMP  HOME               ;MOVE DISK TO TRACK ZERO.
5A1B C3285B             JMP  SELDSK             ;SELECT DISK DRIVE.
5A1E C3CC5B             JMP  SETTRK             ;SEEK TO TRACK IN REG A.
5A21 C3485C             JMP  SETSEC             ;SET SECTOR NUMBER.
5A24 C34D5C             JMP  SETDMA             ;SET DISK STARTING ADR.
5A27 C3865C             JMP  READ               ;READ SELECTED SECTOR.
5A2A C3FB5C             JMP  WRITE              ;WRITE SELECTED SECTOR.
                ;
                        IF   SPOOL              ;IF USING KLH SPOOLER
                        DB   0FFH               ;FLAG FOR SPOOLER.
                        DW   LTBSY              ;LISTER STATUS LOCATION
                        DW   LTBSY              ;FOR SPOOLER - -
                        DW   LTBSY              ;I DON'T KNOW WHY IT'S
                        DW   LTBSY              ;HERE 4 TIMES EITHER.
```

```
                        ENDIF
                    ;
                    ;THE FOLLOWING TABLE DEFINES CP/M AS EITHER SINGLE OR
                    ;DOUBLE DENSITY AND IS CHANGED ON THE FLY WHEN A DISK
                    ;IS SELECTED. THE ORDER OF THIS TABLE MUST BE AS SHOWN.
                    ;
5A2D 004E1A3F03SDTAB:   DB  00H,4EH,26,63,3,7,242       ;SINGLE DENSITY TABLE

                        IF  DUBSID
                        DB  02H,4EH,26,63,4,15,242      ;SINGLE DEN. DOUB SIDED.
                        ENDIF

5A34 010C335F04         DB  01H,0CH,51,95,4,15,237      ;SINGLE SIDED, DOUB DEN.

                        IF  DUBSID
                        DB  03H,0CH,51,95,5,31,237      ;DOUB SIDED, DOUB DEN.
                        ENDIF
                    ;
                    ; BOOT
                    ; THIS SECTION IS EXECUTED WHENEVER RESET AND RUN
                    ; IS PUSHED, AFTER THE COLDSTART LOADER READS IN
                    ; THE CPM SYSTEM.
                    ;
5A3B 318000     BOOT:   LXI  SP,80H     ;SET STACK POINTER.

                        IF  INTRP       ;IF INTERRUPTS ALLOWED,
                        EI              ;ENABLE THEM HERE.
                        ENDIF

                        IF   TIMER AND TARBELL  ;IF USING TARBELL CPU
                        MVI  A,CNTR0+RLWORD+MODE2+BINARY        ;INIT 8253
                        OUT  TCMND              ;SEND IT TO COMMAND PORT
                        LXI  B,33333            ;TIME CONSTANT FOR 60 HZ
                        MOV  A,C
                        OUT  TCH0               ;LS BYTE OF COUNT
                        MOV  A,B
                        OUT  TCH0               ;MS BYTE OF COUNT
                        ENDIF

                        IF   STD        ;IF STANDARD I/O,
5A3E 00000000           NOP!NOP!NOP!NOP ;LEAVE SPACE FOR INIT.
5A42 00000000           NOP!NOP!NOP!NOP
5A46 00000000           NOP!NOP!NOP!NOP
5A4A 00000000           NOP!NOP!NOP!NOP
                        ENDIF

                        IF   MSIO2      ;IF MITS 2SIO,
                        MVI  A,3        ;INITIALIZE 2SIO.
                        OUT  CCOM
                        OUT  LCOM
                        MVI  A,11H
                        OUT  CCOM
                        OUT  LCOM
                        ENDIF

                        IF   TARDEL OR ISIO2
                        LXI  H,IOINIT           ;point to 8251 init. bytes
                        MVI  B,4                ;there are 4 of them
                    INITIO: MOV  A,M            ;get a byte
                        OUT  CCOM               ;out to command port of console
                        OUT  LCOM               ;out to command port of lister
                        INX  H                  ;bump pointer
```

```
                        DCR  B                ;decrease count
                        JNZ  INITIO            ;loop till done.
                        ENDIF

                        IF   TUART       ;IF CROMEMCO TUART,
                        MVI  A,1         ;SET A = 1.
                        OUT  54H         ;SELECT DEVICE A.
                        OUT  52H         ;RESET DEVICE B.
                        LXI  H,BAUDRS    ;GET ADR OF BAUD RATE TABLE.
                        MVI  A,11H       ;OCTUPLE THE CLOCK.
                ITl:    OUT  02H         ;& RESET CURRENT DEV.
                        MOV  A,M         ;GET BAUD RATE FROM TABLE.
                        INX  H           ;INCREMENT POINTER.
                        OUT  0           ;SET BAUD RATE.
                        CALL CONIN       ;READ KEYBOARD.
                        CALL CONIN       ;READ KEYBOARD AGAIN.
                        CPI  0DH         ;IF NOT CARRIAGE-RETURN,
                        MVI  A,1         ;SLOW THE CLOCK.
                        JNZ  ITl         ;UNTIL A CARRIAGE-RETURN.
                        ENDIF

                        IF   SOLOS       ;IF PROC TECH SOLOS,
                        CALL CLRSCR      ;CLEAR SCREEN.
                        ENDIF

5A4E AF                 XRA  A           ;CLEAR SCRATCH AREA.
                        IF   VDM         ;IF PROC TECH VDM,
                        OUT  0C8H        ;CLEAR SCROLL PORT
                        ENDIF
5A4F 32CF5D             STA  LATCH       ;LATCH = 0
5A52 320300             STA  IOBYTE      ;CLEAR I/O BYTE.
5A55 0E0C               MVI  C,ENDZ-STARTZ     ;GET LENGTH OF ZERO AREA.
5A57 21C05D             LXI  H,STARTZ    ;GET SCRATCH ADDRESS.
5A5A 77         BOOTL:  MOV  M,A         ;PUT ZERO IN MEMORY.
5A5B 23                 INX  H           ;INCREMENT POINTER.
5A5C 0D                 DCR  C           ;DECREMENT COUNTER.
5A5D C25A5A             JNZ  BOOTL       ;LOOP TILL DONE.

                        IF   VIDEO       ;IF ANY VIDEO BOARD,
                        CALL CLEAR       ;CLEAR SCREEN.
                        ENDIF

5A60 DB01               IN   CDATA       ;CLEAR CONSOLE STATUS.
5A62 216C5D             LXI  H,SMSG      ;PRINT OPENING MESSAGE.
5A65 CD2A5D             CALL PMSG
5A68 CDEC5A             CALL CONIN       ;READ # OF DISKS.
5A6B 4F                 MOV  C,A         ;ECHO THE CHAR.
5A6C CD025B             CALL CONOT
5A6F E607               ANI  7           ;LOOK AT 3 LSB'S.
5A71 32CC5D             STA  NODSKS      ;SAVE IT.
5A74 0E00               MVI  C,0         ;SELECT DRIVE 0
5A76 CD835A     GOCPM:  CALL SETUP       ;SET UP JUMPS.
5A79 3AC05D             LDA  DISKNO      ;GET DISK NUMBER TO
5A7C 4F                 MOV  C,A         ;PASS TO CCP IN C.
5A7D CD625B             CALL SELDEN      ;CHECK THE DENSITY OF DISK.
5A80 C30045             JMP  CPMB        ;JUMP TO CCP.

                        IF   TARDEL OR ISIO2
                IOINIT: DB   0AAH,040H,0CEH,037H
                        ENDIF

                        IF    UART       ;IF CROMEMCO TUART,
```

```
                BAUDRS: DB   94H,0CEH,0A2H,92H,88H,84H,82H,1
                        ENDIF
                ;
                ;DISK SET UP ROUTINE. THIS ROUTINE IS COMMON TO BOTH THE
                ;READ AND WRITE ROUTINES FOR DMA OPERATION.
                ;
                        IF DMACNTL              ;IF USING DMA CONTROL
                ;
                DMARW:  STA  ERCNT              ;SAVE ERROR COUNT.
                RWDMA:  LDA  SECT               ;GET SECTOR TO READ/WRITE
                        OUT  SECTP              ;AND SEND IT FLOPPY CHIP.
                        LHLD DMAADD             ;GET CPM DMA ADDRESS.
                DMARWE: XRA  A                  ;CLEAR ACCUM.
                        OUT  CMND               ;RESET DMA CHIP.
                        MOV  A,C                ;FORCE INTERRUPT COMMAND BYTE
                        OUT  DCOM               ;SEND IT TO CONTROLLER.
                        MOV  A,E                ;BYTE COUNT TO TRANSFER
                        DCR  A                  ;COUNT = COUNT - 1.
                        OUT  WCT0               ;SEND IT TO DMA CHIP.
                        MOV  A,D                ;GET READ/WRITE CODE.
                        OUT  WCT0               ;AND TELL DMA CHIP WHAT TO DO.
                        MOV  A,L                ;GET LOW ADDRESS BYTE
                        OUT  ADR0               ;AND SEND IT TO DMA CHIP.
                        MOV  A,H                ;GET HIGH ADDRESS BYTE
                        OUT  ADR0               ;AND SEND IT TO DMA CHIP.
                        MVI  A,41H              ;SET UP FOR REQUEST CH. 0
                        OUT  CMND               ;SEND IT TO CONTROLLER.
                        CALL HDLD               ;CHECK HEAD LOAD BIT.
                        ORA  B                  ;'OR' IN THE READ/WRITE BITS.
                        OUT  DCOM               ;TELL FLOPPY CHIP WHAT TO DO.
                ;
                ;ADJUST H,L FOR 128 BYTE INCREASE.
                ;
                        PUSH D                  ;SAVE D,E
                        MOV  A,D                ;GET DMA COMMAND BYTE
                        ANI  3FH                ;STRIP OFF COMMAND BITS 7 & 6
                        MOV  D,A                ;NOW SET FOR H,L ADJUST.
                        DAD  D                  ;ADD IT TO H,L.
                        POP  D                  ;RESTORE D,E
                ;
                ;GENERAL PURPOSE WAIT ROUTINE.
                ;
                        MVI  A,60H              ;COUNT VALUE.
                CNTLOOP:DCR  A
                        JNZ  CNTLOOP            ;LOOP TILL = ZERO.
                SLOOP:  IN   DMACHK             ;CHECK FOR OPERATION DONE.
                        RLC                     ;BY LOOKING AT BIT 7.
                        JC   SLOOP              ;LOOP TILL BIT 7 = 0.
                        IN   DSTAT              ;CHECK AND RETURN DISK STATUS.
                        RET                     ;RETURN TO CALLER.
                        ENDIF
                ;
                ; SET UP JUMPS INTO CP/M IN LOWER MEMORY.
                ;
5A83 3EC3       SETUP:  MVI  A,0C3H     ;PUT JMP TO WBOOT
5A85 320000             STA  0          ;ADR AT ZERO.
5A88 21035A             LXI  H,WBOOTE
5A8B 220100             SHLD 1
5A8E 320500             STA  5
5A91 21064D             LXI  H,BDOS     ;PUT JUMP TO BDOS
5A94 220600             SHLD 6          ;AT ADR 5,6,7.
5A97 218000             LXI  H,80H      ;SET DEFAULT DMA ADR.
```

```
5A9A 22BE5D             SHLD DMAADD
5A9D C9                 RET             ;RETURN FROM SETUP.
                ;
                ; WARM-BOOT:  READ ALL OF CPM BACK IN
                ; EXCEPT BIOS, THEN JUMP TO CCP.
                ;
5A9E 318000     WBOOT:  LXI  SP,80H     ;SET STACK POINTER.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        EI              ;ALLOW THEM HERE.
                        ENDIF

                        IF   LSTPAG     ;IF LIST DEVICE PAGING,
                        XRA  A          ;RESET LINE-FEED COUNT.
                        STA  LPCNT
                        ENDIF

5AA1 3AC05D             LDA  DISKNO     ;SAVE DISK NUMBER.
5AA4 F5                 PUSH PSW        ;SAVE ON STACK
5AA5 0E00               MVI  C,0        ;SELECT DRIVE A
5AA7 CD285B             CALL SELDSK     ;SELECT DRIVE A
5AAA CDC65B             CALL HOME       ;HOME THE DRIVE
5AAD 210000             LXI  H,0        ;CLEAR H,L
5AB0 22C75D             SHLD DRVFLG     ;CLEAR DRIVE FLAGS
5AB3 22C95D             SHLD DRVFLG+2
5AB6 0619               MVI  B,NSECTS   ;GET # SECTORS FOR CPM READ.
5AB8 0E02               MVI  C,2        ;TRACK (B)=0, SECTOR (C)=2.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        DI              ;DISABLE THEM HERE.
                        ENDIF

5ABA 210045             LXI  H,CPMB     ;GET STARTING ADDRESS.
5ABD 22BE5D     RBLK1:  SHLD DMAADD     ;SET STARTING ADDRESS.
5AC0 CD485C             CALL SETSEC     ;READ STARTING AT SECTOR IN C.
5AC3 C5                 PUSH B
5AC4 CD865C             CALL READ
5AC7 C1                 POP  B
5AC8 C2D75A             JNZ  RDERR      ;IF ERROR, PRINT MESSAGE.
5ACB 0C                 INR  C          ;INCREMENT SECTOR NUMBER.
5ACC 05                 DCR  B          ;DECREMENT SECTOR COUNT.
5ACD C2BD5A             JNZ  RBLK1      ;ALL DONE WHEN D=0.
5AD0 F1         ALDON:  POP  PSW        ;RESTORE DISK NUMBER.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        EI              ;ALLOW THEM AGAIN HERE.
                        ENDIF

5AD1 32C05D             STA  DISKNO
5AD4 C3765A             JMP  GOCPM      ;GO BACK TO CPM.
                ;
5AD7 214F5D     RDERR:  LXI  H,BTMSG    ;GET ADDRESS OF "BOOT ERROR".
5ADA CD2A5D             CALL PMSG       ;PRINT IT.
5ADD CDEC5A             CALL CONIN      ;READ A CHAR FROM CONSOLE.
5AE0 C39E5A             JMP  WBOOT      ;DO A WARM BOOT.
                ;
                ; CHECK CONSOLE INPUT STATUS.
                ;
5AE3 DB00       CONST:  IN   CSTAT      ;READ CONSOLE STATUS.
5AE5 E601               ANI  CKBR       ;LOOK AT KB READY BIT.
5AE7 3E00               MVI  A,0        ;SET A=0 FOR RETURN.
```

```
                        IF   RDYLO      ;IF STATUS READY LOW,
5AE9 C0                 RNZ             ;NOT READY WHEN NOT 0.
                        ENDIF

                        IF   RDYHI      ;IF STATUS READY HIGH,
                        RZ              ;NOT READY WHEN ZERO.
                        ENDIF

5AEA 2F                 CMA             ;IF READY A=FF.
5AEB C9                 RET             ;RETURN FROM CONST.
                ;
                ; READ A CHARACTER FROM CONSOLE.
                ;
                CONIN:
                        IF NOT SOLOS    ;IF NOT PROC TECH SOLOS,
5AEC DB00               IN   CSTAT      ;READ CONSOLE STATUS.
5AEE E601               ANI  CKBR       ;IF NOT READY,
                        ENDIF

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                        CALL KBD        ;READ SOL KEYBOARD.
                        JZ   CONIN      ;READY WHEN NOT ZERO.
                        ENDIF

                        IF RDYLO AND NOT SOLOS
5AF0 C2EC5A             JNZ  CONIN      ;LOOP UNTIL LOW.
                        ENDIF

                        IF   RDYHI      ;IF READY WHEN HIGH,
                        JZ   CONIN      ;LOOP UNTIL HIGH.
                        ENDIF

                        IF NOT SOLOS    ;IF NOT PROC TECH SOLOS,
5AF3 DB01               IN   CDATA      ;READ A CHARACTER.
                        ENDIF

5AF5 E67F               ANI  7FH        ;MAKE MOST SIG. BIT = 0.

                        IF BACKSP       ;IF BACKSPACE ACTIVATED,
5AF7 FE7F               CPI  7FH        ;IS IT A RUBOUT?
5AF9 C0                 RNZ             ;RETURN IF NOT.
5AFA 3EFF               MVI  A,0FFH     ;SET NO PRINT FLAG.
5AFC 32C15D             STA  CONOTF
5AFF 3E7F               MVI  A,7FH      ;RESTORE RUBOUT.
                        ENDIF

5B01 C9                 RET             ;RETURN FROM CONIN.
                ;
                ; WRITE A CHARACTER TO THE CONSOLE DEVICE.
                ;
                CONOT:
                        IF   BACKSP     ;IF BACKSPACE ACTIVATED,
5B02 79                 MOV  A,C        ;GET CHARACTER.
5B03 FE7F               CPI  7FH        ;IS IT A RUBOUT?
5B05 C8                 RZ              ;IF SO, DON'T PRINT IT.
5B06 3AC15D             LDA  CONOTF     ;GET NO PRINT FLAG.
5B09 B7                 ORA  A          ;SET CPU FLAGS.
5B0A CA1D5B             JZ   CONOTA     ;NOT SET, SO PRINT.
5B0D AF                 XRA  A          ;RESET THE FLAG
5B0E 32C15D             STA  CONOTF     ;TO ZERO.
5B11 0E08               MVI  C,8        ;PRINT BACKSPACE.
5B13 CD1D5B             CALL CONOTA
```

```
5B16 0E20               MVI  C,20H      ;PRINT SPACE.
5B18 CD1D5B             CALL CONOTA
5B1B 0E08               MVI  C,8        ;ANOTHER BACKSPACE.
                CONOTA:
                        ENDIF

                        IF CONUL AND NOT VIDEO  ;IF NULLS REQUIRED,
                        MVI  A,0DH      ;IF IT'S A CR,
                        CMP  C          ;THEN HOP OUT
                        JZ   CONULL     ;TO NULL ROUTINE.
                        ENDIF
                CONOT1:
                        IF NOT VIDEO AND NOT SOLOS
5B1D DB00               IN   CSTAT      ;READ CONSOLE STATUS.
5B1F E680               ANI  CPTR       ;IF NOT READY,
                        ENDIF

                        IF RDYLO AND NOT VIDEO AND NOT SOLOS
5B21 C21D5B             JNZ  CONOT1     ;LOOP UNTIL LOW.
                        ENDIF

                        IF RDYHI AND NOT VIDEO  ;IF READY WHEN HIGH,
                        JZ   CONOT1     ;LOOP UNTIL HIGH.
                        ENDIF

                        IF NOT VIDEO AND NOT SOLOS
5B24 79                 MOV  A,C        ;GET CHARACTER.
5B25 D301               OUT  CDATA      ;PRINT IT.
5B27 C9                 RET             ;RETURN.
                        ENDIF

                        IF CONUL AND NOT VIDEO
                CONULL: PUSH B          ;SAVE B&C.
                        MVI  B,CNULL+1  ;GET NULL COUNT.
                CONUL1: CALL CONOT1     ;PRINT CR.
                        MVI  C,0        ;GET NULL CHAR.
                        DCR  B          ;DECREMENT COUNTER.
                        JNZ  CONUL1     ;DO NEXT NULL.
                        POP  B          ;RESTORE B&C.
                        MOV  A,C        ;RESTORE A.
                        RET             ;RETURN.
                        ENDIF

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                        PUSH B          ;SAVE B&C.
                        MOV  B,C        ;PUT CHAR IN B REG.
                        CALL SCRN       ;OUTPUT CHAR TO SOLOS.
                        POP  B          ;RESTORE B&C.
                        MOV  A,C        ;PUT CHAR IN A.
                        RET             ;RETURN FROM CONOT.
                        ENDIF

                        IF   VIDEO      ;IF ANY VIDEO BOARD,
                ;
                ;       VIDEO DRIVER FOR VB1-B OR VDM 1 BOARD.
                ;       WRITTEN BY G.W.MULCHIN
                ;       9-16-78
                ;
                        MOV A,C         ;GET THE CHAR INTO REG A
                        PUSH H          ;SAVE REGISTERS
                        PUSH D
                        PUSH B
```

```
                        LXI H,SCRNTOP
                        MVI M,0A0H      ;STUFF CURSOR BACK
                        RET             ;ALL DONE.
                CLEAR:  LXI H,SCREEN    ;CLEAR SCREEN
                        MVI A,ENDSCR    ;THIS IS END CHECK
                CLERA:  CMP H           ;IS IT END YET?
                        JZ FINISH
                        MVI M,' '       ;PUT SPACE ON SCREEN
                        INX H           ;BUMP POINTER
                        JMP CLERA       ;GO BACK IF NOT DONE
                FINISH: LXI H,SCRNTOP
                        MVI M,0A0H      ;STUFF CURSOR BACK AGAIN
                        SHLD VDMP       ;SAVE CURSOR POSITION.
                        RET             ;ALL DONE.
                        ENDIF           ;END OF VDM DRIVER.
                ;
                ; SELECT DISK NUMBER ACCORDING TO REGISTER C.
                ; ALSO CHECK IF DISK HAS BEEN LOGGED IN BEFORE.
                ; IF NOT, LOG IT IN AND CHECK DENSITY
                ;
5B28 79         SELDSK: MOV  A,C        ;GET NEW DISK NUMBER.
5B29 E603               ANI  3          ;ONLY LOOK AT 2 LSB'S.
5B2B 21C05D             LXI  H,DISKNO   ;GET ADR OF OLD DISK NO.
5B2E BE                 CMP  M          ;NEW = OLD?
5B2F C8                 RZ              ;IF SO, RETURN.
5B30 F5                 PUSH A          ;SAVE DISK NUMBER.
5B31 3ACC5D             LDA  NODSKS     ;GET NUMBER OF DISKS.
5B34 3D                 DCR  A          ;IF MORE THAN ONE DISK,
5B35 C24D5B             JNZ  SELMOR     ;TAKE CARE OF IT.
5B38 21635D             LXI  H,MNTMSG   ;GET ADR OF MOUNT MESSAGE.
5B3B CD2A5D             CALL PMSG       ;PRINT "MOUNT ".
5B3E F1                 POP  A          ;GET DISK NUMBER.
5B3F 32C05D             STA  DISKNO     ;UPDATE OLD WITH NEW.
5B42 C641               ADI  'A'        ;ADD ASCII FOR 'A'.
5B44 4F                 MOV  C,A        ;PUT INTO C.
5B45 CD025B             CALL CONOT      ;PRINT IT.
5B48 CDEC5A             CALL CONIN      ;READ A CARRIAGE RETURN.
5B4B AF                 XRA  A          ;SET A=0 FOR NO ERRO IND.
5B4C C9                 RET             ;RETURN FROM SELDSK.
5B4D F1         SELMOR: POP  A          ;MAKE STACK RIGHT.
5B4E 7E                 MOV  A,M        ;GET OLD DISK NUMBER.
                        IF   DUAL       ;IF DUAL DRIVE,
                        ANI  0FEH       ;CLEAR OUT BIT 0.
                        ENDIF

5B4F 5F                 MOV  E,A        ;PUT OLD DISK NO. IN D&E.
5B50 1600               MVI  D,0
5B52 21C35D             LXI  H,TRTAB    ;GET ADDRESS OF TRACK TABLE.
5B55 E5                 PUSH H          ;SAVE FOR LATER
5B56 19                 DAD  D          ;ADD DISK NO. TO ADDRESS.
5B57 DBF9               IN   TRACK      ;READ 1771 TRACK REGISTER.
5B59 77                 MOV  M,A        ;PUT INTO TABLE.
5B5A 79                 MOV  A,C        ;GET NEW DISK NUMBER.

                        IF   DUAL       ;IF A DUAL DRIVE,
                        ANI  0FEH       ;CLEAR BIT 0.
                        ENDIF

5B5B 5F                 MOV  E,A        ;PUT NEW DISK NO. IN D&E.
5B5C E1                 POP  H          ;RESTORE ADDRESS OF TRACK TABLE
5B5D 19                 DAD  D          ;ADD DISK NO. TO ADDRESS.
5B5E 7E                 MOV  A,M        ;GET NEW TRACK NUMBER.
```

```
                PUSH PSW        ;CHAR. IS IN REG A
                CALL VIDPRO     ;DO VIDEO ROUTINE
                POP PSW         ;RESTORE REGISTERS
                POP B
                POP D
                POP H
                RET             ;BACK TO CALLER
;
VIDPRO:  LHLD VDMP      ;GET SCREEN POSITION POINTER
         CPI CR         ;IS IT CARRIAGE RETURN?
         JZ CARRET
         CPI LF         ;IS IT LINEFEED?
         JZ LFNOT       ;PUT IN A BLANK
         CPI BKSP       ;IS IT A RUBOUT?
         JZ BS
         CPI  FF        ;IS IT CNTL-L?
         JZ CLEAR
         MOV M,A        ;IT HAS TO BE DATA
UPDATE:  INX H          ;UPDATE POSITION
GONE:    MVI M,0A0H     ;PUT CURSOR ON SCREEN
         JMP MAXLIN     ;CHECK FOR LINE > 64
LFNOT:   MVI M,' '      ;PUT IN A SPACE
         JMP UPDATE     ;GET OUT NOW
BS:      MVI M,' '
         LHLD VDMP      ;GET CURRENT POSITION
         DCX H
         SHLD VDMP      ;SAVE CURSOR POSITION
         JMP GONE
CARRET:  MVI M,' '      ;CHAR. IS A CARRIAGE RET.
         MOV A,L        ;UPDATE NEXT POSITION
         ANI 0C0H
         ADI 40H        ;SET UP FOR NEW LINE
         MOV L,A        ;ADDRESS OF NEW LINE
         MVI A,0
         ADC H
         MOV H,A
MAXLIN:  SHLD VDMP      ;SAVE POINTER FOR NEXT CHAR.
         MVI A,7FH
         ANA L
         RNZ            ;EXIT BACK TO MAIN PROGRAM
         MVI M,' '
         LXI H,SCRNTOP
         SHLD VDMP
         LXI H,LINES15  ;15 LINES OF SCREEN DATA
         LXI D,SCREEN   ;TOP OF SCREEN. SET UP
         LXI B,SCRLCNT  ; TO SCROLL 15 LINES
SCROLL:  MOV A,M        ;START SCROLLLING UP
         STAX D         ;STUFF REG A BY WAY OF D,E
         INX H
         INX D
         DCX B
         XRA A
         CMP B          ;15 LINES YET?
         JNZ SCROLL
         CMP C
         JNZ SCROLL     ;NOT DONE YET!
         LXI H,SCRNTOP
BLANK:   MVI M,' '      ;PUT BLANK ON SCREEN
         INX H          ;BLANK ENTIRE DATA LINE
         MOV A,L
         ANI 3FH
         JNZ    NK
```

```
5B5F D3F9               OUT  TRACK      ;PUT INTO 1771 TRACK REG.
5B61 79                 MOV  A,C        ;UPDATE OLD DISK NUMBER.
5B62 32C05D    SELDEN:  STA  DISKNO
5B65 87                 ADD  A          ;PUT BITS 1&2 AT 4&5.
5B66 87                 ADD  A
5B67 87                 ADD  A
5B68 87                 ADD  A
5B69 32CF5D             STA  LATCH      ;SAVE NEW LATCH CODE.
                ;
                ;SELECT DENSITY BASED ON BYTE VALUE
                ;IN TRACK 0 SECTOR 1 ADDRESS 7EH.
                ;IF THIS BYTE IS A 0DDH, THEN THE
                ;DISK IS DOUBLE DENSITY, ELSE, IT'S
                ;SINGLE DENSITY.
                ;
5B6C C5         DENSITY:PUSH B          ;SAVE B,C
5B6D 21C75D             LXI H,DRVFLG    ;INDEX INTO DRIVE BYTE FLAG
5B70 0600               MVI B,0         ;ZERO REG B
5B72 09                 DAD B           ;ADD THE DRIVE NUMBER
5B73 7E                 MOV A,M         ;GET THE BYTE FLAG
5B74 B7                 ORA A           ;SET THE FLAGS
5B75 FA9E5B             JM LOGED        ;SKIP IF LOGGED IN BEFORE
5B78 E5                 PUSH H          ;SAVE DRIVE TABLE POINTER.
5B79 3ACF5D             LDA  LATCH
5B7C D3FC               OUT  DCONT      ;SELECT DRIVE
5B7E 2ABE5D             LHLD DMAADD     ;GET PRESENT DMA ADDRESS
5B81 E5                 PUSH H          ;SAVE ON STACK
5B82 CDC65B             CALL HOME       ;HOME DRIVE
5B85 3E01               MVI A,1         ;SET FOR SECTOR 1
5B87 32BD5D             STA  SECT       ;SAVE SECTOR TO READ
5B8A 21D15D             LXI H,DBUFF     ;POINT TO DMA BUFFER
5B8D 22BE5D             SHLD DMAADD     ;SET UP THE DMA ADDRESS
5B90 CD865C             CALL READ       ;READ TRK 0, SEC 1 INTO DBUFF
5B93 E1                 POP  H          ;RECOVER DMAADD FROM STACK
5B94 22BE5D             SHLD DMAADD     ;RESTORE IT
5B97 E1                 POP  H          ;RESTORE DRIVE TABLE POINTER.
5B98 3A4F5E             LDA DBUFF+7EH   ;GET THE DENSITY CODE BYTE
5B9B F680               ORI  80H        ;set logged in bit
5B9D 77                 MOV  M,A        ;place it in flag table.
5B9E 010700    LOGED:   LXI  B,7        ;index value through drive table.
5BA1 E612               ANI  12H        ;MASK DENSITY AND SIDE BITS OUT.
5BA3 B7                 ORA  A          ;SINGLE DENSITY?
5BA4 212D5A             LXI  H,SDTAB    ;point to start of tables
5BA7 CAAB5B             JZ   DENSIT1    ;yes, overlay param. block
                ;
                        IF   DUBSID
                        DAD  B          ;no, add offset to next table
                        CPI  2          ;single den doub sided?
                        JZ   DENSIT1    ;yes
                        DAD  B          ;no
                        CPI  10H        ;doub den single sided?
                        JZ   DENSIT1    ;yes
                        ENDIF
                ;
5BAA 09                 DAD  B          ;no, must be doub den , doub sided
5BAB EB         DENSIT1:XCHG            ;drive table pointer --> d,e
5BAC 1A                 LDAX D          ;GET LOG BYTE
5BAD 13                 INX  D          ;BUMP POINTER
5BAE 32CB5D             STA  DENS       ;SET DENSITY
5BB1 1A                 LDAX D          ;GET A BYTE FROM TABLE
5BB2 13                 INX  D          ;BUMP POINTER
5BB3 32154D             STA BDOS+15     ;CHANGE TRANS FUNCTION
```

```
5BB6 213A4D             LXI H,BDOS+52   ;POINT TO CONSTANT DATA AREA (CP/M)
5BB9 0605               MVI B,5         ;COUNT = 5
5BBB 1A         MOVE:   LDAX D          ;GET A BYTE FROM TABLE
5BBC 77                 MOV M,A         ;OVERLAY CP/M
5BBD 13                 INX D           ;BUMP
5BBE 23                 INX H           ; POINTERS
5BBF 05                 DCR B           ;DECREASE COUNT
5BC0 C2BB5B             JNZ MOVE        ; AND LOOP TILL DONE
5BC3 C1                 POP  B          ;RESTORE B,C
5BC4 AF                 XRA  A          ;SET A = 0.
5BC5 C9                 RET             ;RETURN FROM SELDSK.
                ;
                ; MOVE DISK TO TRACK ZERO.
                ;
5BC6 3E02       HOME:   MVI  A,0+STPRAT ;RESTORE
5BC8 D3F8               OUT  DCOM
5BCA 0E00               MVI  C,0        ;TRACK 0
                ;
                ; SET TRACK NUMBER TO WHATEVER IS IN REGISTER C.
                ; ALSO PERFORM MOVE TO THE CORRECT TRACK (SEEK).
                ;
5BCC 2ACF5D     SETTRK: LHLD LATCH              ;get new and old latch.
5BCF 7C                 MOV  A,H                ;get latch value.
5BD0 E6B7               ANI  0B7H               ;strip density and side bits.
5BD2 67                 MOV  H,A                ;restore it.
                ;
                        IF   DUBSID             ;if using double sided drive.
                        LDA  DENS               ;check if double sided.
                        RRC
                        RRC                     ;look at bit 1.
                        JNC  NOTSID             ;if bit 1 = 0, it's single sided.
                        MOV  A,C                ;it's doub sided, so get track number.
                        RRC                     ;divide by 2.
                        MOV  B,A                ;save it in reg b.
                        MOV  A,L                ;get old latch value.
                        JC   SIDE2              ;change side if odd track.
                        ANI  0BFH               ;clear side bit from latch.
                        JMP  SETLAT             ;go set the latch.
                ;
                SIDE2:  ORI  40H                ;turn on side select bit.
                SETLAT: STA  CLATCH             ;save it for later.
                        ANI  0BFH               ;clear side bit.
                        CALL OLDLAT             ;check for drive change.
                        MOV  A,B                ;restore doub sided trk number.
                        ANI  7FH                ;clear bit 7.
                        MOV  C,A                ;trk number now in reg c.
                        JMP  TRKSET             ;check for density of track going to.
                        ENDIF
                ;
                        IF   NOT DUBSID         ;if not using double sided drive
5BD3 C3E15B             JMP  NOTSID             ;jump around subroutine.
                        ENDIF
                ;
5BD6 BC         OLDLAT: CMP  H                  ;new = old?
5BD7 3EFF               MVI  A,0FFH             ;if not, set = ff
5BD9 C2DD5B             JNZ  SFLAG
5BDC 2F                 CMA                     ;new = old, set = 0.
5BDD 32C25D     SFLAG:  STA  HLSF               ;save head load select flag.
5BE0 C9                 RET
                ;
5BE1 7D         NOTS    MOV  A,L                ;get latch value.
5BE2 32D05D             STA  CLATCH             ;save it
```

```
5BE5 CDD65B             CALL OLDLAT             ;check for drive change.
                        ;
5BE8 3ACB5D     TRKSET: LDA  DENS       ;CHECK DRIVE DENSITY FLAG
5BEB 0F                 RRC             ;IS IT ZERO ?
5BEC D2FD5B             JNC  TRKSD      ;YES, WE ARE SINGLE DENSITY
5BEF 79                 MOV  A,C        ;RESTORE TRACK NUMBER
5BF0 FE01               CPI  1          ;IS IT TRACK 1 ?
5BF2 DAFD5B             JC   TRKSD      ;IF LESS THEN, SET SINGLE DEN.
5BF5 3AD05D             LDA  CLATCH     ;GET THE CURRENT LATCH VALUE
5BF8 F608               ORI  DDEN       ;SET BIT 4 ON (DOUB DENSITY ON)
5BFA C3025C             JMP  TRKDD
5BFD 3AD05D     TRKSD:  LDA  CLATCH     ;GET CURRENT LATCH VALUE
5C00 E6F7               ANI  DDDSB      ;TURN OFF BIT 4 (SINGLE DENSITY ON)
5C02 32D05D     TRKDD:  STA  CLATCH     ;SAVE NEW CURRENT LATCH VALUE
5C05 D3FC               OUT  DCONT      ;  AND SET THE HARDWARE LATCH
5C07 79                 MOV  A,C        ;RESTORE TRACK NUMBER
5C08 32BC5D             STA  TRK        ;UPDATE OLD WITH NEW.
                        ;
                        ; MOVE THE HEAD TO THE TRACK IN REGISTER A.
                        ;
5C0B C5         SEEK:   PUSH B          ;SAVE B&C.
5C0C 47                 MOV  B,A        ;SAVE DESTINATION TRACK.
5C0D 3E0A               MVI  A,RTCNT    ;GET RETRY COUNT.
5C0F 32CE5D     SRETRY: STA  SERCNT     ;STORE IN ERROR COUNTER.
5C12 DBF9               IN   TRACK      ;READ PRESENT TRACK NO.
5C14 4F                 MOV  C,A        ;SAVE IN C.
5C15 79                 MOV  A,C        ;DELAY.
5C16 B8                 CMP  B          ;SAME AS NEW TRACK NO.?
5C17 78                 MOV  A,B
5C18 C21D5C             JNZ  NOTHR      ;JUMP IF NOT THERE.
5C1B C1         THERE:  POP  B          ;RESTORE B&C.
5C1C C9                 RET             ;RETURN FROM SEEK.
                        NOTHR:
                        ;DELAY LOOP TO ALLOW TUNNEL
                        ;ERASE TO END DURING WRITE.
                        ;
5C1D F5                 PUSH PSW
5C1E 3ED0               MVI  A,0D0H     ;COUNT = 208
5C20 3D         BUSY1:  DCR  A          ;LOOP
5C21 C2205C             JNZ  BUSY1      ;  TILL ZERO
5C24 F1                 POP  PSW
5C25 78                 MOV  A,B        ;RESTORE A FROM B.
5C26 D3FB               OUT  DDATA      ;TRACK TO DATA REGISTER.
5C28 3E16               MVI  A,14H+STPRAT+HLAB  ;GET STEP RATE, DO
5C2A D3F8               OUT  DCOM       ;SEEK WITH VERIFY.

                        IF   NOT DMACNTL
5C2C DBFC               IN   WAIT
5C2E DBF8               IN   DSTAT      ;CHECK STATUS
                        ENDIF

                        IF   DMACNTL
                        BUSY2:  CALL SLOOP      ;USE DMA CHECK PORT
                        ENDIF

5C30 E691               ANI  91H        ;LOOK AT BITS.
5C32 CA1B5C             JZ   THERE      ;OK IF ZERO.

                        IF TESTING      ;IF TESTING FOR ERRORS
                        PUSH H          ;SAVE H&L.
                        LXI  H,SECNT    ;GET ADR OF SEEK ERR CTR.
                        INR  M          ;ONE MORE SEEK ERROR.
```

```
                        POP  H          ;RESTORE H&L.
                        ENDIF

5C35 3ACE5D             LDA  SERCNT     ;GET ERROR COUNT.
5C38 3D                 DCR  A          ;DECREMENT COUNT.
5C39 C20F5C             JNZ  SRETRY     ;RETRY SEEK.
5C3C C1                 POP  B          ;RESTORE B&C.
5C3D 215B5D             LXI  H,SKMSG    ;PRINT "SEEK ".
5C40 DBF8               IN   DSTAT      ;READ DISK STATUS.
5C42 E691               ANI  91H        ;LOOK AT ERROR BITS.
5C44 57                 MOV  D,A        ;PUT IN REG D.
5C45 C3AF5C             JMP  ERMSG      ;DO COMMON ERR MESSAGES.
                        ;
                        ; SET DISK SECTOR NUMBER.
                        ;
5C48 79         SETSEC: MOV  A,C        ;GET SECTOR NUMBER.
5C49 32BD5D             STA  SECT       ;PUT AT SECT # ADDRESS.
5C4C C9                 RET             ;RETURN FROM SETSEC.
                        ;
                        ; SET DISK DMA ADDRESS.
                        ;
5C4D 60         SETDMA: MOV  H,B        ;MOVE B&C TO H&L.
5C4E 69                 MOV  L,C
5C4F 22BE5D             SHLD DMAADD     ;PUT AT DMA ADR ADDRESS.
5C52 C9                 RET             ;RETURN FROM SETDMA.
                        ;
                        ; HDLD - GET HEAD-LOAD BIT IF REQUIRED.
                        ;
5C53 3AC25D     HDLD:   LDA  HLSF       ;GET HEAD-LOAD FLAG.
5C56 B7                 ORA  A          ;IS A = ZERO?
5C57 CA685C             JZ   HDLD1      ;HOP IF SO.
5C5A 2F                 CMA             ;SET A = 0.
5C5B 32C25D             STA  HLSF       ;SET FLAG = 0 IF NOT.
                        ;
                        ; IF CHANGING TO A NEW DRIVE, PERFORM A SEEK
                        ; TO THE SAME TRACK TO ALLOW THE HEAD TO UNLOAD.
                        ;
5C5E DBF9               IN   TRACK      ;GET PRESENT TRACK
5C60 D3FB               OUT  DDATA      ; AND TELL CONTROLLER
5C62 3E16               MVI  A,14H+STPRAT       ;GET STEP COMMAND
5C64 D3F8               OUT  DCOM       ;SEND IT TO FLOPPY CONTROLLER

                        IF   NOT DMACNTL
5C66 DBFC               IN   WAIT       ;WAIT FOR INTRQ
                        ENDIF

                        IF   DMACNTL
                        CALL SLOOP      ;USE DMA CHECK PORT
                        ENDIF

5C68 DBF8       HDLD1:  IN   DSTAT      ;READ 1771 STATUS.
5C6A E620               ANI  20H        ;LOOK AT HL BIT.
5C6C 3E04               MVI  A,4
5C6E C8                 RZ
5C6F 97                 SUB  A          ;HEAD IS LOADED
5C70 C9                 RET             ;RETURN FROM HDLD.
                        ;
                        IF   NOT DMACNTL
5C71 32CD5D     DSKSET: STA  ERCNT      ;SAVE RETRY COCUNT
5C74 3ED0               MVI  A,0D0H     ;CAUSE INTRP
5C76 D3F8               OUT  DCOM
5C78 E3                 XTHL            ;SOME
```

```
5C79 E3                 XTHL            ;DELAY
                        ENDIF

                        IF   INTRP
                        DI
                        ENDIF

                        IF   NOT DMACNTL
5C7A 2ABE5D             LHLD DMAADD     ;STARTING ADDRESS
5C7D 3ABD5D             LDA  SECT       ;GET SECTOR NUMBER
5C80 D3FA               OUT  SECTP      ;TELL CONTROLLER
5C82 CD535C             CALL HDLD       ;CHECK FOR HEAD LOADED
5C85 C9                 RET
                        ENDIF
                ;
                ; READ THE SECTOR AT SECT, FROM THE PRESENT TRACK.
                ; USE STARTING ADDRESS AT DMAADD.
                ;
5C86 3E0A       READ:   MVI  A,RTCNT    ;GET RETRY COUNT.
                RRETRY:
                        IF   DMACNTL
                        LXI  B,80D0H    ;FLOPPY READ, FORCE INTRP.
                        LXI  D,4080H    ;DMA READ, BYTE COUNT.
                        CALL DMARW      ;COMMON READ/WRITE ROUTINE.
                        ENDIF

                        IF  INTRP       ;IF INTERRUPTS ALLOWED,
                        DI              ;DISABLE THEM HERE.
                        ENDIF

                        IF   NOT DMACNTL
5C88 0680               MVI  B,80H      ;FLOPPY READ COMMAND.
5C8A CD715C             CALL DSKSET     ;SET UP DISK CONTROLLER
5C8D B0                 ORA  B          ;SET READ COMMAND
5C8E D3F8       READE:  OUT  DCOM       ;SEND COMMAND TO 1771.
5C90 DBFC       RLOOP:  IN   WAIT       ;WAIT FOR DRQ OR INTRQ.
5C92 B7                 ORA  A          ;SET FLAGS.
5C93 F29D5C             JP   RDDONE     ;DONE IF INTRQ.
5C96 DBFB               IN   DDATA      ;READ A DATA BYTE FROM DISK.
5C98 77                 MOV  M,A        ;PUT BYTE INTO MEMORY.
5C99 23                 INX  H          ;INCREMENT MEMORY POINTER.
5C9A C3905C             JMP  RLOOP      ;KEEP READING.
5C9D DBF8       RDDONE: IN   DSTAT      ;READ DISK STATUS.
                        ENDIF

                        IF  INTRP       ;IF INTERRUPTS ALLOWED,
                        EI              ;ALLOW AGAIN HERE.
                        ENDIF

5C9F E69D               ANI  9DH        ;LOOK AT ERROR BITS.
5CA1 C8                 RZ              ;RETURN IF NONE.
5CA2 CDCE5C             CALL ERCHK      ;CHECK FOR SEEK ERROR.

                        IF  TESTING
                        LXI  H,RECNT    ;GET RD ERR COUNT ADDR.
                        INR  M          ;ONE MORE ERROR.
                        MOV  A,M
                        CMA
                        OUT  0FFH
                        ENDIF

5CA5 3ACD5D             LDA  ERCNT      ;GET ERROR COUNT.
```

```
5CA8 3D              DCR  A          ;DECREMENT COUNT.
5CA9 C2885C          JNZ  RRETRY     ;TRY TO READ AGAIN.
5CAC 213E5D          LXI  H,RDMSG    ;PRINT "READ ".
5CAF CD2A5D    ERMSG: CALL PMSG      ;PRINT ORIGIN MESSAGE.
               ERMSG1:
                     IF NOT VIDEO    ;NEED MORE ROOM?
5CB2 7A              MOV  A,D        ;GET ERROR BITS.
5CB3 E610            ANI  10H        ;IF BIT 4 IS HIGH,
5CB5 21355D          LXI  H,RNMSG    ;PRINT "RECORD NOT FOUND"
5CB8 C42A5D          CNZ PMSG
5CBB 7A              MOV  A,D        ;GET ERROR BITS.
5CBC E608            ANI  8H         ;IF BIT 3 IS HIGH,
5CBE 21395D          LXI  H,CRCMSG   ;PRINT "CRC ERROR".
5CC1 C42A5D          CNZ PMSG
                     ENDIF

5CC4 21545D          LXI  H,ERRMSG   ;PRINT "ERROR."
5CC7 CD2A5D          CALL PMSG
5CCA 3E01            MVI  A,1        ;SET FOR PERM ERR MSG.
5CCC B7              ORA  A          ;SET FLAGS.
5CCD C9              RET
                ;
                ; ERCHK - CHECK FOR RECORD NOT FOUND ERROR.
                ;
5CCE 57         ERCHK: MOV  D,A      ;SAVE ERROR BITS IN D.
5CCF E610            ANI  10H        ;IF RECORD NOT FOUND,
5CD1 C8              RZ
                ;
                ;CHECK FOR SEEK TO CORRECT TRACK,
                ;AND CHANGE IF NECESSARY.
5CD2 3EC4       CHKSK: MVI  A,0C4H   ;SEND COMMAND TO 1771
5CD4 D3F8            OUT  DCOM       ;TO READ ADDRESS.

                     IF   NOT DMACNTL
5CD6 DBFC            IN   WAIT       ;WAIT FOR DRQ OR INTRQ.
                     ENDIF

                     IF   DMACNTL
                     CALL SLOOP
                     ENDIF

5CD8 DBFB            IN   DDATA      ;READ THE TRACK ADDRESS.
5CDA 47              MOV  B,A        ;SAVE IN REGISTER B.
5CDB DBFC       CHKS2: IN   WAIT     ;WAIT FOR INTRQ.
5CDD B7              ORA  A          ;SET FLAGS.
5CDE F2E65C          JP   CHKS3      ;DONE WITH READ ADR OP.
5CE1 DBFB            IN   DDATA      ;READ ANOTHER BYTE.
5CE3 C3DB5C          JMP  CHKS2      ;DO IT AGAIN.
5CE6 DBF8       CHKS3: IN   DSTAT    ;READ DISK STATUS.
5CE8 B7              ORA  A          ;SET FLAGS.
5CE9 CAF55C          JZ   CHKS4      ;READ ADR OK IF 0.
5CEC CDC65B          CALL HOME       ;OTHERWISE, HOME FIRST.
5CEF 3ABC5D     CHKS5: LDA  TRK
5CF2 C30B5C          JMP  SEEK
5CF5 78         CHKS4: MOV  A,B      ;UPDATE TRACK REGISTER.
5CF6 D3F9            OUT  TRACK
5CF8 C3EF5C          JMP  CHKS5      ;RETURN FROM ERCHK.
                ;
                ; WRITE THE SECTOR AT SECT, ON THE PRESENT TRACK.
                ; USE STARTING ADDRESS AT DMAADD.
                ;
5CFB 3E0A       WRITE: MVI  A,RTCNT  ;GET RETRY COUNT.
```

```
                WRETRY:
                        IF   DMACNTL
                        LXI  B,0A0D0H   ;FLOPPY WRITE, FORCE INTRP
                        LXI  D,08080H   ;DMA WRITE, BYTE COUNT
                        CALL DMARW      ;COMMON ROUTINE
                        ENDIF

                        IF   NOT DMACNTL
5CFD 06A0               MVI  B,0A0H     ;WRITE COMMAND
5CFF CD715C             CALL DSKSET
5D02 B0                 ORA  B
5D03 D3F8       WRITE2: OUT  DCOM
5D05 DBFC       WLOOP1: IN   WAIT       ;WAIT FOR READY.
5D07 B7                 ORA  A          ;SET FLAGS.
5D08 F2125D             JP   WDONE      ;HOP OUT WHEN DONE.
5D0B 7E                 MOV  A,M        ;GET BYTE FROM MEM.
5D0C D3FB               OUT  DDATA      ;WRITE ONTO DISK.
5D0E 23                 INX  H          ;INCREMENT MEM PTR.
5D0F C3055D             JMP  WLOOP1     ;KEEP WRITING.
5D12 DBF8       WDONE:  IN   DSTAT
                        ENDIF

                        IF  INTRP       ;IF INTERRUPTS ALLOWED,
                        EI              ;ENABLE AGAIN HERE.
                       .ENDIF

5D14 E6FD               ANI  0FDH       ;LOOK AT THESE BITS.
5D16 C8                 RZ              ;RETURN IF NO ERR.
5D17 CDCE5C             CALL ERCHK      ;CHECK/CORRECT SEEK ERR.

                        IF  TESTING     ;IF TESTING FOR ERRORS
                        LXI  H,WECNT    ;GET ADR OF WRITE ERR CTR.
                        INR  M          ;ONE MORE WRITE ERROR.
                        MOV  A,M
                        CMA
                        OUT  0FFH
                        ENDIF

5D1A 3ACD5D             LDA  ERCNT      ;GET ERROR COUNT.
5D1D 3D                 DCR  A          ;DECREMENT COUNT.
5D1E C2FD5C             JNZ  WRETRY     ;TRY TO WRITE AGAIN.
5D21 21465D             LXI  H,WTMSG    ;PRINT "WRITE ".

                        IF NOT VIDEO    ;NEED MORE ROOM?
5D24 CD2A5D             CALL PMSG
5D27 C3B25C             JMP  ERMSG1     ;DO COMMON MESSAGES.
                        ENDIF

                        IF   VIDEO      ;WE NEED A RETURN.
                        JMP  ERMSG
                        ENDIF
                ;
                ; PRINT THE MESSAGE AT H&L UNTIL A ZERO.
                ;
5D2A 7E         PMSG:    MOV  A,M       ;GET A CHARACTER.
5D2B B7                  ORA  A         ;IF IT'S ZERO,
5D2C C8                  RZ             ;RETURN.
5D2D 4F                  MOV  C,A       ;OTHERWISE,
5D2E CD025B              CALL CONOT     ;PRINT IT.
5D31 23                  INX  H         ;INCREMENT H&L,
5D32 C32A5D              JMP  PMSG      ;AND GET ANOTHER.
                ;
```

```
                ; CBIOS MESSAGES
                ;
                        IF NOT VIDEO    ;NEED MORE ROOM?
5D35 49442000  RNMSG:   DB   'ID ',0
5D39 4352432000CRCMSG:  DB   'CRC ',0
                        ENDIF

5D3E 0D0A526561RDMSG:   DB   0DH,0AH,'Read ',0
5D46 0D0A577269WTMSG:   DB   0DH,0AH,'Write ',0
5D4F 426F6F7420BTMSG:   DB   'Boot '
5D54 4552524F52ERRMSG:  DB   'ERROR.',0
5D5B 0D0A536565SKMSG:   DB   0DH,0AH,'Seek ',0
5D63 0D0A4D6F75MNTMSG:  DB   0DH,0AH,'Mount ',0
5D6C 0D0A546172SMSG:    DB   0DH,0AH,'Tarbell '
5D76 3234               DB   MSIZE/10+'0',MSIZE MOD 10 + '0'
5D78 4B2043504D         DB   'K CPM V1.4 of 10-1-80'
5D8D 0D0A               DB   0DH,0AH

                        IF  TARBELL     ;IF USING TARBELL CPU.
                        DB  'Tarbell CPU, '
                        ENDIF

                        IF  STD         ;IF STANDARD I/O,
5D8F 5374616E64         DB   'Standard '
                        ENDIF

                        IF  MSIO2       ;IF MITS 2SIO,
                        DB   '2SIO '
                        ENDIF

                        IF  ISIO2       ;IF IMSAI SIO-2,
                        DB  'SIO-2 '
                        ENDIF

                        IF  TUART       ;IF TUART,
                        DB  'Tuart '
                        ENDIF

                        IF  SOLOS       ;IF PROC TECH SOLOS,
                        DB  'Solos '
                        ENDIF

                        IF   VDM        ;IF PROC TECH VDM,
                        DB   'VDM '
                        ENDIF

                        IF   FLASH      ;IF VG FLASHWRITER,
                        DB   'Flashwriter '
                        ENDIF

                        IF   VB1        ;IF SSM VB1-B,
                        DB   'VB1 '
                        ENDIF

                        IF  DUBSID      ;IF DOUBLE-SIDED,
                        DB   'Double-Sided '
                        ENDIF

                        IF  DUAL        ;IF DUAL DRIVE,
                        DB  'Dual '
                        ENDIF
```

```
                        IF  DMACNTL     ;IF USING DMA CONTROL
                        DB  'DMA '
                        ENDIF
5D98 5645522E           DB   'VER.'

5D9C 0D0A486F77         DB   0DH,0AH,'How Many Disks? ',0
                ;
                ; WRITE A CHARACTER ON LISTING DEVICE.
                ;
                LIST:
                        IF LSTNUL       ;IF NULLS OR PAGING,
                        MVI  A,0DH      ;IF IT'S A CR,
                        CMP  C          ;THEN HOP OUT TO
                        JZ   LINUL      ;NULL ROUTINE.
                        ENDIF

                        IF LSTPAG       ;IF PAGING
                        MVI A,0AH       ;GET A LINEFEED
                        CMP C           ;DOES IT MATCH?
                        JZ LINUL3
                        ENDIF

5DAF DB02       LTBSY:  IN   LSTAT      ;READ LISTER STATUS.

                        IF   NOT TARDEL
5DB1 E680               ANI  LRBIT      ;LOOK AT READY BIT.
                        ENDIF

                        IF   TARDEL
                        ANI  81H        ;MASK
                        XRI  81H
                        ENDIF

                        IF   TARDEL OR RDYLO    ;IF READY WHEN LOW,
5DB3 C2AF5D             JNZ  LTBSY      ;LOOP TILL LOW.
                        ENDIF

                        IF   NOT TARDEL AND RDYHI       ;IF READY WHEN HIGH,
                        JZ   LTBSY      ;LOOP TILL HIGH.
                        ENDIF

5DB6 79                 MOV  A,C        ;GET DATA BYTE.
5DB7 D303               OUT  LDATA      ;PRINT IT.
5DB9 C9                 RET             ;RETURN FROM LIST.

                        IF LSTNUL OR LSTPAG  ;IF NULLS OR PAGING,
                LINUL:  PUSH B          ;SAVE B&C.
                        MVI  B,(LNULL AND 0FFH)+1 ;GET NULL COUNT
                LINUL1: CALL LTBSY      ;PRINT (CR FIRST).
                        MVI  C,0        ;GET NULL CHAR.
                        DCR  B          ;DECREMENT COUNTER.
                        JNZ  LINUL1     ;DO NEXT NULL.
                        JMP  LINUL2     ;EXIT THE ROUTINE.
                        ENDIF

                        IF   LSTPAG     ;IF LIST DEV. PAGING,
                LINUL3: PUSH B          ;SAVE B,C PAIR
                        LDA  LFCNT      ;GET LINE-FEED COUNT.
                        INR  A          ;INCREMENT IT.
                        STA  LFCNT      ;SAVE IT BACK.
                        CPI  LINCNT-(LINCNT/11)  ;END OF PAGE?
                        MVI  B,1        ;SET UP FOR 1 LF.
```

```
                        JNZ  NOTEOP     ;HOP IF NOT END.
                        XRA  A          ;SET LF COUNT = 0.
                        STA  LFCNT
                        MVI  B,(LINCNT/11)+1  ;BETWEEN PAGES.
                NOTEOP: MVI  C,0AH      ;GET LINE-FEED CODE.
                LSTPA1: CALL LTBSY      ;PRINT LINE-FEED.
                        DCR  B          ;DECREMENT LF COUNTER.
                        JNZ  LSTPA1     ;DO NEXT LINE FEED?
                        ENDIF

                        IF LSTNUL OR LSTPAG  ;IF NULLS OR PAGING,
                LINUL2: POP  B          ;RESTORE B&C.
                        MOV  A,C        ;RESTORE A.
                        RET             ;RETURN FROM LIST.
                        ENDIF
                ;
                ; PUNCH PAPER TAPE.
                ;
5DBA C9         PUNCH:  RET             ;RETURN FROM PUNCH.
                ;
                ;  NORMALLY USED TO READ PAPER TAPE.
                ;
5DBB C9         READER: RET             ;RETURN FROM READER.
                ;
                ;NOTE:  AS THERE ARE ONLY NINE SECTORS
                ;AVAILABLE FOR CBIOS ON THE SECOND SYSTEM TRACK (1),
                ;THE LAST ADDRESS BEFORE THIS POINT SHOULD BE NO
                ;GREATER THAN THE CBIOS STARTING ADDRESS + 047F (HEX).
                ;THIS WILL NORMALLY BE XE7F (HEX).
                ;
                ; BIOS SCRATCH AREA.
                ;
5DBC            TRK:    DS   1          ;CURRENT TRACK NUMBER.
5DBD            SECT:   DS   1          ;CURRENT SECTOR NUMBER.
5DBE            DMAADD: DS   2          ;DISK TRANSFER ADDRESS.
                ;
                ; THE NEXT SEVERAL BYTES, BETWEEN STARTZ AND
                ; ENDZ, ARE SET TO ZERO AT COLD BOOT TIME.
                ;
                STARTZ:                 ;START OF ZEROED AREA.
5DC0            DISKNO: DS   1          ;DISK NUMBER (TO CP/M).

                        IF  TESTING
                ;
                ; ERROR COUNTS.  THESE LOCATIONS KEEP TRACK OF THE
                ; NUMBER OF ERRRS THAT OCCUR DURING READ, WRITE,
                ; OR SEEK OPERATIONS.  THEY ARE INITIALIZED ONLY
                ; WHEN A COLD-START IS PERFORMED BY THE BOOOTSTRAP.
                ;
                RECNT:  DS   1          ;READ ERROR COUNT.
                WECNT:  DS   1          ;WRITE ERROR COUNT.
                SECNT:  DS   1          ;SEEK ERROR COUNT.
                        ENDIF
                ;
                ; SPECIAL FLAGS.
                ;
5DC1            CONOTF: DS   1          ;NO-PRINT FLAG (WHEN FF).
5DC2            HLSF:   DS   1          ;HEAD-LOAD SELECT FLAG.

                        IF  LSTPAG
                LFCNT:  DS   1          ;PAGING LINE-FEED COUNT.
                        ENDIF
```

```
                        ;
                        ; TRTAB - DISK TRACK TABLE - PRESENT POSITION OF
                        ;       HEADS FOR UP TO 4 DRIVES.
                        ;
5DC3                    TRTAB:  DS      4
5DC7                    DRVFLG: DS      4               ;DRIVE FLAG BYTES FOR 4 DRIVES
5DCB                    DENS:   DS      1               ;CURRENT DRIVE FLAG BYTE
                        ;
                        ; VDM SCRATCH AREA.
                        ;
                        ENDZ:                           ;END OF ZEROED AREA.
                                IF      VIDEO           ;IF VIDEO BOARD IN,
                        VDMP:   DS      2               ;VIDEO CURSOR POSITION.
                                ENDIF

5DCC                    NODSKS: DS      1               ;NUMBER OF DISKS.
5DCD                    ERCNT:  DS      1               ;ERROR COUNT FOR RETRIES.
5DCE                    SERCNT: DS      1               ;SEEK RETRY COUNTER.
5DCF                    LATCH:  DS      1               ;NEW CODE FOR LATCH.
5DD0                    CLATCH: DS      1               ;CURRENT CODE IN LATCH.
5DD1                    DBUFF:  DS      128             ;DENSITY SELECT BUFFER
5E51                            END
```

```
                ;
                ; TARBELL ELECTRONICS CP/M COLDSTART LOADER
                ; VERSION OF 7-31-80.
                ;-----------------------------------------------------------
                ; Modified for DMA Control - 1-5-80.
                ; Modified for reading larger bios from TRK 1  - 6-5-80.
                ; Modified for Tarbell CPU Card  -  7-3-80.
                ; G.W.Mulchin
                ; Tarbell Electronics
                ;-----------------------------------------------------------
                ; Copyright (c) 1980 Tarbell Electronics
                ;
                ** NOTE **
                ;       The equate for Double Density (DOUBDEN) must only be
                ; set true for a disk which is formatted in double density only
                ; and one which you wish to put an operating system on to.
                ; Otherwise, leave it false if you are building an operating
                ; system on to a single density formatted disk.
                ;
                ; THIS PROGRAM IS LOADED AT LOCATION ZERO BY THE BOOTSTRAP PROGRAM,
                ; AND EXECUTED. ITS PURPOSE IS TO LOAD AND EXECUTE THE CP/M DISK
                ; OPERATING SYSTEM AT THE TOP OF THE MEMORY IN USE.
                ;
0000 =          FALSE   EQU  0          ;DEFINE VALUE OF FALSE.
FFFF =          TRUE    EQU  NOT FALSE  ;DEFINE VALUE OF TRUE.
                ;
                ;********* THIS IS THE AREA TO MAKE CHANGES IN ******************
                ;********* FOR DIFFERENT SYSTEM CONFIGURATIONS ******************
                ;                                                             **
0018 =          MSIZE   EQU  24         ;MEMORY SIZE IN DECIMAL KB.         **
0000 =          TARBELL EQU  FALSE      ;TRUE IF USING TARBELL CPU.         **
0000 =          DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED SYSTEMS.     **
0000 =          DELTA   EQU  FALSE      ;TRUE IF USING DELTA CPU CARD       **
0000 =          DOUBDEN EQU  FALSE      ;TRUE IF DOUB. DEN DISK.            **
0000 =          DMACNTL EQU  FALSE      ;TRUE IF USING DMA CONTROL          **
0000 =          BASE    EQU  0          ;TARBELL I/O PORTS (00 or 10 HEX)   **
001A =          SPT     EQU  26         ;NUMBER OF SECTORS PER TRACK.       **
001A =          DDS     EQU  26         ;sectors in trk 1 , (Range = 26 to 51) **
00F8 =          DISK    EQU  0F8H       ;DISK PORT BASE ADDRESS.            **
                ;                                                             **
                ;*************************************************************
                ;
                        IF   TARBELL
                IO      EQU  BASE       ;i/o ports on tarbell cpu.
                MMENB   EQU  IO+10      ;memory management enable port.
                MEMMAG  EQU  BASE+32    ;memory management port.
                        ENDIF

00E0 =          ADR0    EQU  0E0H       ;DMA ADDRESS PORT.
00E1 =          WCT0    EQU  0E1H       ;DMA WORD COUNT PORT.
00E8 =          CMND    EQU  0E8H       ;DMA COMMAND PORT.
00F8 =          DCOM    EQU  DISK       ;COMMAND PORT.
00F8 =          DSTAT   EQU  DISK       ;STATUS PORT.
00F9 =          TRACK   EQU  DISK+1     ;TRACK PORT.
00FA =          SECT    EQU  DISK+2     ;SECTOR PORT.
00FB =          DATA    EQU  DISK+3     ;DATA PORT.
00FC =          WAIT    EQU  DISK+4     ;WAIT PORT.
00FC =          DCONT   EQU  DISK+4     ;CONTROL PORT.
00FD =          DMACHK  EQU  DISK+5     ;DMA CHECK PORT.
00FF =          PANEL   EQU  0FFH       ;front panel machines.
```

```
0019 =          SDS     EQU  25          ;always 25 sectors to read in trk 1.
1C00 =          CBASE   EQU  (MSIZE-17)*1024
4500 =          CPMB    EQU  CBASE+2900H;START OF CP/M.
5A00 =          BOOTE   EQU  CBASE+3E00H;COLD BOOT ENTRY POINT.
0033 =          NSECTS  EQU  SDS + DDS  ;SECTORS OF CP/M.
000A =          RTCNT   EQU  10         ;NUMBER OF RETRYS.
                ;
0000                    ORG  0          ;START OF LOADER.
                ;
                BOOT:
                        IF   TARBELL    ;IF USING TARBELL CPU
                        OUT  MMENB      ;ENABLE MEMORY MANAGEMENT.
                        LXI  D,1000H    ;COUNT=16, DATA BYTE = 0
                        MVI  C,MEMMAG AND 0FFH
                MLOOP:  MOV  A,E        ;GET ADDRESS VALUE
                        ORA  C          ;MAKE I/O PORT VALUE
                        STA  MOUT+1     ;MODIFY PORT ON THE FLY
                        MOV  A,E        ;GET INIT VALUE.
                        CMA             ;FLIP IT FOR RAM ON CPU
                MOUT:   OUT  BASE       ;PUT IT TO RAM ON CPU
                        INR  E          ;BUMP DATA VALUE
                        DCR  D          ;DECREASE COUNT
                        JNZ  MLOOP      ;LOOP 16 TIMES.
                        ENDIF
                ;
                        IF   DELTA      ;IF USING DELTA CPU.
                        MVI  A,1        ;GET A 1 IN REG A.
                        OUT  9          ; AND DISABLE CPU ROM SLOT.
                        ENDIF
                ;
0000 1E0A               MVI  E,RTCNT    ;GET RETRY COUNT.
0002 310001     BLOOP:  LXI  SP,100H    ;SET STACK POINTER.
0005 210045             LXI  H,CPMB     ;CP/M STARTS HERE.
0008 1633               MVI  D,NSECTS   ;NUMBER OF SECTORS TO READ.
000A 0E02               MVI  C,2        ;SECTOR NUMBER.
000C 0604       RNTRK:  MVI  B,4        ;FOR HEAD LOAD.
000E CD2900     RNSEC:  CALL READ       ;READ FIRST SECTOR.
0011 15                 DCR  D          ;IF DONE,
0012 CA005A             JZ   BOOTE      ;GO TO CP/M.
0015 0600               MVI  B,0        ;FOR NO HEAD LOAD.
0017 0C                 INR  C          ;INCREMENT SECTOR COUNT.
0018 79                 MOV  A,C        ;DONE WITH
0019 FE1B       SECCMP: CPI  SPT+1      ;THIS TRACK?
001B DA0E00             JC   RNSEC      ;IF NOT, READ NEXT SECTOR.
                ;
                        IF   DOUBDEN AND NOT DUBSID
                        MVI  A,DDS + 1  ;number of sectors to read on trk 2.
                        STA  SECCMP+1   ;modify sector compare value.
                        MVI  A,8        ;GET SET DOUBLE DENSITY CODE
                        OUT  WAIT       ;SET LATCH FOR D.DENSITY
                        ENDIF
                ;
                        IF   NOT DUBSID
001E 3E5B               MVI  A,5BH      ;STEP COMMAND.
0020 D3F8               OUT  DCOM       ;ISSUE IT.
0022 DBFC               IN   WAIT       ;WAIT UNTIL DONE.
                        ENDIF
                ;
                        IF   DUBSID     ;IF DOUBLE SIDED SYSTEM.
                        MVI  A,40H      ;SIDE SELECT COMMAND.
                        OUT  DCONT      ;ISSUE IT.
                        ENDIF
```

```
                        ;
0024 0E01                       MVI  C,1        ;SECTOR NUMBER.
0026 C30C00                     JMP  RNTRK      ;READ NEXT TRACK.
                        ;
                        READ:
                                IF   DMACNTL    ;IF USING DMA CONTROL.
                                MVI  A,41H      ;SET UP FOR CHAN 0 REQ.
                                OUT  CMND
                                MVI  A,7FH      ;COUNT FOR 128 BYTES
                                OUT  WCT0
                                MVI  A,40H      ;READ COMMAND
                                OUT  WCT0
                                MOV  A,L        ;GET LOW ADDRESS BYTE
                                OUT  ADR0
                                MOV  A,H        ;HIGH ADDRESS BYTE
                                OUT  ADR0
                                ENDIF
                        ;
0029 79                         MOV  A,C        ;SECTOR IN A.
002A D3FA                       OUT  SECT       ;SET SECTOR REGISTER.
002C 3E88                       MVI  A,88H      ;COMMAND FOR READ.
002E B0                         ORA  B          ;GET HEAD LOAD BIT.
002F D3F8                       OUT  DCOM       ;ISSUE COMMAND.
                        ;
                                IF   NOT DMACNTL;IF NOT USING DMA CONTROL.
0031 DBFC               RLOOP:  IN   WAIT       ;WAIT FOR DRQ.
0033 B7                         ORA  A          ;SET FLAGS.
0034 F23E00                     JP   CHECK      ;JUMP IF DONE.
0037 DBFB                       IN   DATA       ;READ DATA.
0039 77                         MOV  M,A        ;PUT IN MEMORY.
003A 23                         INX  H          ;INCREMENT POINTER.
003B C33100                     JMP  RLOOP      ;LOOP UNTIL DONE.
                                ENDIF
                        ;
                                IF  DMACNTL
                        SLOPP:  IN   DMACHK     ;CHECK DMA STATUS
                                RLC             ; BIT 7
                                JC   SLOPP      ;LOOP IF CARRY
                                PUSH B          ;SAVE B,C PAIR
                                LXI  B,128      ;COUNT SET FOR 128 BYTES
                                DAD  B          ;ADJUST H,L BY 128 BYTES
                                POP  B          ;RESTORE BC
                                ENDIF
                        ;
003E DBF8               CHECK:  IN   DSTAT      ;READ STATUS.
0040 E69D                       ANI  9DH        ;LOOK AT ERROR BITS.
0042 C8                         RZ              ;OK IF ZERO.
0043 1D                         DCR  E          ;DECREMENT RETRY COUNT.
0044 C20200                     JNZ  BLOOP      ;TRY AGAIN IF NOT ZERO.
0047 2F                         CMA             ;flip for front panel
0048 D3FF                       OUT  PANEL      ;show error code from floppy.
004A C34A00             HERE:   JMP  HERE       ;LOOP.
                        ;
007D                            ORG  7DH        ;PUT JUMP HERE.
007D C7                         RST  0          ;USE RESTART
                        ;
                                IF   DOUBDEN
                                DB   0DDH       ;THIS BYTE MUST BE HERE IF DOUB DEN.
                                DB   0          ;THIS BYTE UNUSED
                                ENDIF
                        ;
                                IF   NOT DOUBDEN
```

```
007E E5                 DB    0E5H
007F 00                 DB    0
                        ENDIF
                ;
0080                    END   BOOT      ;END OF BOOT.
```

Tarbell Electronics

I.O. FILES for CP/M Plus vers. 3.0

Version of 4-14-83

Here is the first release of the I.O. drivers neccessary to implement CP/M plus on The Tarbell Computer. This version has been tested to run in a banked or non-banked configuration.

The following is a description of the various files:

512COPY.COM - This utility is used to copy a 512 byte per sector, deblocked disk to another. This utility only runs under CP/M 2 or a non-banked CP/M plus.

512FMT - This utility is used to format a diskette to the 512 byte per sector deblocked format. Track 0, side 0 is formatted to 26 x 128 byte sectors. Track 0, side 1, and all the other tracks are formatted to 16 x 512 byte sectors. This utility only runs under CP/M 2 or a non-banked CP/M plus.

512SGEN - This utility is used to place the CPMLDR+3BOOT image onto the system tracks of a 512 x 16 deblocked disk. It functions essentially the same as SYSGEN or WRTSYS. This utility only runs under CP/M 2. It will NOT run under CP/M plus at all.

COPYSYS - This is Digital Reaearchs' CP/M plus version of SYSGEN. It works on single density single sided disks only. If you place your CP/M plus on a single density single sided disk it will work fine and COPYSYS will copy it. *** NOTE *** You gan generate a single sided single density disk for booting if you wish and still use the full capabilities of TAR3DVR. After the system is up the disk in drive A: may be changed to any Tarbell format as long as each disk which is placed in drive A: has a copy of CCP.COM on it.

TAR3DVR.ASM - This is the floppy disk driver module for The Tarbell double density controller. It is auto select and supports all 6 Tarbell disk formats. It has been tested on all 6 formats and works ok.

MOVE.ASM - This is the memory management module. It contains the move , xmove and bank select routines . The bank select routine uses a memory segment table to set up the memory manager on the Tarbell cpu board. Each byte in the memory segment table defines a 4K segment of memory. The most significant 4 bits of each byte become address lines 19,18,17 & 16 while the lower 4 bits become address lines 15,14,13 & 12. It takes 16 bytes to define a full 64K bytes of memory I.E. 16-4K segments. Notice that the first 16 bytes in the memory segment table are consecutive from 00h to 0fh. This is because of the fact that when the cpu board is reset

the memory manager is disabled and since the CPMLDR loads the system into bank 0, the memory segment table must be such that bank 0 acts as if there is no translation of addresses. (DO NOT ALTER THE FIRST 16 BYTES IN THE MEMORY SEGMENT TABLE) Notice also that the last 4 bytes of each 16 byte bank definition are always 0ch,0dh,0eh,0fh. This is the COMMON area of memory referred to in the CP/M plus documentation. Changing these bytes will result in instant system crash. The first 12 bytes of the bank 1 definition should remain unchanged to be compatible with future releases of Tarbell disk utilities. The remaining bytes may be changed to reflect your memory situation as long as the last 4 bytes in each bank definition remain common. As shipped the memory table is set up to handle from 128K to 256K of memory. Using CP/M plus in the banked configuration with less than 128K is possible but requires some alteration of the memory segment table to use a larger common area. If you wish to use only 128K, all that is necessary is to use GENCPM to redefine the memory available. The MOVE module also contains a few public routines for use by other modules , chief of which is the routine which returns the real memory address for disk dma operations.

BOOT.ASM - This module performs some initialization and reads in the CCP.COM file on cold boot and reloads it on warm boot. It also prints the signon message. Also contained in the boot module is the initialization and interrupt service routines for the real time clock.

3BOOT.ASM - This is the cold boot code that goes on the first sector of the disk and reads in the CPMLDR. It has equates for use with single or double sided sided (DUBSID), single or double density (DUBDEN), and deblocked or non deblocked (DBLK) disks. Note - if DBLK is true, DUBDEN is ignored. As shipped, 3BOOT is setup for double sided deblocked disks.

DRVTBL.ASM - This module contains a list of 16 addresses which point to all of the disk drive dph's in the system. A drives disk parameter header or dph contains the information or addresses of information which describes each particular disk drive to the CP/M plus. An extension of each dph (xdph) also contains the addresses of the read, write, init and login routines for each drive. The order of the drvtbl list determines the designation letter of the drive. The first word in the list addresses the dph for drive A: the second word addresses the dph for drive B: and so on. If a disk does not exist, 0000h must be inserted in the list and the list must always be 16 words long, no Les no more. The DTBL macro is used to assure that the length of the drive table is always 16 words.

CHARIO.ASM - This module contains the character out , character in and I.O. device initialization routines. These routines are table driven and will support up to 16 serial I.O. devices. Presently it is set up for the 2 serial ports on the Tarbell CPU/IO board. More devices may be added by expanding the tables or non Tarbell I.O. may be used by changing the bytes in the tables. The last byte in the table is reserved for future use. I plan to add ETX/ACK handshaking and possibly parallel I.O. in the future. The initialization strings are defined by byte 6 in the table for any particular device , a 00h in byte 6 means no init, otherwise the high order nibble defines which init string to use and the low order nibble defines the length of the string. If init is used the string is simply output one byte after another to the command port of the device. Fancier init must be implemented by the user (in the boot module maybe). This module has been fully tested and works very well on Tarbell stuff.

LDRDVR.ASM - This is a revamped copy of TAR3DVR which has the dph's and dpb's hand coded for use in the CPMLDR utility. This module has been tested and works.

LDRKRNL.ASM - This is a revamped copy of the BIOSKRNL for use by the CPMLDR .

GENBIOS.SUB - A submit file to make linking your BIOS3.SPR easier.

GENBBIOS.SUB - A submit file to make linking your BNKBIOS3.SPR easier.

GENLDR.SUB - A submit file to make linking your CPMLDR.COM easier.

*** BUILDING YOUR CP/M PLUS SYSTEM DISK ***

The first thing you must do is make a backup copy of all three disks. Use only your copies to work with and put the originals in a safe place.

Next you will need to format and sysgen a work disk in the highest capacity format you have, Preferably 512x16-deblocked-double sided.

Next place a copy of PIP.COM and SUBMIT.COM on the new disk. Then rename PIP.COM to OPIP.COM and SUBMIT.COM to OSUBMIT.COM on the new disk.The reason for this is that the CP/M plus version of these utilities will not run under CP/M 2.

Place your work disk in drive A: and reboot the system. Then use OPIP to copy all the files from your three backup disketts to your work disk. Be sure to use the [OV] options.

```
A>OPIP A:=B:*.*[OV]
```

If you do not now have a deblocked,double sided system you will need to edit the 3BOOT.ASM file before you can boot up CP/M plus direct. Also if your disk drives will not step at 3 msec. you must edit TAR3DVR.ASM and LDRDVR.ASM to change the step rate equate to the proper value.

If you are planning to bring up a non-banked system first you must edit the BIOSKRNL.ASM file and change the BANKED equate to FALSE.

The above modules as well as some of the modules provided by Digital Research must be assembled with RMAC and linked with LINK to obtain the bios for CP/M plus. Use the following command lines to assemble and link the bios. You must use ZASM to assemble 3BOOT.

```
A>OSUBMIT ASMALL        Assembles everything

A>OSUBMIT GENBIOS       For a non-banked system

                        OR

A>OSUBMIT GENBBIOS      For a banked system
```

After you have linked your bios you must use GENCPM to create a CPM3.SYS file. This is what the CPMLDR loads on cold boot. You must reserve 1 page at the bottom of all memory segments defined in GENCPM. This is to avoid blowing up the real time clock interrupt vector as well as any future interrupt capabilities. If you specify a banked memory system the following samples may be helpful.

For a 128K system:

```
CP/M 3.0 System Generation
Copyright (C) 1982, Digital Research

Default entries are shown in (parens).
Default base is Hex, precede entry with # for decimal

Use GENCPM.DAT for defaults (Y) ?

Create a new GENCPM.DAT file (N) ?

Display Load Map at Cold Boot (Y) ?

Number of console columns (#80) ?
Number of lines in console page (#24) ?
Backspace echoes erased character (N) ?
Rubout echoes erased character (N) ?

Initial default drive (A:) ?

Top page of memory (FF) ?
Bank switched memory (Y) ?
Common memory base page (C0) ?

Long error messages (Y) ?

Accept new system definition (Y) ?

*** Bank 1 and Common are not included ***
*** in the memory segment table.       ***

Number of memory segments (#2) ?

CP/M 3 Base,size,bank (88,38,00)
Enter memory segment table:
 Base,size,bank (01,87,00) ?
 Base,size,bank (01,3F,04) ?

 CP/M 3 Sys    8800H 3800H  Bank 00
 Memseg No. 00 0100H 8700H  Bank 00
 Memseg No. 01 0100H 3F00H  Bank 04

Accept new memory segment table entries (Y) ?

Setting up directory hash tables:
 Enable hashing for drive A: (Y) ?
 Enable hashing for drive B: (Y) ?

Setting up Blocking/Deblocking buffers:

The physical record size is 0200H:
     Available space in 256 byte pages:

     TPA = 00F4H, Bank 0 = 0087H, Other banks = 0039H
```

```
                Number of directory buffers for drive A: (#32) ?

      Available space in 256 byte pages:

      TPA = 00F4H, Bank 0 = 0045H, Other banks = 0039H

                Number of data buffers for drive A: (#60) ?
                Allocate buffers outside of Common (Y) ?

      Available space in 256 byte pages:

      TPA = 00F4H, Bank 0 = 0001H, Other banks = 0001H

                Number of directory buffers for drive B: (#0) ?
                Share buffer(s) with which drive (A:) ?

      Available space in 256 byte pages:

      TPA = 00F4H, Bank 0 = 0001H, Other banks = 0001H

                Number of data buffers for drive B: (#0) ?
                Share buffer(s) with which drive (A:) ?

      Available space in 256 byte pages:

      TPA = 00F4H, Bank 0 = 0001H, Other banks = 0001H

Accept new buffer definitions (Y) ?

 BNKBIOS3 SPR  FA00H  0600H
 BNKBIOS3 SPR  B000H  1000H
 RESBDOS3 SPR  F400H  0600H
 BNKBDOS3 SPR  8200H  2E00H

*** CP/M 3.0 SYSTEM GENERATION DONE ***
```

For a 256K system:

```
CP/M 3.0 System Generation
Copyright (C) 1982, Digital Research

Default entries are shown in (parens).
Default base is Hex, precede entry with # for decimal

Use GENCPM.DAT for defaults (Y) ?

Create a new GENCPM.DAT file (N) ?

Display Load Map at Cold Boot (Y) ?

Number of console columns (#80) ?
Number of lines in console page (#24) ?
Backspace echoes erased character (N) ?
Rubout echoes erased character (N) ?

Initial default drive (A:) ?

Top page of memory (FF) ?
Bank switched memory (Y) ?
Common memory base page (C0) ?

Long error messages (Y) ?

Accept new system definition (Y) ?

*** Bank 1 and Common are not included ***
*** in the memory segment table.       ***

Number of memory segments (#4) ?

CP/M 3 Base,size,bank (88,38,00)

Enter memory segment table:

 Base,size,bank (01,87,00) ?
 Base,size,bank (01,BF,02) ?
 Base,size,bank (01,BF,03) ?
 Base,size,bank (01,BF,04) ?

 CP/M 3 Sys     8800H 3800H  Bank 00
 Memseg No. 00 0100H 8700H  Bank 00
 Memseg No. 01 0100H BF00H  Bank 02
 Memseg No. 02 0100H BF00H  Bank 03
 Memseg No. 03 0100H BF00H  Bank 04

Accept new memory segment table entries (Y) ?

Setting up directory hash tables:

 Enable hashing for drive A: (Y) ?
 Enable hashing for drive B: (Y) ?
```

```
Setting up Blocking/Deblocking buffers:

The physical record size is 0200H:

     Available space in 256 byte pages:

     TPA = 00F4H, Bank 0 = 0087H, Other banks = 0237H

               Number of directory buffers for drive A: (#43) ?

     Available space in 256 byte pages:

     TPA = 00F4H, Bank 0 = 002EH, Other banks = 0237H

               Number of data buffers for drive A: (#255) ?
               Allocate buffers outside of Common (Y) ?

     Available space in 256 byte pages:

     TPA = 00F4H, Bank 0 = 0020H, Other banks = 0039H

               Number of directory buffers for drive B: (#0) ?
               Share buffer(s) with which drive (A:) ?

     Available space in 256 byte pages:

     TPA = 00F4H, Bank 0 = 0020H, Other banks = 0039H

               Number of data buffers for drive B: (#0) ?
               Share buffer(s) with which drive (A:) ?

     Available space in 256 byte pages:

     TPA = 00F4H, Bank 0 = 0020H, Other banks = 0039H


Accept new buffer definitions (Y) ?

 BNKBIOS3 SPR  FA00H  0600H
 BNKBIOS3 SPR  A400H  1C00H
 RESBDOS3 SPR  F400H  0600H
 BNKBDOS3 SPR  7600H  2E00H

*** CP/M 3.0 SYSTEM GENERATION DONE ***
```

I chose to share buffers, as this provides for dynamic use of the buffers. (I.E. if one drive has more activity it gets more buffers.) A limitation of GENCPM is that you cannot assign more than 255 data buffers to one drive.

In a non-banked system I recommend that you use only 1 directory and 1 data buffer and share them on all drives. This will give you the largest tpa possible.

Finally you must creat a CPMLDR.COM and either execute it by typing A>CPMLDR or place it on to the system track along with the 3BOOT to boot directly. Use the following to link the CPMLDR.COM.

```
A>OSUBMIT GENLDR
```

To place the CPMLDR on the system track you must use SID or DDT to bring in the CPMLDR.COM with a bias of 880h then bring in the 3BOOT.HEX with a bias of 900h. Exit from the debugger and use 512SGEN, WRTSYS or SYSGEN to write the CPMLDR onto the system tracks. 512SGEN, WRTSYS or SYSGEN must be run under CP/M 2. Once you have the loader on the system tracks it is not necessary to change it unless you change your cpu or disk controller. CPMLDR will load any configuration of CP/M plus.

Use the following SID procedure under CP/M 2:

```
A>SID                                   You type

CP/M 3 SID - Version 3.0
#RCPMLDR.COM,880                        You type
NEXT MSZE  PC  END
1E00 1E00 0100 D0FF
#R3BOOT.HEX,900                         You type
NEXT MSZE  PC  END
0980 1E00 0100 D0FF
#G0                                     You type

A>
```

Then use 512SGEN to place the CPMLDR image on the system tracks of your work disk.

*** IN CONCLUSION ***

If you did everything correctly you should have a bootable CP/M plus disk. If you wish to make copies of this disk you must use 512COPY or 512SGEN & PIP running under CP/M 2.

Disclaimer:

Tarbell Electronics makes no representations or warranties with respect to the contents hereof or to the product herein described, and specifically disclaims any implied warranties or merchantability or fitness for any particular purpose. Further, Tarbell Electronics reserves the right to revise this publication, or any other product of Tarbell without obligation of Tarbell Electronics to notify any person of such revisions or changes. Tarbell Electronics assumes no responsibility for consequential damages to any other equipment, or for time lost, or for software or data lost, which may arise from the use of its products or software, even if Tarbell Electronics has been informed of the possibility of such damage or loss.

4-14-81

```
B>
B>TYPE DBOOT24.PRN
```

New controller

```
                ;
                ; TARBELL ELECTRONICS CP/M COLDSTART LOADER
                ; VERSION OF 7-3-80.
                ;-----------------------------------------------------------------
                ; Modified for DMA Control - 1-5-80.
                ; Modified for reading larger bios from TRK 1  - 6-5-80.
                ; Modified for Tarbell CPU Card  -  7-3-80.
                ; G.W.Mulchin
                ; Tarbell Electronics
                ;-----------------------------------------------------------------
                ; Copyright (c) 1980 Tarbell Electronics
                ;
                ** NOTE **
                ;       The equate for Double Density (DOUBDEN) must only be
                ; set true for a disk which is formatted in double density only
                ; and one which you wish to put an operating system on to.
                ; Otherwise, leave it false if you are building an operating
                ; system on to a single density formatted disk.
                ;
                ; THIS PROGRAM IS LOADED AT LOCATION ZERO BY THE BOOTSTRAP PROG
                ; AND EXECUTED. ITS PURPOSE IS TO LOAD AND EXECUTE THE CP/M DIS
                ; OPERATING SYSTEM AT THE TOP OF THE MEMORY IN USE.
                ;
0000 =          FALSE   EQU  0          ;DEFINE VALUE OF FALSE.
FFFF =          TRUE    EQU  NOT FALSE  ;DEFINE VALUE OF TRUE.
                ;
                ;********* THIS IS THE AREA TO MAKE CHANGES IN ***************
                ;********* FOR DIFFERENT SYSTEM CONFIGURATIONS ***************
                ;
0018 =          MSIZE   EQU  24         ;MEMORY SIZE IN DECIMAL KB.
0000 =          TARBELL EQU  FALSE      ;TRUE IF USING TARBELL CPU.
0000 =          DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED SYSTEMS.
0000 =          DELTA   EQU  FALSE      ;TRUE IF USING DELTA CPU CARD
0000 =          DOUBDEN:EQU  FALSE      ;TRUE IF DOUB. DEN DISK.
0000 =          DMACNTL EQU  FALSE      ;TRUE IF USING DMA CONTROL
0000 =          BASE    EQU  0          ;TARBELL I/O PORTS (00 or 10 HEX)
001A =          SPT     EQU  26         ;NUMBER OF SECTORS PER TRACK.
001A =          DDS     EQU  26         ;sectors in trk 1 , (Range = 26 to 51)
00F8 =          DISK    EQU  0F8H       ;DISK PORT BASE ADDRESS.
                ;
                ;*****************************************************************
                ;
                        IF   TARBELL
                IO      EQU  BASE       ;i/o ports on tarbell cpu.
                MMENB   EQU  IO+10      ;memory management enable port.
                MEMMAG  EQU  BASE+32    ;memory management port.
                        ENDIF

00E0 =          ADRO    EQU  0E0H       ;DMA ADDRESS PORT.
00E1 =          WCTO    EQU  0E1H       ;DMA WORD COUNT PORT.
00E8 =          CMND    EQU  0E8H       ;DMA COMMAND PORT.
00F8 =          DCOM    EQU  DISK       ;COMMAND PORT.
00F8 =          DSTAT   EQU  DISK       ;STATUS PORT.
00F9 =          TRACK   EQU  DISK+1     ;TRACK PORT.
00FA =          SECT    EQU  DISK+2     ;SECTOR PORT.
00FB =          DATA    EQU  DISK+3     ;DATA PORT.
00FC =          WAIT    EQU  DISK+4     ;WAIT PORT.
00FC =          DCONT   EQU  DISK+4     ;CONTROL PORT.
00FD =          DMACHK  EQU  DISK+5     ;DMA CHECK PORT.
```

```
00FF =          PANEL   EQU  0FFH        ;front panel machines.
0019 =          SDS     EQU  25          ;always 25 sectors to read in trk 1.
1C00 =          CBASE   EQU  (MSIZE-17)*1024
4500 =          CPMB    EQU  CBASE+2900H;START OF CP/M.
5A00 =          BOOTE   EQU  CBASE+3E00H;COLD BOOT ENTRY POINT.
0033 =          NSECTS  EQU  SDS + DDS   ;SECTORS OF CP/M.
000A =          RTCNT   EQU  10          ;NUMBER OF RETRYS.
                ;
0000                    ORG  0           ;START OF LOADER.
                ;
                BOOT:
                        IF   TARBELL     ;IF USING TARBELL CPU
                        OUT  MMENB       ;ENABLE MEMORY MANAGEMENT.
                        LXI  D,1000H     ;COUNT=16, DATA BYTE = 0
                        MVI  C,MEMMAG AND 0FFH
                MLOOP:  MOV  A,E         ;GET ADDRESS VALUE
                        ORA  C           ;MAKE I/O PORT VALUE
                        STA  MOUT+1      ;MODIFY PORT ON THE FLY
                        MOV  A,E         ;GET INIT VALUE.
                        CMA              ;FLIP IT FOR RAM ON CPU
                MOUT:   OUT  BASE        ;PUT IT TO RAM ON CPU
                        INR  E           ;BUMP DATA VALUE
                        DCR  D           ;DECREASE COUNT
                        JNZ  MLOOP       ;LOOP 16 TIMES.
                        ENDIF
                ;
                        IF   DELTA       ;IF USING DELTA CPU.
                        MVI  A,1         ;GET A 1 IN REG A.
                        OUT  9           ; AND DISABLE CPU ROM SLOT.
                        ENDIF
                ;
0000 1E0A               MVI  E,RTCNT     ;GET RETRY COUNT.
0002 310001     BLOOP:  LXI  SP,100H     ;SET STACK POINTER.
0005 210045             LXI  H,CPMB      ;CP/M STARTS HERE.
0008 1633               MVI  D,NSECTS    ;NUMBER OF SECTORS TO READ.
000A 0E02               MVI  C,2         ;SECTOR NUMBER.
000C 0604       RNTRK:  MVI  B,4         ;FOR HEAD LOAD.
000E CD2900     RNSEC:  CALL READ        ;READ FIRST SECTOR.
0011 15                 DCR  D           ;IF DONE,
0012 CA005A             JZ   BOOTE       ;GO TO CP/M.
0015 0600               MVI  B,0         ;FOR NO HEAD LOAD.
0017 0C                 INR  C           ;INCREMENT SECTOR COUNT.
0018 79                 MOV  A,C         ;DONE WITH
0019 FE1B       SECCMP: CPI  SPT+1       ;THIS TRACK?
001B DA0E00             JC   RNSEC       ;IF NOT, READ NEXT SECTOR.
                ;
                        IF   DOUBDEN AND NOT DUBSID
                        MVI  A,DDS + 1   ;number of sectors to read on trk 2.
                        STA  SECCMP+1    ;modify sector compare value.
                        MVI  A,8         ;GET SET DOUBLE DENSITY CODE
                        OUT  WAIT        ;SET LATCH FOR D.DENSITY
                        ENDIF
                ;
                        IF   NOT DUBSID
001E 3E5B               MVI  A,5BH       ;STEP COMMAND.
0020 D3F8               OUT  DCOM        ;ISSUE IT.
0022 DBFC               IN   WAIT        ;WAIT UNTIL DONE.
                        ENDIF
                ;
                        IF   DUBSID      ;IF DOUBLE SIDED SYSTEM.
                        MVI  A,40H       ;SIDE SELECT COMMAND.
                        OUT  DCONT       ;ISSUE IT.
                        ENDIF
                ;
0024 0E01               MVI  C,1         ;SECTOR NUMBER.
0026 C30C00             JMP  RNTRK       ;READ NEXT TRACK.
```

```
                        ;
                        READ:
                                IF    DMACNTL     ;IF USING DMA CONTROL.
                                MVI   A,41H       ;SET UP FOR CHAN 0 REQ.
                                OUT   CMND
                                MVI   A,7FH       ;COUNT FOR 128 BYTES
                                OUT   WCTO
                                MVI   A,40H       ;READ COMMAND
                                OUT   WCTO
                                MOV   A,L         ;GET LOW ADDRESS BYTE
                                OUT   ADRO
                                MOV   A,H         ;HIGH ADDRESS BYTE
                                OUT   ADRO
                                ENDIF
                        ;
0029 79                         MOV   A,C         ;SECTOR IN A.
002A D3FA                       OUT   SECT        ;SET SECTOR REGISTER.
002C 3E88                       MVI   A,88H       ;COMMAND FOR READ.
002E B0                         ORA   B           ;GET HEAD LOAD BIT.
002F D3F8                       OUT   DCOM        ;ISSUE COMMAND.
                        ;
                                IF    NOT DMACNTL;IF NOT USING DMA CONTROL.
0031 DBFC               RLOOP:  IN    WAIT        ;WAIT FOR DRQ.
0033 B7                         ORA   A           ;SET FLAGS.
0034 F23E00                     JP    CHECK       ;JUMP IF DONE.
0037 DBFB                       IN    DATA        ;READ DATA.
0039 77                         MOV   M,A         ;PUT IN MEMORY.
003A 23                         INX   H           ;INCREMENT POINTER.
003B C33100                     JMP   RLOOP       ;LOOP UNTIL DONE.
                                ENDIF
                        ;
                                IF   DMACNTL
                        SLOPP:  IN    DMACHK      ;CHECK DMA STATUS
                                RLC               ; BIT 7
                                JC    SLOPP       ;LOOP IF CARRY
                                PUSH  B           ;SAVE B,C PAIR
                                LXI   B,128       ;COUNT SET FOR 128 BYTES
                                DAD   B           ;ADJUST H,L BY 128 BYTES
                                POP   B           ;RESTORE BC
                                ENDIF
                        ;
003E DBF8               CHECK:  IN    DSTAT       ;READ STATUS.
0040 E69D                       ANI   9DH         ;LOOK AT ERROR BITS.
0042 C8                         RZ                ;OK IF ZERO.
0043 1D                         DCR   E           ;DECREMENT RETRY COUNT.
0044 C20200                     JNZ   BLOOP       ;TRY AGAIN IF NOT ZERO.
0047 2F                         CMA               ;flip for front panel
0048 D3FF                       OUT   PANEL       ;show error code from floppy.
004A C34A00             HERE:   JMP   HERE        ;LOOP.
                        ;
007D                            ORG   7DH         ;PUT JUMP HERE.
007D C7                         RST   0           ;USE RESTART
                        ;
                                IF    DOUBDEN
                                DB    0DDH        ;THIS BYTE MUST BE HERE IF DOUB DEN.
                                DB    0           ;THIS BYTE UNUSED
                                ENDIF
                        ;
                                IF    NOT DOUBDEN
007E E5                         DB    0E5H
007F 00                         DB    0
                                ENDIF
                        ;
0080                            END    BOOT       ;END OF BOOT.
```

New Controller 4-19-81

```
A>
B>TYPE ABIOS24.PRN


                ; CP/M BASIC INPUT/OUTPUT OPERATING SYSTEM (BIOS)
                ; TARBELL ELECTRONICS CPM 1.4 VERSION OF 7-14-80.
                ; Copyright (c) 1980 Tarbell Electronics.
                ;
                ; THIS MODULE CONTAINS ALL THE INPUT/OUTPUT ROUTINES FOR
                ; THE CP/M SYSTEM, INCLUDING THE DISK ROUTINES FOR AUTOMATIC
                ; SELECTION OF THE DISK DENSITY AND THE NECESSARY DMA ROUTINES.
                ; THIS SECTION ALSO DEFINES THE I/O PORTS AND STATUS BITS. BY
                ; SETTING THE PROPER VALUES FOR THE EQU STATEMENTS, THE I/O MAY
                ; AUTOMATICALLY RECONFIGURED TO FIT MOST SITUATIONS. THE TRUE
                ; FALSE ONES CONTROL CONDITIONAL ASSEMBLIES OF DIFFERENT SECTIO
                ; OF I/O ROUTINES TO FIT DIFFERENT INTERFACE REQUIREMENTS.

FFFF =          TRUE    EQU  OFFFFH     ;DEFINE VALUE OF TRUE.
0000 =          FALSE   EQU  NOT TRUE   ;DEFINE VALUE OF FALSE.

                ********************************************************
                *** THIS BEGINS THE AREA WHICH REQUIRES CHANGES ***
                ***       FOR DIFFERENT CONSOLE I/O SYSTEMS       ***
                ********************************************************

0018 =          MSIZE   EQU  24         ;MEMORY SIZE IN KBYTES.
0000 =          INTRP   EQU  FALSE      ;TRUE IF INTERRUPTS ALLOWED.
0000 =          TESTING EQU  FALSE      ;TRUE FOR TESTING ERRORS
0000 =          TARBELL EQU  FALSE      ;TRUE IF USING THE TARBELL Z-80 CPU.
0000 =          IOBASE  EQU  0          ;BASE IO ADDR FOR TARBELL CPU (0 or 10
0000 =          TIMER   EQU  FALSE      ;TRUE IF USING CPU TIMER (Tarbell CPU b
0000 =          STD     EQU  FALSE      ;TRUE IF STANDARD I/O.
0000 =          VDB8024 EQU  FALSE      ;TRUE IF USING SD SALES VDB-8024.
0000 =          DELTA   EQU  FALSE      ;TRUE IF USING DELTA CPU CARD.
FFFF =          MSIO2   EQU  TRUE       ;TRUE IF MITS 2SIO.
0000 =          ISIO2   EQU  FALSE      ;TRUE IF IMSAI SIO-2.
0000 =          TUART   EQU  FALSE      ;TRUE IF CROMEMCO TUART.
0000 =          VDM     EQU  FALSE      ;TRUE IF PROC TECH VDM.
0000 =          FLASH   EQU  FALSE      ;TRUE IF VG FLASHWRITER.
0000 =          VB1     EQU  FALSE      ;TRUE IF SSM VB1-B.
0000 =          OTHER   EQU  FALSE      ;TRUE IF SOMETHING ELSE.
0000 =          SOLOS   EQU  FALSE      ;TRUE IF PROC TECH SOLOS.
FFFF =          BACKSP  EQU  TRUE       ;AUTO-BACKSPACE FOR CRT'S.
0000 =          DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED DRIVES.(1 LOGIC
0000 =          DMACNTL EQU  FALSE      ;TRUE IF DMA CONTROLLER
0000 =          SPOOL   EQU  FALSE      ;TRUE IF USING KLH SPOOLER
                ;
                        IF NOT SOLOS AND NOT TARDEL     ;IF NOT PROC TECH SOLOS
0000 =          CSTAT   EQU  0          ;CONSOLE STATUS PORT.
0000 =          CCOM    EQU  0          ;CONSOLE COMMAND PORT.
0001 =          CDATA   EQU  1          ;CONSOLE DATA PORT.
0002 =          LSTAT   EQU  2          ;LIST STATUS PORT.
0002 =          LCOM    EQU  2          ;LIST COMMAND PORT.
0003 =          LDATA   EQU  3          ;LIST DATA PORT.
                        ENDIF
                ;
0000 =          CONUL   EQU  FALSE      ;CONSOLE NULLS?
0010 =          CNULL   EQU  16         ;CONSOLE NULL COUNT.
0000 =          LSTNUL  EQU  FALSE      ;LIST DEVICE NULLS?
0000 =          LNULL   EQU  0          ;LIST NULL COUNT.
0000 =          LSTPAG  EQU  FALSE      ;LIST DEVICE PAGING?
```

```
0008 =          HLAB    EQU  8              ;8 FOR HD LD AT BEG OF SEEK.
0001 =          STPRAT  EQU  1              ;RATE 0=3ms,1=6MS, 2=10MS, 3=20MS.
0000 =          DUAL    EQU  FALSE          ;TRUE IF DUAL DRIVE.(2 HEADS MOVE TOGETH

                ********************************************************
                *** THIS IS THE END OF THE AREA WHICH NORMALLY NEED ***
                ***      BE CHANGED FOR MOST CONSOLE I/O SYSTEMS     ***
                ********************************************************
0000 =          VIDEO   EQU  VDM OR FLASH OR VB1    ;TRUE FOR ANY VIDEO.
0000 =          RDYLO   EQU  STD OR SOLOS OR OTHER  ;STATUS READY WHEN LOW.
FFFF =          RDYHI   EQU  NOT RDYLO
0000 =          TARDEL  EQU  TARBELL OR DELTA   ;IF USING TARBELL OR DELTA CPU.
                ;
                        IF   TARDEL     ;IF USING TARBELL OR DELTA CPU
                CCOM    EQU  IOBASE+1   ;CONSOLE COMMAND PORT
                CSTAT   EQU  IOBASE+1   ;CONSOLE STATUS PORT ( CHAN A.)
                CDATA   EQU  IOBASE+0   ;CONSOLE DATA PORT
                LCOM    EQU  IOBASE+3   ;LIST COMMAND PORT
                LSTAT   EQU  IOBASE+3   ;LIST STATUS PORT (CHAN B.)
                LDATA   EQU  IOBASE+2   ;LIST DATA PORT
                        ENDIF
                ;
                        IF      TIMER AND TARBELL       ;MUST BE USING TARBELL
                ;
                ; TIMER EQUATES
                ;
                TCH0    EQU     IOBASE+4        ;TIMER CHAN 0 ADDRESS
                TCH1    EQU     IOBASE+5        ;TIMER CHAN 1 ADDRESS
                TCH2    EQU     IOBASE+6        ;TIMER CHAN 2 ADDRESS
                TCMND   EQU     IOBASE+7        ;TIMER COMMAND PORT
                IMASK   EQU     IOBASE+8        ;INTERRUPT MASKING PORT
                CNTR0   EQU     00000000B       ;counter 0
                CNTR1   EQU     01000000B       ;counter 1
                CNTR2   EQU     10000000B       ;counter 2
                RLWORD  EQU     00110000B       ;read/load lsb 1st, msb 2nd.
                RLHBYTE EQU     00100000B       ;read/load msb only.
                RLLBYTE EQU     00010000B       ;read/load lsb only.
                CNTRLT  EQU     00000000B       ;counter latching operation.
                BINARY  EQU     00000000B       ;select binary operation.
                BCD     EQU     00000001B       ;select BCD operation.
                MODE0   EQU     00000000B       ;interrupt on terminal count.
                MODE1   EQU     00000010B       ;programmable One-shot.
                MODE2   EQU     00000100B       ;rate generator.
                MODE3   EQU     00000110B       ;square wave rate generator.
                MODE4   EQU     00001000B       ;software triggered strobe.
                MODE5   EQU     00001010B       ;hardware triggered strobe.
                        ENDIF

                        IF   VDM                ;IF PROC TECH VDM1,
                SCREEN  EQU  0CC00H             ;SCREEN PLACE IN MEMORY.
                        ENDIF

                        IF  FLASH OR VB1
                SCREEN  EQU  0F800H             ;SCREEN PLACE IS DIFFERENT.
                        ENDIF

                        IF  FLASH               ;IF VG FLASHWRITTER
                ENDSCR  EQU (SCREEN+2048) SHR 8
                        ENDIF

                        IF  VB1 OR VDM                  ;IF SSM VB1-B BOARD
                ENDSCR  EQU (SCREEN+1024) SHR 8
                        ENDIF
                ;
                        IF  VIDEO
                LINES15 EQU SCREEN + 64         ;VIDEO LINES
```

```
                SCRLCNT EQU 960                 ;LINES TO SCROLL
                SCRNTOP EQU SCREEN + SCRLCNT    ;TOP OF SCREEN
                        ENDIF
                ;
0008 =          BKSP    EQU 08H                 ;BACKSPACE EQUATE
000A =          LF      EQU 0AH                 ;LINEFEED EQUATE
000D =          CR      EQU 0DH                 ;CARRIAGE RET EQUATE
000C =          FF      EQU 0CH                 ;FORM-FEED EQUATE.
                ;
0008 =          DDEN    EQU  08H        ;DOUBLE DENSITY ENABLE CODE
00F7 =          DDDSB   EQU  0F7H       ;DOUBL. DENSITY DISABLE CODE
                ;
                        IF   SOLOS              ;IF PROC TECH SOLOS,
                CSTAT   EQU  0FAH               ;CONSOLE STATUS PORT.
                KBD     EQU  0C02EH             ;SOLOS KEYBOARD.
                CLRSCR  EQU  0C0D5H             ;CLEAR SCREEN.
                SCRN    EQU  0C054H             ;SOLOS OUTPUT.
                        ENDIF

                        IF NOT SOLOS            ;IF NOT PROC TECH SOLOS,
00E0 =          DMAP    EQU  0E0H               ;DMA BASE ADDRESS.
00F8 =          DISK    EQU  0F8H               ;DISK BASE ADDRESS.
                        ENDIF
                ;
                        IF SOLOS                ;IF PROC TECH SOLOS,
                DMAP    EQU  060H               ;DMA BASE ADDRESS.
                DISK    EQU  078H               ;DIFFERENT DISK PORTS.
                        ENDIF
                ;
00E0 =          ADR0    EQU  DMAP+0             ;DMA ADDRESS REG PORT.
00E1 =          WCT0    EQU  DMAP+1             ;DMA WORD COUNT REG PORT.
00E8 =          CMND    EQU  DMAP+8             ;DMA COMMAND PORT.
00F8 =          DCOM    EQU  DISK               ;DISK COMMAND PORT.
00F8 =          DSTAT   EQU  DISK               ;DISK STATUS PORT.
00F9 =          TRACK   EQU  DISK+1             ;DISK TRACK PORT.
00FA =          SECTP   EQU  DISK+2             ;DISK SECTOR PORT.
00FB =          DDATA   EQU  DISK+3             ;DISK DATA PORT.
00FC =          WAIT    EQU  DISK+4             ;DISK WAIT PORT.
00FC =          DCONT   EQU  DISK+4             ;DISK CONTROL PORT.
00FD =          DMACHK  EQU  DISK+5             ;DMA CHECK PORT.
000A =          RTCNT   EQU  10                 ;RETRY COUNT.

                        IF   STD                ;IF STANDARD I/O,
                CKBR    EQU  00000001B          ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B          ;CONS OUTPUT RDY BIT.
                        ENDIF

                        IF   VDB8024
                CKBR    EQU  00000010B          ;KEYBOARD RDY BIT.
                CPTR    EQU  00000100B          ;CON RDY BIT.
                        ENDIF

                        IF   MSIO2              ;IF MITS 2SIO,
0001 =          CKBR    EQU  00000001B          ;KEYBOARD READY BIT.
0002 =          CPTR    EQU  00000010B          ;PRINT READY BIT.
                        ENDIF

                        IF   TARDEL OR ISIO2            ;IF IMSAI SIO-2,
                CKBR    EQU  00000010B          ;KEYBOARD READY BIT.
                CPTR    EQU  00000001B          ;PRINT READY BIT.
                        ENDIF

                        IF   TUART              ;IF CROMEMCO TUART,
                CKBR    EQU  01000000B          ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B          ;PRINT READY BIT.
                        ENDIF
```

```
                        ENDIF

                        IF   SOLOS              ;IF PROC TECH SOLOS,
                CKBR    EQU  00000001B          ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B          ;DUMMY EQU.
                        ENDIF

                        IF OTHER                ;IF SOMETHING ELSE,
                CKBR    EQU  00000010B          ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B          ;PRINTER READY BIT.
                        ENDIF

0002 =          LRBIT   EQU  CPTR               ;LISTER READY BIT.
0003 =          IOBYTE  EQU  3                  ;ADDRESS OF I/O BYTE.
1C00 =          CBASE   EQU  (MSIZE-17)*1024    ;BIAS FOR LARGER THAN 17K.
4500 =          CPMB    EQU  CBASE+2900H        ;START OF CPM.
4D06 =          BDOS    EQU  CBASE+3106H        ;START OF BDOS 1.4.
1500 =          CPML    EQU  1500H              ;LENGTH OF CPM SYSTEM-BIOS.
0019 =          NSECTS  EQU  25                 ;NUMBER OF SECTORS TO READ.
                ;
5572                    ORG  CPMB+1072H         ;CP/M PATCH.
5572 CDF44F             CALL CPMB+0AF4H
5575 000000             NOP!NOP!NOP
5578 79                 MOV  A,C
5579 21F859             LXI  H,CPMB+14F8H
                ;
5A00                    ORG  CPMB+1500H         ;START OF BIOS.
                ;
                ; I/O JUMP VECTOR
                ; THIS IS WHERE CPM CALLS WHENEVER IT NEEDS
                ; TO DO ANY INPUT/OUTPUT OPERATION.
                ; USER PROGRAMS MAY USE THESE ENTRY POINTS
                ; ALSO, BUT NOTE THAT THE LOCATION OF THIS
                ; VECTOR CHANGES WITH THE MEMORY SIZE.
                ;
5A00 C33B5A             JMP  BOOT               ;FROM COLD START LOADER.
5A03 C3975A     WBOOTE: JMP  WBOOT              ;FROM WARM BOOT.
5A06 C3DC5A             JMP  CONST              ;CHECK CONSOLE KB STATUS.
5A09 C3E55A             JMP  CONIN              ;READ CONSOLE CHARACTER.
5A0C C3FB5A             JMP  CONOT              ;WRITE CONSOLE CHARACTER.
5A0F C3A45D             JMP  LIST               ;WRITE LISTING CHAR.
5A12 C3AF5D             JMP  PUNCH              ;WRITE PUNCH CHAR.
5A15 C3B05D             JMP  READER             ;READ READER CHAR.
5A18 C3BF5B             JMP  HOME               ;MOVE DISK TO TRACK ZERO.
5A1B C3215B             JMP  SELDSK             ;SELECT DISK DRIVE.
5A1E C3C55B             JMP  SETTRK             ;SEEK TO TRACK IN REG A.
5A21 C3415C             JMP  SETSEC             ;SET SECTOR NUMBER.
5A24 C3465C             JMP  SETDMA             ;SET DISK STARTING ADR.
5A27 C37F5C             JMP  READ               ;READ SELECTED SECTOR.
5A2A C3F45C             JMP  WRITE              ;WRITE SELECTED SECTOR.
                ;
                        IF   SPOOL              ;IF USING KLH SPOOLER
                        DB   0FFH               ;FLAG FOR SPOOLER.
                        DW   LTBSY              ;LISTER STATUS LOCATION
                        DW   LTBSY              ;FOR SPOOLER - -
                        DW   LTBSY              ;I DON'T KNOW WHY IT'S
                        DW   LTBSY              ;HERE 4 TIMES EITHER.
                        ENDIF
                ;
                ;THE FOLLOWING TABLE DEFINES CP/M AS EITHER SINGLE OR
                ;DOUBLE DENSITY AND IS CHANGED ON THE FLY WHEN A DISK
                ;IS SELECTED. THE ORDER OF THIS TABLE MUST BE AS SHOWN.
                ;
5A2D 004E1A3F03SDTAB:   DB   00H,4EH,26,63,3,7,242      ;SINGLE DENSITY TABLE

                        IF   DUBSID
                        DB   02H,4EH,26,63,4,15,242     ;SINGLE DEN, DOUB SIDE
```

```
                        ENDIF

5A34 010C335F04         DB  01H,0CH,51,95,4,15,237      ;SINGLE SIDED, DOUB DEN.

                        IF  DUBSID
                        DB  03H,0CH,51,95,5,31,237      ;DOUB SIDED, DOUB DEN.
                        ENDIF
                ;
                ; BOOT
                ; THIS SECTION IS EXECUTED WHENEVER RESET AND RUN
                ; IS PUSHED, AFTER THE COLDSTART LOADER READS IN
                ; THE CPM SYSTEM.
                ;
5A3B 318000     BOOT:   LXI  SP,80H     ;SET STACK POINTER.

                        IF  INTRP       ;IF INTERRUPTS ALLOWED,
                        EI              ;ENABLE THEM HERE.
                        ENDIF

                        IF   TIMER AND TARBELL  ;IF USING TARBELL CPU
                        MVI  A,CNTR0+RLWORD+MODE2+BINARY        ;INIT 8253
                        OUT  TCMND              ;SEND IT TO COMMAND PORT
                        LXI  B,33333            ;TIME CONSTANT FOR 60 HZ
                        MOV  A,C
                        OUT  TCH0               ;LS BYTE OF COUNT
                        MOV  A,B
                        OUT  TCH0               ;MS BYTE OF COUNT
                        ENDIF

                        IF   STD        ;IF STANDARD I/O,
                        NOP!NOP!NOP!NOP ;LEAVE SPACE FOR INIT.
                        NOP!NOP!NOP!NOP
                        NOP!NOP!NOP!NOP
                        NOP!NOP!NOP!NOP
                        ENDIF

                        IF   MSIO2      ;IF MITS 2SIO,
5A3E 3E03               MVI  A,3        ;INITIALIZE 2SIO.
5A40 D300               OUT  CCOM
5A42 D302               OUT  LCOM
5A44 3E11               MVI  A,11H
5A46 D300               OUT  CCOM
5A48 D302               OUT  LCOM
                        ENDIF

                        IF   TARDEL OR ISIO2
                        LXI  H,IOINIT           ;point to 8251 init. bytes
                        MVI  B,4                ;there are 4 of them
                INITIO: MOV  A,M                ;get a byte
                        OUT  CCOM               ;out to command port of console
                        OUT  LCOM               ;out to command port of liste
                        INX  H                  ;bump pointer
                        DCR  B                  ;decrease count
                        JNZ  INITIO             ;loop till done.
                        ENDIF

                        IF   TUART      ;IF CROMEMCO TUART,
                        MVI  A,1        ;SET A = 1.
                        OUT  54H        ;SELECT DEVICE A.
                        OUT  52H        ;RESET DEVICE B.
                        LXI  H,BAUDRS   ;GET ADR OF BAUD RATE TABLE.
                        MVI  A,11H      ;OCTUPLE THE CLOCK.
                IT1:    OUT  02H        ;& RESET CURRENT DEV.
                        MOV  A,M        ;GET BAUD RATE FROM TABLE.
                        INX  H          ;INCREMENT POINTER.
                        OUT  0          ;SET BAUD RATE
```

```
                        CALL CONIN      ;READ KEYBOARD.
                        CALL CONIN      ;READ KEYBOARD AGAIN.
                        CPI  0DH        ;IF NOT CARRIAGE-RETURN,
                        MVI  A,1        ;SLOW THE CLOCK.
                        JNZ  IT1        ;UNTIL A CARRIAGE-RETURN.
                        ENDIF

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                        CALL CLRSCR     ;CLEAR SCREEN.
                        ENDIF

5A4A AF                 XRA  A          ;CLEAR SCRATCH AREA.
                        IF   VDM        ;IF PROC TECH VDM,
                        OUT  0C8H       ;CLEAR SCROLL PORT
                        ENDIF
5A4B 32C45D             STA  LATCH      ;LATCH = 0
5A4E 320300             STA  IOBYTE     ;CLEAR I/O BYTE.
5A51 0E0C               MVI  C,ENDZ-STARTZ    ;GET LENGTH OF ZERO AREA.
5A53 21B55D             LXI  H,STARTZ   ;GET SCRATCH ADDRESS.
5A56 77         BOOTL:  MOV  M,A        ;PUT ZERO IN MEMORY.
5A57 23                 INX  H          ;INCREMENT POINTER.
5A58 0D                 DCR  C          ;DECREMENT COUNTER.
5A59 C2565A             JNZ  BOOTL      ;LOOP TILL DONE.

                        IF   VIDEO      ;IF ANY VIDEO BOARD,
                        CALL CLEAR      ;CLEAR SCREEN.
                        ENDIF

5A5C DB01               IN   CDATA      ;CLEAR CONSOLE STATUS.
5A5E 21655D             LXI  H,SMSG     ;PRINT OPENING MESSAGE.
5A61 CD235D             CALL PMSG
5A64 CDE55A             CALL CONIN      ;READ # OF DISKS.
5A67 4F                 MOV  C,A        ;ECHO THE CHAR.
5A68 CDFB5A             CALL CONOT
5A6B E607               ANI  7          ;LOOK AT 3 LSB'S.
5A6D 32C15D             STA  NODSKS     ;SAVE IT.
5A70 0E00               MVI  C,0        ;SELECT DRIVE 0
5A72 CD7C5A     GOCPM:  CALL SETUP      ;SET UP JUMPS.
5A75 3AB55D             LDA  DISKNO     ;GET DISK NUMBER TO
5A78 4F                 MOV  C,A        ;PASS TO CCP IN C.
5A79 C30045             JMP  CPMB       ;JUMP TO CCP.

                        IF   TARDEL OR ISIO2
                IOINIT: DB   0AAH,040H,0CEH,037H
                        ENDIF

                        IF   TUART      ;IF CROMEMCO TUART,
                BAUDRS: DB   94H,0CEH,0A2H,92H,88H,84H,82H,1
                        ENDIF
                ;
                ;DISK SET UP ROUTINE. THIS ROUTINE IS COMMON TO BOTH THE
                ;READ AND WRITE ROUTINES FOR DMA OPERATION.
                ;
                        IF DMACNTL              ;IF USING DMA CONTROL
                ;
                DMARW:  STA  ERCNT              ;SAVE ERROR COUNT.
                RWDMA:  LDA  SECT               ;GET SECTOR TO READ/WRITE
                        OUT  SECTP              ;AND SEND IT FLOPPY CHIP.
                        LHLD DMAADD             ;GET CPM DMA ADDRESS.
                DMARWE: XRA  A                  ;CLEAR ACCUM.
                        OUT  CMND               ;RESET DMA CHIP.
                        MOV  A,C                ;FORCE INTERRUPT COMMAND BYTE
                        OUT  DCOM               ;SEND IT TO CONTROLLER.
                        MOV  A,E                ;BYTE COUNT TO TRANSFER
                        DCR  A                  ;COUNT = COUNT - 1.
                        OUT  WCTC               ;SEND IT TO DMA CHIP.
```

```
                        MOV   A,D                 ;GET READ/WRITE CODE.
                        OUT   WCTO                ;AND TELL DMA CHIP WHAT TO DO.
                        MOV   A,L                 ;GET LOW ADDRESS BYTE
                        OUT   ADRO                ;AND SEND IT TO DMA CHIP.
                        MOV   A,H                 ;GET HIGH ADDRESS BYTE
                        OUT   ADRO                ;AND SEND IT TO DMA CHIP.
                        MVI   A,41H               ;SET UP FOR REQUEST CH. 0
                        OUT   CMND                ;SEND IT TO CONTROLLER.
                        CALL  HDLD                ;CHECK HEAD LOAD BIT.
                        ORA   B                   ;'OR' IN THE READ/WRITE BITS.
                        OUT   DCOM                ;TELL FLOPPY CHIP WHAT TO DO.
                    ;
                    ;ADJUST H,L FOR 128 BYTE INCREASE.
                    ;
                        PUSH  D                   ;SAVE D,E
                        MOV   A,D                 ;GET DMA COMMAND BYTE
                        ANI   3FH                 ;STRIP OFF COMMAND BITS 7 & 6
                        MOV   D,A                 ;NOW SET FOR H,L ADJUST.
                        DAD   D                   ;ADD IT TO H,L.
                        POP   D                   ;RESTORE D,E
                    ;
                    ;GENERAL PURPOSE WAIT ROUTINE.
                    ;
                        MVI   A,60H               ;COUNT VALUE.
                    CNTLOOP:DCR   A
                        JNZ   CNTLOOP             ;LOOP TILL = ZERO.
                    SLOOP:  IN    DMACHK          ;CHECK FOR OPERATION DONE.
                        RLC                       ;BY LOOKING AT BIT 7.
                        JC    SLOOP               ;LOOP TILL BIT 7 = 0.
                        IN    DSTAT               ;CHECK AND RETURN DISK STATUS.
                        RET                       ;RETURN TO CALLER.
                        ENDIF
                    ;
                    ; SET UP JUMPS INTO CP/M IN LOWER MEMORY.
                    ;
5A7C 3EC3           SETUP:  MVI   A,0C3H      ;PUT JMP TO WBOOT
5A7E 320000             STA   0               ;ADR AT ZERO.
5A81 21035A             LXI   H,WBOOTE
5A84 220100             SHLD  1
5A87 320500             STA   5
5A8A 21064D             LXI   H,BDOS          ;PUT JUMP TO BDOS
5A8D 220600             SHLD  6               ;AT ADR 5,6,7.
5A90 218000             LXI   H,80H           ;SET DEFAULT DMA ADR.
5A93 22B35D             SHLD  DMAADD
5A96 C9                 RET                   ;RETURN FROM SETUP.
                    ;
                    ; WARM-BOOT:  READ ALL OF CPM BACK IN
                    ; EXCEPT BIOS, THEN JUMP TO CCP.
                    ;
5A97 318000         WBOOT:  LXI   SP,80H      ;SET STACK POINTER.

                            IF  INTRP         ;IF INTERRUPTS ALLOWED,
                            EI                ;ALLOW THEM HERE.
                            ENDIF

                            IF    LSTPAG      ;IF LIST DEVICE PAGING,
                            XRA   A           ;RESET LINE-FEED COUNT.
                            STA   LFCNT
                            ENDIF

5A9A 3AB55D                 LDA   DISKNO      ;SAVE DISK NUMBER.
5A9D F5                     PUSH  PSW         ;SAVE ON STACK
5A9E 0E00                   MVI   C,0         ;SELECT DRIVE A
5AA0 CD215B                 CALL  SELDSK      ;SELECT DRIVE A
5AA3 CDBF5B                 CALL  HOME        ;HOME THE DRIVE
```

```
5AA9 22BC5D             SHLD DRVFLG     ;CLEAR DRIVE FLAGS
5AAC 22BE5D             SHLD DRVFLG+2
5AAF 0619               MVI  B,NSECTS   ;GET # SECTORS FOR CPM READ.
5AB1 0E02               MVI  C,2        ;TRACK (B)=0, SECTOR (C)=2.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        DI              ;DISABLE THEM HERE.
                        ENDIF

5AB3 210045             LXI  H,CPMB     ;GET STARTING ADDRESS.
5AB6 22B35D     RBLK1:  SHLD DMAADD     ;SET STARTING ADDRESS.
5AB9 CD415C             CALL SETSEC     ;READ STARTING AT SECTOR IN C.
5ABC C5                 PUSH B
5ABD CD7F5C             CALL READ
5AC0 C1                 POP  B
5AC1 C2D05A             JNZ  RDERR      ;IF ERROR, PRINT MESSAGE.
5AC4 0C                 INR  C          ;INCREMENT SECTOR NUMBER.
5AC5 05                 DCR  B          ;DECREMENT SECTOR COUNT.
5AC6 C2B65A             JNZ  RBLK1      ;ALL DONE WHEN D=0.
5AC9 F1         ALDON:  POP  PSW        ;RESTORE DISK NUMBER.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        EI              ;ALLOW THEM AGAIN HERE.
                        ENDIF

5ACA 32B55D             STA  DISKNO
5ACD C3725A             JMP  GOCPM      ;GO BACK TO CPM.
                ;
5AD0 21485D     RDERR:  LXI  H,BTMSG    ;GET ADDRESS OF "BOOT ERROR".
5AD3 CD235D             CALL PMSG       ;PRINT IT.
5AD6 CDE55A             CALL CONIN      ;READ A CHAR FROM CONSOLE.
5AD9 C3975A             JMP  WBOOT      ;DO A WARM BOOT.
                ;
                ; CHECK CONSOLE INPUT STATUS.
                ;
5ADC DB00       CONST:  IN   CSTAT      ;READ CONSOLE STATUS.
5ADE E601               ANI  CKBR       ;LOOK AT KB READY BIT.
5AE0 3E00               MVI  A,0        ;SET A=0 FOR RETURN.

                        IF   RDYLO      ;IF STATUS READY LOW,
                        RNZ             ;NOT READY WHEN NOT 0.
                        ENDIF

                        IF   RDYHI      ;IF STATUS READY HIGH,
5AE2 C8                 RZ              ;NOT READY WHEN ZERO.
                        ENDIF

5AE3 2F                 CMA             ;IF READY A=FF.
5AE4 C9                 RET             ;RETURN FROM CONST.
                ;
                ; READ A CHARACTER FROM CONSOLE.
                ;
                CONIN:
                        IF NOT SOLOS    ;IF NOT PROC TECH SOLOS,
5AE5 DB00               IN   CSTAT      ;READ CONSOLE STATUS.
5AE7 E601               ANI  CKBR       ;IF NOT READY,
                        ENDIF

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                        CALL KBD        ;READ SOL KEYBOARD.
                        JZ   CONIN      ;READY WHEN NOT ZERO.
                        ENDIF

                        IF RDYLO AND NOT SOLOS
                        JNZ  CONIN      ;LOOP UNTIL LOW.
                        ENDIF
```

```
                        IF   RDYHI        ;IF READY WHEN HIGH,
5AE9 CAE55A             JZ   CONIN        ;LOOP UNTIL HIGH.
                        ENDIF

                        IF NOT SOLOS      ;IF NOT PROC TECH SOLOS,
5AEC DB01               IN   CDATA        ;READ A CHARACTER.
                        ENDIF

5AEE E67F               ANI  7FH          ;MAKE MOST SIG. BIT = 0.

                        IF BACKSP         ;IF BACKSPACE ACTIVATED,
5AF0 FE7F               CPI  7FH          ;IS IT A RUBOUT?
5AF2 C0                 RNZ               ;RETURN IF NOT.
5AF3 3EFF               MVI  A,0FFH       ;SET NO PRINT FLAG.
5AF5 32B65D             STA  CONOTF
5AF8 3E7F               MVI  A,7FH        ;RESTORE RUBOUT.
                        ENDIF

5AFA C9                 RET               ;RETURN FROM CONIN.
                ;
                ; WRITE A CHARACTER TO THE CONSOLE DEVICE.
                ;
                CONOT:
                        IF   BACKSP       ;IF BACKSPACE ACTIVATED,
5AFB 79                 MOV  A,C          ;GET CHARACTER.
5AFC FE7F               CPI  7FH          ;IS IT A RUBOUT?
5AFE C8                 RZ                ;IF SO, DON'T PRINT IT.
5AFF 3AB65D             LDA  CONOTF       ;GET NO PRINT FLAG.
5B02 B7                 ORA  A            ;SET CPU FLAGS.
5B03 CA165B             JZ   CONOTA       ;NOT SET, SO PRINT.
5B06 AF                 XRA  A            ;RESET THE FLAG
5B07 32B65D             STA  CONOTF       ;TO ZERO.
5B0A 0E08               MVI  C,8          ;PRINT BACKSPACE.
5B0C CD165B             CALL CONOTA
5B0F 0E20               MVI  C,20H        ;PRINT SPACE.
5B11 CD165B             CALL CONOTA
5B14 0E08               MVI  C,8          ;ANOTHER BACKSPACE.
                CONOTA:
                        ENDIF

                        IF CONUL AND NOT VIDEO  ;IF NULLS REQUIRED,
                        MVI  A,0DH        ;IF IT'S A CR,
                        CMP  C            ;THEN HOP OUT
                        JZ   CONULL       ;TO NULL ROUTINE.
                        ENDIF
                CONOT1:
                        IF NOT VIDEO AND NOT SOLOS
5B16 DB00               IN   CSTAT        ;READ CONSOLE STATUS.
5B18 E602               ANI  CPTR         ;IF NOT READY,
                        ENDIF

                        IF RDYLO AND NOT VIDEO AND NOT SOLOS
                        JNZ  CONOT1       ;LOOP UNTIL LOW.
                        ENDIF

                        IF RDYHI AND NOT VIDEO  ;IF READY WHEN HIGH,
5B1A CA165B             JZ   CONOT1       ;LOOP UNTIL HIGH.
                        ENDIF

                        IF NOT VIDEO AND NOT SOLOS
5B1D 79                 MOV  A,C          ;GET CHARACTER.
5B1E D301               OUT  CDATA        ;PRINT IT.
5B20 C9                 RET               ;RETURN.
                        ENDIF
```

```
          IF CONUL AND NOT VIDEO
CONULL:   PUSH B            ;SAVE B&C.
          MVI  B,CNULL+1    ;GET NULL COUNT.
CONUL1:   CALL CONOT1       ;PRINT CR.
          MVI  C,0          ;GET NULL CHAR.
          DCR  B            ;DECREMENT COUNTER.
          JNZ  CONUL1       ;DO NEXT NULL.
          POP  B            ;RESTORE B&C.
          MOV  A,C          ;RESTORE A.
          RET               ;RETURN.
          ENDIF

          IF   SOLOS        ;IF PROC TECH SOLOS,
          PUSH B            ;SAVE B&C.
          MOV  B,C          ;PUT CHAR IN B REG.
          CALL SCRN         ;OUTPUT CHAR TO SOLOS.
          POP  B            ;RESTORE B&C.
          MOV  A,C          ;PUT CHAR IN A.
          RET               ;RETURN FROM CONOT.
          ENDIF

          IF   VIDEO        ;IF ANY VIDEO BOARD,
;
;         VIDEO DRIVER FOR VB1-B OR VDM 1 BOARD.
;         WRITTEN BY G.W.MULCHIN
;         9-16-78
;
          MOV A,C           ;GET THE CHAR INTO REG A
          PUSH H            ;SAVE REGISTERS
          PUSH D
          PUSH B
          PUSH PSW          ;CHAR. IS IN REG A
          CALL VIDPRO       ;DO VIDEO ROUTINE
          POP PSW           ;RESTORE REGISTERS
          POP B
          POP D
          POP H
          RET               ;BACK TO CALLER
;
VIDPRO:   LHLD VDMP         ;GET SCREEN POSITION POINTER
          CPI CR            ;IS IT CARRIAGE RETURN?
          JZ CARRET
          CPI LF            ;IS IT LINEFEED?
          JZ LFNOT          ;PUT IN A BLANK
          CPI BKSP          ;IS IT A RUBOUT?
          JZ BS
          CPI  FF           ;IS IT CNTL-L?
          JZ CLEAR
          MOV M,A           ;IT HAS TO BE DATA
UPDATE:   INX H             ;UPDATE POSITION
GONE:     MVI M,0A0H        ;PUT CURSOR ON SCREEN
          JMP MAXLIN        ;CHECK FOR LINE > 64
LFNOT:    MVI M,' '         ;PUT IN A SPACE
          JMP UPDATE        ;GET OUT NOW
BS:       MVI M,' '
          LHLD VDMP         ;GET CURRENT POSITION
          DCX H
          SHLD VDMP         ;SAVE CURSOR POSITION
          JMP GONE
CARRET:   MVI M,' '         ;CHAR. IS A CARRIAGE RET.
          MOV A,L           ;UPDATE NEXT POSITION
          ANI 0C0H
          ADI 40H           ;SET UP FOR NEW LINE
          EOV L,A           ;ADDRESS OF NEW LINE
          MVI A,0
          ADC H
```

```
                        MOV H,A
                MAXLIN: SHLD VDMP       ;SAVE POINTER FOR NEXT CHAR.
                        MVI A,7FH
                        ANA L
                        RNR             ;EXIT BACK TO MAIN PROGRAM
                        MVI M,' '
                        LXI H,SCRNTOP
                        SHLD VDMP
                        LXI H,LINES15   ;15 LINES OF SCREEN DATA
                        LXI D,SCREEN    ;TOP OF SCREEN. SET UP
                        LXI B,SCRLCNT   ; TO SCROLL 15 LINES
                SCROLL: MOV A,M         ;START SCROLLLING UP
                        STAX D          ;STUFF REG A BY WAY OF D,E
                        INX H
                        INX D
                        DCX B
                        XRA A
                        CMP B           ;15 LINES YET?
                        JNZ SCROLL
                        CMP C
                        JNZ SCROLL      ;NOT DONE YET!
                        LXI H,SCRNTOP
                BLANK:  MVI M,' '       ;PUT BLANK ON SCREEN
                        INX H           ;BLANK ENTIRE DATA LINE
                        MOV A,L
                        ANI 3FH
                        JNZ BLANK
                        LXI H,SCRNTOP
                        MVI M,0AOH      ;STUFF CURSOR BACK
                        RET             ;ALL DONE.
                CLEAR:  LXI H,SCREEN    ;CLEAR SCREEN
                        MVI A,ENDSCR    ;THIS IS END CHECK
                CLERA:  CMP H           ;IS IT END YET?
                        JZ FINISH
                        MVI M,' '       ;PUT SPACE ON SCREEN
                        INX H           ;BUMP POINTER
                        JMP CLERA       ;GO BACK IF NOT DONE
                FINISH: LXI H,SCRNTOP
                        MVI M,0AOH      ;STUFF CURSOR BACK AGAIN
                        SHLD VDMP       ;SAVE CURSOR POSITION.
                        RET             ;ALL DONE.
                        ENDIF           ;END OF VDM DRIVER.
                ;
                ; SELECT DISK NUMBER ACCORDING TO REGISTER C.
                ; ALSO CHECK IF DISK HAS BEEN LOGGED IN BEFORE.
                ;  IF NOT, LOG IT IN AND CHECK DENSITY
                ;
5B21 79         SELDSK: MOV  A,C        ;GET NEW DISK NUMBER.
5B22 E603               ANI  3          ;ONLY LOOK AT 2 LSB'S.
5B24 21B55D             LXI  H,DISKNO   ;GET ADR OF OLD DISK NO.
5B27 BE                 CMP  M          ;NEW = OLD?
5B28 C8                 RZ              ;IF SO, RETURN.
5B29 F5                 PUSH A          ;SAVE DISK NUMBER.
5B2A 3AC15D             LDA  NODSKS     ;GET NUMBER OF DISKS.
5B2D 3D                 DCR  A          ;IF MORE THAN ONE DISK,
5B2E C2465B             JNZ  SELMOR     ;TAKE CARE OF IT.
5B31 215C5D             LXI  H,MNTMSG   ;GET ADR OF MOUNT MESSAGE.
5B34 CD235D             CALL PMSG       ;PRINT "MOUNT ".
5B37 F1                 POP  A          ;GET DISK NUMBER.
5B38 32B55D             STA  DISKNO     ;UPDATE OLD WITH NEW.
5B3B C641               ADI  'A'        ;ADD ASCII FOR 'A'.
5B3D 4F                 MOV  C,A        ;PUT INTO C.
5B3E CDFB5A             CALL CONOT      ;PRINT IT.
5B41 CDE55A             CALL CONIN      ;READ A CARRIAGE RETURN.
5B44 AF                 XRA  A          ;SET A=0 FOR NO ERRO IND.
5B45 C9                 RET             ;RETURN FROM SELDSK
```

```
5B46 F1          SELMOR: POP  A           ;MAKE STACK RIGHT.
5B47 7E                  MOV  A,M         ;GET OLD DISK NUMBER.
                         IF   DUAL        ;IF DUAL DRIVE,
                         ANI  0FEH        ;CLEAR OUT BIT 0.
                         ENDIF

5B48 5F                  MOV  E,A         ;PUT OLD DISK NO. IN D&E.
5B49 1600                MVI  D,0
5B4B 21B85D              LXI  H,TRTAB     ;GET ADDRESS OF TRACK TABLE.
5B4E E5                  PUSH H           ;SAVE FOR LATER
5B4F 19                  DAD  D           ;ADD DISK NO. TO ADDRESS.
5B50 DBF9                IN   TRACK       ;READ 1771 TRACK REGISTER.
5B52 77                  MOV  M,A         ;PUT INTO TABLE.
5B53 79                  MOV  A,C         ;GET NEW DISK NUMBER.

                         IF   DUAL        ;IF A DUAL DRIVE,
                         ANI  0FEH        ;CLEAR BIT 0.
                         ENDIF

5B54 5F                  MOV  E,A         ;PUT NEW DISK NO. IN D&E.
5B55 E1                  POP  H           ;RESTORE ADDRESS OF TRACK TABLE
5B56 19                  DAD  D           ;ADD DISK NO. TO ADDRESS.
5B57 7E                  MOV  A,M         ;GET NEW TRACK NUMBER.
5B58 D3F9                OUT  TRACK       ;PUT INTO 1771 TRACK REG.
5B5A 79                  MOV  A,C         ;UPDATE OLD DISK NUMBER.
5B5B 32B55D              STA  DISKNO
5B5E 87                  ADD  A           ;PUT BITS 1&2 AT 4&5.
5B5F 87                  ADD  A
5B60 87                  ADD  A
5B61 87                  ADD  A
5B62 32C45D              STA  LATCH       ;SAVE NEW LATCH CODE.
                ;
                ;SELECT DENSITY BASED ON BYTE VALUE
                ;IN TRACK 0 SECTOR 1 ADDRESS 7EH.
                ;IF THIS BYTE IS A 0DDH, THEN THE
                ;DISK IS DOUBLE DENSITY, ELSE, IT'S
                ;SINGLE DENSITY.
                ;
5B65 C5         DENSITY:PUSH B           ;SAVE B,C
5B66 21BC5D             LXI H,DRVFLG     ;INDEX INTO DRIVE BYTE FLAG
5B69 0600               MVI B,0          ;ZERO REG B
5B6B 09                 DAD B            ;ADD THE DRIVE NUMBER
5B6C 7E                 MOV A,M          ;GET THE BYTE FLAG
5B6D B7                 ORA A            ;SET THE FLAGS
5B6E FA975B             JM LOGED         ;SKIP IF LOGGED IN BEFORE
5B71 E5                 PUSH H           ;SAVE DRIVE TABLE POINTER.
5B72 3AC45D             LDA  LATCH
5B75 D3FC               OUT  DCONT       ;SELECT DRIVE
5B77 2AB35D             LHLD DMAADD      ;GET PRESENT DMA ADDRESS
5B7A E5                 PUSH H           ;SAVE ON STACK
5B7B CDBF5B             CALL HOME        ;HOME DRIVE
5B7E 3E01               MVI A,1          ;SET FOR SECTOR 1
5B80 32B25D             STA  SECT        ;SAVE SECTOR TO READ
5B83 21C65D             LXI H,DBUFF      ;POINT TO DMA BUFFER
5B86 22B35D             SHLD DMAADD      ;SET UP THE DMA ADDRESS
5B89 CD7F5C             CALL READ        ;READ TRK 0, SEC 1 INTO DBUFF
5B8C E1                 POP  H           ;RECOVER DMAADD FROM STACK
5B8D 22B35D             SHLD DMAADD      ;RESTORE IT
5B90 E1                 POP  H           ;RESTORE DRIVE TABLE POINTER.
5B91 3A445E             LDA DBUFF+7EH    ;GET THE DENSITY CODE BYTE
5B94 F680               ORI  80H         ;set logged in bit
5B96 77                 MOV  M,A         ;place it in flag table.
5B97 010700     LOGED:  LXI  B,7         ;index value through drive table.
5B9A E612               ANI  12H         ;MASK DENSITY AND SIDE BITS OUT.
5B9C B7                 ORA  A           ;SINGLE DENSITY?
5B9D 212D5A             LXI  H,SDTAB     ;point to start of tables
```

```
5BA0 CAA45B             JZ   DENSIT1     ;yes, overlay param. block
                ;
                        IF   DUBSID
                        DAD  B           ;no, add offset to next table
                        CPI  2           ;single den doub sided?
                        JZ   DENSIT1     ;yes
                        DAD  B           ;no
                        CPI  10H         ;doub den single sided?
                        JZ   DENSIT1     ;yes
                        ENDIF
                ;
5BA3 09                 DAD  B           ;no, must be doub den , doub sided
5BA4 EB         DENSIT1:XCHG             ;drive table pointer --> d,e
5BA5 1A                 LDAX D           ;GET LOG BYTE
5BA6 13                 INX  D           ;BUMP POINTER
5BA7 32C05D             STA  DENS        ;SET DENSITY
5BAA 1A                 LDAX D           ;GET A BYTE FROM TABLE
5BAB 13                 INX  D           ;BUMP POINTER
5BAC 32154D             STA BDOS+15      ;CHANGE TRANS FUNCTION
5BAF 213A4D             LXI H,BDOS+52    ;POINT TO CONSTANT DATA AREA (CP/M)
5BB2 0605               MVI B,5          ;COUNT = 5
5BB4 1A         MOVE:   LDAX D           ;GET A BYTE FROM TABLE
5BB5 77                 MOV M,A          ;OVERLAY CP/M
5BB6 13                 INX D            ;BUMP
5BB7 23                 INX H            ; POINTERS
5BB8 05                 DCR B            ;DECREASE COUNT
5BB9 C2B45B             JNZ MOVE         ; AND LOOP TILL DONE
5BBC C1                 POP  B           ;RESTORE B,C
5BBD AF                 XRA  A           ;SET A = 0.
5BBE C9                 RET              ;RETURN FROM SELDSK.
                ;
                ; MOVE DISK TO TRACK ZERO.
                ;
5BBF 3E01       HOME:   MVI  A,0+STPRAT ;RESTORE
5BC1 D3F8               OUT  DCOM
5BC3 0E00               MVI  C,0         ;TRACK 0
                ;
                ; SET TRACK NUMBER TO WHATEVER IS IN REGISTER C.
                ; ALSO PERFORM MOVE TO THE CORRECT TRACK (SEEK).
                ;
5BC5 2AC45D     SETTRK: LHLD LATCH               ;set new and old latch.
5BC8 7C                 MOV  A,H                 ;set latch value.
5BC9 E6B7               ANI  0B7H                ;strip density and side bits.
5BCB 67                 MOV  H,A                 ;restore it.
                ;
                        IF   DUBSID              ;if using double sided drive.
                        LDA  DENS                ;check if double sided.
                        RRC
                        RRC                      ;look at bit 1.
                        JNC  NOTSID              ;if bit 1 = 0, it's single side
                        MOV  A,C                 ;it's doub sided, so set track
                        RRC                      ;divide by 2.
                        MOV  B,A                 ;save it in reg b.
                        MOV  A,L                 ;set old latch value.
                        JC   SIDE2               ;change side if odd track.
                        ANI  0BFH                ;clear side bit from latch.
                        JMP  SETLAT              ;go set the latch.
                ;
                SIDE2:  ORI  40H                 ;turn on side select bit.
                SETLAT: STA  CLATCH              ;save it for later.
                        ANI  0BFH                ;clear side bit.
                        CALL OLDLAT              ;check for drive change.
                        MOV  A,B                 ;restore doub sided trk number.
                        ANI  7FH                 ;clear bit 7.
                        MOV  C,A                 ;trk number now in reg c.
                        JMP  TRKSET              ;check for density of track ad
```

```
                        ENDIF
                ;
                        IF   NOT DUBSID          ;if not using double sided drive
5BCC C3DA5B             JMP  NOTSID              ;jump around subroutine.
                        ENDIF
                ;
5BCF BC         OLDLAT: CMP  H                   ;new = old?
5BD0 3EFF               MVI  A,0FFH              ;if not, set = ff
5BD2 C2D65B             JNZ  SFLAG
5BD5 2F                 CMA                      ;new = old, set = 0.
5BD6 32B75D     SFLAG:  STA  HLSF                ;save head load select flag.
5BD9 C9                 RET
                ;
5BDA 7D         NOTSID: MOV  A,L                 ;set latch value.
5BDB 32C55D             STA  CLATCH              ;save it
5BDE CDCF5B             CALL OLDLAT              ;check for drive change.
                ;
5BE1 3AC05D     TRKSET: LDA  DENS        ;CHECK DRIVE DENSITY FLAG
5BE4 0F                 RRC              ;IS IT ZERO ?
5BE5 D2F65B             JNC  TRKSD       ;YES, WE ARE SINGLE DENSITY
5BE8 79                 MOV  A,C         ;RESTORE TRACK NUMBER
5BE9 FE01               CPI  1           ;IS IT TRACK 1 ?
5BEB DAF65B             JC   TRKSD       ;IF LESS THEN, SET SINGLE DEN.
5BEE 3AC55D             LDA  CLATCH      ;GET THE CURRENT LATCH VALUE
5BF1 F608               ORI  DDEN        ;SET BIT 4 ON (DOUB DENSITY ON)
5BF3 C3FB5B             JMP  TRKDD
5BF6 3AC55D     TRKSD:  LDA  CLATCH      ;GET CURRENT LATCH VALUE
5BF9 E6F7               ANI  DDDSB       ;TURN OFF BIT 4 (SINGLE DENSITY ON)
5BFB 32C55D     TRKDD:  STA  CLATCH      ;SAVE NEW CURRENT LATCH VALUE
5BFE D3FC               OUT  DCONT       ;  AND SET THE HARDWARE LATCH
5C00 79                 MOV  A,C         ;RESTORE TRACK NUMBER
5C01 32B15D             STA  TRK         ;UPDATE OLD WITH NEW.
                ;
                ; MOVE THE HEAD TO THE TRACK IN REGISTER A.
                ;
5C04 C5         SEEK:   PUSH B           ;SAVE B&C.
5C05 47                 MOV  B,A         ;SAVE DESTINATION TRACK.
5C06 3E0A               MVI  A,RTCNT     ;GET RETRY COUNT.
5C08 32C35D     SRETRY: STA  SERCNT      ;STORE IN ERROR COUNTER.
5C0B DBF9               IN   TRACK       ;READ PRESENT TRACK NO.
5C0D 4F                 MOV  C,A         ;SAVE IN C.
5C0E 79                 MOV  A,C         ;DELAY.
5C0F B8                 CMP  B           ;SAME AS NEW TRACK NO.?
5C10 78                 MOV  A,B
5C11 C2165C             JNZ  NOTHR       ;JUMP IF NOT THERE.
5C14 C1         THERE:  POP  B           ;RESTORE B&C.
5C15 C9                 RET              ;RETURN FROM SEEK.
                NOTHR:
                ;DELAY LOOP TO ALLOW TUNNEL
                ;ERASE TO END DURING WRITE.
                ;
5C16 F5                 PUSH PSW
5C17 3ED0               MVI  A,0D0H      ;COUNT = 208
5C19 3D         BUSY1:  DCR  A           ;LOOP
5C1A C2195C             JNZ  BUSY1       ;  TILL ZERO
5C1D F1                 POP  PSW
5C1E 78                 MOV  A,B         ;RESTORE A FROM B.
5C1F D3FB               OUT  DDATA       ;TRACK TO DATA REGISTER.
5C21 3E1D               MVI  A,14H+STPRAT+HLAB  ;GET STEP RATE, DO
5C23 D3F8               OUT  DCOM        ;SEEK WITH VERIFY.

                        IF   NOT DMACNTL
5C25 DBFC               IN   WAIT
5C27 DBF8               IN   DSTAT       ;CHECK STATUS
                        ENDIF
```

```
                        IF    DMACNTL
                BUSY2:  CALL  SLOOP       ;USE DMA CHECK PORT
                        ENDIF

5C29 E691               ANI   91H         ;LOOK AT BITS.
5C2B CA145C             JZ   THERE        ;OK IF ZERO.

                        IF TESTING        ;IF TESTING FOR ERRORS
                        PUSH H            ;SAVE H&L.
                        LXI   H,SECNT     ;GET ADR OF SEEK ERR CTR.
                        INR   M           ;ONE MORE SEEK ERROR.
                        POP   H           ;RESTORE H&L.
                        ENDIF

5C2E 3AC35D             LDA   SERCNT      ;GET ERROR COUNT.
5C31 3D                 DCR   A           ;DECREMENT COUNT.
5C32 C2085C             JNZ   SRETRY      ;RETRY SEEK.
5C35 C1                 POP   B           ;RESTORE B&C.
5C36 21545D             LXI   H,SKMSG     ;PRINT "SEEK ".
5C39 DBF8               IN    DSTAT       ;READ DISK STATUS.
5C3B E691               ANI   91H         ;LOOK AT ERROR BITS.
5C3D 57                 MOV   D,A         ;PUT IN REG D.
5C3E C3A85C             JMP   ERMSG       ;DO COMMON ERR MESSAGES.
                ;
                ; SET DISK SECTOR NUMBER.
                ;
5C41 79         SETSEC:  MOV  A,C         ;GET SECTOR NUMBER.
5C42 32B25D              STA  SECT        ;PUT AT SECT # ADDRESS.
5C45 C9                  RET              ;RETURN FROM SETSEC.
                ;
                ; SET DISK DMA ADDRESS.
                ;
5C46 60         SETDMA:  MOV  H,B         ;MOVE B&C TO H&L.
5C47 69                  MOV  L,C
5C48 22B35D              SHLD DMAADD      ;PUT AT DMA ADR ADDRESS.
5C4B C9                  RET              ;RETURN FROM SETDMA.
                ;
                ; HDLD - GET HEAD-LOAD BIT IF REQUIRED.
                ;
5C4C 3AB75D     HDLD:    LDA  HLSF        ;GET HEAD-LOAD FLAG.
5C4F B7                  ORA  A           ;IS A = ZERO?
5C50 CA615C              JZ   HDLD1       ;HOP IF SO.
5C53 2F                  CMA              ;SET A = 0.
5C54 32B75D              STA  HLSF        ;SET FLAG = 0 IF NOT.
                ;
                ; IF CHANGING TO A NEW DRIVE, PERFORM A SEEK
                ; TO THE SAME TRACK TO ALLOW THE HEAD TO UNLOAD.
                ;
5C57 DBF9                IN   TRACK       ;GET PRESENT TRACK
5C59 D3FB                OUT  DDATA       ; AND TELL CONTROLLER
5C5B 3E15                MVI  A,14H+STPRAT       ;GET STEP COMMAND
5C5D D3F8                OUT  DCOM        ;SEND IT TO FLOPPY CONTROLLER

                         IF   NOT DMACNTL
5C5F DBFC                IN   WAIT        ;WAIT FOR INTRQ
                         ENDIF

                         IF   DMACNTL
                         CALL SLOOP       ;USE DMA CHECK PORT
                         ENDIF

5C61 DBF8       HDLD1:   IN   DSTAT       ;READ 1771 STATUS.
5C63 E620                ANI  20H         ;LOOK AT HL BIT.
5C65 3E04                MVI  A,4
5C67 C8                  RZ
                         OUR  A           ;HEAD IS LOADED
```

```
5C69 C9                 RET             ;RETURN FROM HDLD.
                ;
                        IF    NOT DMACNTL
5C6A 32C25D     DSKSET: STA   ERCNT     ;SAVE RETRY COOUNT
5C6D 3ED0               MVI   A,0D0H    ;CAUSE INTRP
5C6F D3F8               OUT   DCOM
5C71 E3                 XTHL            ;SOME
5C72 E3                 XTHL            ;DELAY
                        ENDIF

                        IF    INTRP
                        DI
                        ENDIF

                        IF    NOT DMACNTL
5C73 2AB35D             LHLD DMAADD     ;STARTING ADDRESS
5C76 3AB25D             LDA   SECT      ;GET SECTOR NUMBER
5C79 D3FA               OUT   SECTP     ;TELL CONTROLLER
5C7B CD4C5C             CALL HDLD       ;CHECK FOR HEAD LOADED
5C7E C9                 RET
                        ENDIF
                ;
                ; READ THE SECTOR AT SECT, FROM THE PRESENT TRACK.
                ; USE STARTING ADDRESS AT DMAADD.
                ;
5C7F 3E0A       READ:   MVI   A,RTCNT   ;GET RETRY COUNT.
                RRETRY:
                        IF    DMACNTL
                        LXI   B,80D0H   ;FLOPPY READ, FORCE INTRP.
                        LXI   D,4080H   ;DMA READ, BYTE COUNT.
                        CALL DMARW      ;COMMON READ/WRITE ROUTINE.
                        ENDIF

                        IF  INTRP       ;IF INTERRUPTS ALLOWED,
                        DI              ;DISABLE THEM HERE.
                        ENDIF

                        IF    NOT DMACNTL
5C81 0680               MVI   B,80H     ;FLOPPY READ COMMAND.
5C83 CD6A5C             CALL DSKSET     ;SET UP DISK CONTROLLER
5C86 B0                 ORA   B         ;SET READ COMMAND
5C87 D3F8       READE:  OUT   DCOM      ;SEND COMMAND TO 1771.
5C89 DBFC       RLOOP:  IN    WAIT      ;WAIT FOR DRQ OR INTRQ.
5C8B B7                 ORA   A         ;SET FLAGS.
5C8C F2965C             JP    RDDONE    ;DONE IF INTRQ.
5C8F DBFB               IN    DDATA     ;READ A DATA BYTE FROM DISK.
5C91 77                 MOV   M,A       ;PUT BYTE INTO MEMORY.
5C92 23                 INX   H         ;INCREMENT MEMORY POINTER.
5C93 C3895C             JMP   RLOOP     ;KEEP READING.
5C96 DBF8       RDDONE: IN    DSTAT     ;READ DISK STATUS.
                        ENDIF

                        IF  INTRP       ;IF INTERRUPTS ALLOWED,
                        EI              ;ALLOW AGAIN HERE.
                        ENDIF

5C98 E69D               ANI   9DH       ;LOOK AT ERROR BITS.
5C9A C8                 RZ              ;RETURN IF NONE.
5C9B CDC75C             CALL ERCHK      ;CHECK FOR SEEK ERROR.

                        IF  TESTING
                        LXI   H,RECNT   ;GET RD ERR COUNT ADDR.
                        INR   M         ;ONE MORE ERROR.
                        MOV   A,M
                        CMA
                        OUT   0FFH
```

```
                        ENDIF

5C9E 3AC25D             LDA   ERCNT       ;GET ERROR COUNT.
5CA1 3D                 DCR   A           ;DECREMENT COUNT.
5CA2 C2815C             JNZ   RRETRY      ;TRY TO READ AGAIN.
5CA5 21375D             LXI   H,RDMSG     ;PRINT "READ ".
5CA8 CD235D     ERMSG:  CALL  PMSG        ;PRINT ORIGIN MESSAGE.
                ERMSG1:
                        IF NOT VIDEO      ;NEED MORE ROOM?
5CAB 7A                 MOV   A,D         ;GET ERROR BITS.
5CAC E610               ANI   10H         ;IF BIT 4 IS HIGH,
5CAE 212E5D             LXI   H,RNMSG     ;PRINT "RECORD NOT FOUND"
5CB1 C4235D             CNZ PMSG
5CB4 7A                 MOV   A,D         ;GET ERROR BITS.
5CB5 E608               ANI   8H          ;IF BIT 3 IS HIGH,
5CB7 21325D             LXI   H,CRCMSG    ;PRINT "CRC ERROR".
5CBA C4235D             CNZ PMSG
                        ENDIF

5CBD 214D5D             LXI   H,ERRMSG    ;PRINT "ERROR."
5CC0 CD235D             CALL  PMSG
5CC3 3E01               MVI   A,1         ;SET FOR PERM ERR MSG.
5CC5 B7                 ORA   A           ;SET FLAGS.
5CC6 C9                 RET
                ;
                ; ERCHK - CHECK FOR RECORD NOT FOUND ERROR.
                ;
5CC7 57         ERCHK:  MOV   D,A         ;SAVE ERROR BITS IN D.
5CC8 E610               ANI   10H         ;IF RECORD NOT FOUND,
5CCA C8                 RZ
                ;
                ;CHECK FOR SEEK TO CORRECT TRACK,
                ;AND CHANGE IF NECESSARY.
5CCB 3EC4       CHKSK:  MVI   A,0C4H      ;SEND COMMAND TO 1771
5CCD D3F8               OUT   DCOM        ;TO READ ADDRESS.

                        IF    NOT DMACNTL
5CCF DBFC               IN    WAIT        ;WAIT FOR DRQ OR INTRQ.
                        ENDIF

                        IF    DMACNTL
                        CALL SLOOP
                        ENDIF

5CD1 DBFB               IN    DDATA       ;READ THE TRACK ADDRESS.
5CD3 47                 MOV   B,A         ;SAVE IN REGISTER B.
5CD4 DBFC       CHKS2:  IN    WAIT        ;WAIT FOR INTRQ.
5CD6 B7                 ORA   A           ;SET FLAGS.
5CD7 F2DF5C             JP    CHKS3       ;DONE WITH READ ADR OP.
5CDA DBFB               IN    DDATA       ;READ ANOTHER BYTE.
5CDC C3D45C             JMP   CHKS2       ;DO IT AGAIN.
5CDF DBF8       CHKS3:  IN    DSTAT       ;READ DISK STATUS.
5CE1 B7                 ORA   A           ;SET FLAGS.
5CE2 CAEE5C             JZ    CHKS4       ;READ ADR OK IF 0.
5CE5 CDBF5B             CALL  HOME        ;OTHERWISE, HOME FIRST.
5CE8 3AB15D     CHKS5:  LDA   TRK
5CEB C3045C             JMP   SEEK
5CEE 78         CHKS4:  MOV   A,B         ;UPDATE TRACK REGISTER.
5CEF D3F9               OUT   TRACK
5CF1 C3E85C             JMP   CHKS5       ;RETURN FROM ERCHK.
                ;
                ; WRITE THE SECTOR AT SECT, ON THE PRESENT TRACK.
                ; USE STARTING ADDRESS AT DMAADD.
                ;
5CF4 3E0A       WRITE:  MVI   A,RTCNT     ;GET RETRY COUNT.
                WRETRY:
```

```
                        IF    DMACNTL
                        LXI   B,0A0DOH    ;FLOPPY WRITE, FORCE INTRP
                        LXI   D,0808OH    ;DMA WRITE, BYTE COUNT
                        CALL  DMARW       ;COMMON ROUTINE
                        ENDIF

                        IF    NOT DMACNTL
5CF6 06A0               MVI   B,0A0H      ;WRITE COMMAND
5CF8 CD6A5C             CALL  DSKSET
5CFB B0                 ORA   B
5CFC D3F8       WRITE2: OUT   DCOM
5CFE DBFC       WLOOP1: IN    WAIT        ;WAIT FOR READY.
5D00 B7                 ORA   A           ;SET FLAGS.
5D01 F20B5D             JP    WDONE       ;HOP OUT WHEN DONE.
5D04 7E                 MOV   A,M         ;GET BYTE FROM MEM.
5D05 D3FB               OUT   DDATA       ;WRITE ONTO DISK.
5D07 23                 INX   H           ;INCREMENT MEM PTR.
5D08 C3FE5C             JMP   WLOOP1      ;KEEP WRITING.
5D0B DBF8       WDONE:  IN    DSTAT
                        ENDIF

                        IF  INTRP         ;IF INTERRUPTS ALLOWED,
                        EI                ;ENABLE AGAIN HERE.
                        ENDIF

5D0D E6FD               ANI   0FDH        ;LOOK AT THESE BITS.
5D0F C8                 RZ                ;RETURN IF NO ERR.
5D10 CDC75C             CALL  ERCHK       ;CHECK/CORRECT SEEK ERR.

                        IF  TESTING       ;IF TESTING FOR ERRORS
                        LXI   H,WECNT     ;GET ADR OF WRITE ERR CTR.
                        INR   M           ;ONE MORE WRITE ERROR.
                        MOV   A,M
                        CMA
                        OUT   0FFH
                        ENDIF

5D13 3AC25D             LDA   ERCNT       ;GET ERROR COUNT.
5D16 3D                 DCR   A           ;DECREMENT COUNT.
5D17 C2F65C             JNZ   WRETRY      ;TRY TO WRITE AGAIN.
5D1A 213F5D             LXI   H,WTMSG     ;PRINT "WRITE ".

                        IF NOT VIDEO      ;NEED MORE ROOM?
5D1D CD235D             CALL  PMSG
5D20 C3AB5C             JMP   ERMSG1      ;DO COMMON MESSAGES.
                        ENDIF

                        IF    VIDEO       ;WE NEED A RETURN.
                        JMP   ERMSG
                        ENDIF
                ;
                ; PRINT THE MESSAGE AT H&L UNTIL A ZERO.
                ;
5D23 7E         PMSG:    MOV   A,M        ;GET A CHARACTER.
5D24 B7                  ORA   A          ;IF IT'S ZERO,
5D25 C8                  RZ               ;RETURN.
5D26 4F                  MOV   C,A        ;OTHERWISE,
5D27 CDFB5A              CALL  CONOT      ;PRINT IT.
5D2A 23                  INX   H          ;INCREMENT H&L,
5D2B C3235D              JMP   PMSG       ;AND GET ANOTHER.
                ;
                ; CBIOS MESSAGES
                ;
                        IF NOT VIDEO      ;NEED MORE ROOM?
5D2E 49442000   RNMSG:  DB    'ID ',0
5D32 435243200CRCMSG:   DB    'CRC ',0
```

```
                        ENDIF

5D37 0D0A524541RDMSG:   DB    0DH,0AH,'Read ',0
5D3F 0D0A575249WTMSG:   DB    0DH,0AH,'Write ',0
5D48 424F4F5420BTMSG:   DB    'Boot '
5D4D 4552524F52ERRMSG:  DB    'ERROR.',0
5D54 0D0A534545SKMSG:   DB    0DH,0AH,'Seek ',0
5D5C 0D0A4D4F55MNTMSG:  DB    0DH,0AH,'Mount ',0
5D65 0D0A544152SMSG:    DB    0DH,0AH,'Tarbell '
5D6F 3234               DB    MSIZE/10+'0',MSIZE MOD 10 + '0'
5D71 4B2043504D         DB    'K CPM V1.4 of 7-14-80'
5D86 0D0A               DB    0DH,0AH

                        IF  TARBELL     ;IF USING TARBELL CPU.
                        DB  'Tarbell CPU, '
                        ENDIF

                        IF  STD         ;IF STANDARD I/O,
                        DB   'Standard '
                        ENDIF

                        IF  MSIO2       ;IF MITS 2SIO,
5D88 3253494F20         DB   '2SIO '
                        ENDIF

                        IF  ISIO2       ;IF IMSAI SIO-2,
                        DB  'SIO-2 '
                        ENDIF

                        IF  TUART       ;IF TUART,
                        DB  'Tuart '
                        ENDIF

                        IF  SOLOS       ;IF PROC TECH SOLOS,
                        DB  'Solos '
                        ENDIF

                        IF   VDM        ;IF PROC TECH VDM,
                        DB   'VDM '
                        ENDIF

                        IF   FLASH      ;IF VG FLASHWRITER,
                        DB   'Flashwriter '
                        ENDIF

                        IF   VB1        ;IF SSM VB1-B,
                        DB   'VB1 '
                        ENDIF

                        IF  DUBSID      ;IF DOUBLE-SIDED,
                        DB   'Double-Sided '
                        ENDIF

                        IF  DUAL        ;IF DUAL DRIVE,
                        DB  'Dual '
                        ENDIF

                        IF  DMACNTL     ;IF USING DMA CONTROL
                        DB  'DMA '
                        ENDIF
5D8D 5645522E           DB   'VER.'

5D91 0D0A484F57         DB   0DH,0AH,'How Many Disks? ',0
                ;
                ; WRITE A CHARACTER ON LISTING DEVICE.
```

```
                LIST:
                        IF LSTNUL           ;IF NULLS OR PAGING,
                        MVI  A,0DH          ;IF IT'S A CR,
                        CMP  C              ;THEN HOP OUT TO
                        JZ   LINUL          ;NULL ROUTINE.
                        ENDIF

                        IF LSTPAG           ;IF PAGING
                        MVI A,0AH           ;GET A LINEFEED
                        CMP C               ;DOES IT MATCH?
                        JZ LINUL3
                        ENDIF

5DA4 DB02       LTBSY:  IN   LSTAT          ;READ LISTER STATUS.

                        IF   NOT TARDEL
5DA6 E602               ANI  LRBIT          ;LOOK AT READY BIT.
                        ENDIF

                        IF   TARDEL
                        ANI  81H            ;MASK
                        XRI  81H
                        ENDIF

                        IF   TARDEL OR RDYLO    ;IF READY WHEN LOW,
                        JNZ  LTBSY          ;LOOP TILL LOW.
                        ENDIF

                        IF   NOT TARDEL AND RDYHI       ;IF READY WHEN HIGH,
5DA8 CAA45D             JZ   LTBSY          ;LOOP TILL HIGH.
                        ENDIF

5DAB 79                 MOV  A,C            ;GET DATA BYTE.
5DAC D303               OUT  LDATA          ;PRINT IT.
5DAE C9                 RET                 ;RETURN FROM LIST.

                        IF LSTNUL OR LSTPAG  ;IF NULLS OR PAGING,
                LINUL:  PUSH B              ;SAVE B&C.
                        MVI  B,(LNULL AND OFFH)+1 ;GET NULL COUNT
                LINUL1: CALL LTBSY          ;PRINT (CR FIRST).
                        MVI  C,0            ;GET NULL CHAR.
                        DCR  B              ;DECREMENT COUNTER.
                        JNZ  LINUL1         ;DO NEXT NULL.
                        JMP  LINUL2         ;EXIT THE ROUTINE.
                        ENDIF

                        IF   LSTPAG         ;IF LIST DEV. PAGING,
                LINUL3: PUSH B              ;SAVE B,C PAIR
                        LDA  LFCNT          ;GET LINE-FEED COUNT.
                        INR  A              ;INCREMENT IT.
                        STA  LFCNT          ;SAVE IT BACK.
                        CPI  LINCNT-(LINCNT/11)  ;END OF PAGE?
                        MVI  B,1            ;SET UP FOR 1 LF.
                        JNZ  NOTEOP         ;HOP IF NOT END.
                        XRA  A              ;SET LF COUNT = 0.
                        STA  LFCNT
                        MVI  B,(LINCNT/11)+1  ;BETWEEN PAGES.
                NOTEOP: MVI  C,0AH          ;GET LINE-FEED CODE.
                LSTPA1: CALL LTBSY          ;PRINT LINE-FEED.
                        DCR  B              ;DECREMENT LF COUNTER.
                        JNZ  LSTPA1         ;DO NEXT LINE FEED?
                        ENDIF

                        IF LSTNUL OR LSTPAG  ;IF NULLS OR PAGING,
                LINUL2: POP  B              ;RESTORE B&C.
                        MOV  A,C            ;RESTORE A
```

```
                        RET             ;RETURN FROM LIST.
                        ENDIF
                ;
                ; PUNCH PAPER TAPE.
                ;
5DAF C9         PUNCH:  RET             ;RETURN FROM PUNCH.
                ;
                ;  NORMALLY USED TO READ PAPER TAPE.
                ;
5DB0 C9         READER: RET             ;RETURN FROM READER.
                ;
                ;NOTE:  AS THERE ARE ONLY NINE SECTORS
                ;AVAILABLE FOR CBIOS ON THE SECOND SYSTEM TRACK (1),
                ;THE LAST ADDRESS BEFORE THIS POINT SHOULD BE NO
                ;GREATER THAN THE CBIOS STARTING ADDRESS + 047F (HEX).
                ;THIS WILL NORMALLY BE XE7F (HEX).
                ;
                ; BIOS SCRATCH AREA.
                ;
5DB1            TRK:    DS   1          ;CURRENT TRACK NUMBER.
5DB2            SECT:   DS   1          ;CURRENT SECTOR NUMBER.
5DB3            DMAADD: DS   2          ;DISK TRANSFER ADDRESS.
                ;
                ; THE NEXT SEVERAL BYTES, BETWEEN STARTZ AND
                ; ENDZ, ARE SET TO ZERO AT COLD BOOT TIME.
                ;
                STARTZ:                 ;START OF ZEROED AREA.
5DB5            DISKNO: DS   1          ;DISK NUMBER (TO CP/M).
                        IF  TESTING
                ;
                ; ERROR COUNTS.  THESE LOCATIONS KEEP TRACK OF THE
                ; NUMBER OF ERRRS THAT OCCUR DURING READ, WRITE,
                ; OR SEEK OPERATIONS.  THEY ARE INITIALIZED ONLY
                ; WHEN A COLD-START IS PERFORMED BY THE BOOOTSTRAP.
                ;
                RECNT:  DS   1          ;READ ERROR COUNT.
                WECNT:  DS   1          ;WRITE ERROR COUNT.
                SECNT:  DS   1          ;SEEK ERROR COUNT.
                        ENDIF
                ;
                ; SPECIAL FLAGS.
                ;
5DB6            CONOTF: DS   1          ;NO-PRINT FLAG (WHEN FF).
5DB7            HLSF:   DS   1          ;HEAD-LOAD SELECT FLAG.

                        IF  LSTPAG
                LFCNT:  DS   1          ;PAGING LINE-FEED COUNT.
                        ENDIF
                ;
                ; TRTAB - DISK TRACK TABLE - PRESENT POSITION OF
                ;       HEADS FOR UP TO 4 DRIVES.
                ;
5DB8            TRTAB:  DS   4
5DBC            DRVFLG: DS   4          ;DRIVE FLAG BYTES FOR 4 DRIVES
5DC0            DENS:   DS   1          ;CURRENT DRIVE FLAG BYTE
                ;
                ; VDM SCRATCH AREA.
                ;
                ENDZ:                   ;END OF ZEROED AREA.
                        IF   VIDEO      ;IF VIDEO BOARD IN,
                VDMP:   DS   2          ;VIDEO CURSOR POSITION.
                        ENDIF

5DC1            NODSKS: DS   1          ;NUMBER OF DISKS.
5DC2            ERCNT:  DS   1          ;ERROR COUNT FOR RETRIES
```

```
5DC2            ERCNT:  DS    1           ;ERROR COUNT FOR RETRIES.
5DC3            SERCNT: DS    1           ;SEEK RETRY COUNTER.
5DC4            LATCH:  DS    1           ;NEW CODE FOR LATCH.
5DC5            CLATCH: DS    1           ;CURRENT CODE IN LATCH.
5DC6            DBUFF:  DS    128         ;DENSITY SELECT BUFFER
5E46                    END
```

B>TYPE 2ABIOS64.PRN 2.2 S.D. 64K - PRN

```
                ;-----------------------------------------------------
                ; CP/M BASIC INPUT/OUTPUT OPERATING SYSTEM (BIOS)
                ; TARBELL ELECTRONICS
                ; 2.X VERSION OF 11-4-80
                ; Copyright (c) 1980 Tarbell Electronics
                ;-----------------------------------------------------
                ; This bios module is the CPM V2.X Auto Select Bios.
                ; This bios reads single or double density disk.
                ; The Double density disk contains 51 sectors/track,
                ; 77 tracks. Track 0 = single density, Tracks 1 - 76
                ; are double density at 51 sectors per track.
                ; Note:  If you leave DMACNTL false, you must have a CPU
                ; which runs at 4 MHz to run double density.
                ; This bios now supports double sided single/double density.
                ; THIS SECTION DEFINES THE I/O PORTS AND STATUS BITS.
                ; BY SETTING THE PROPER VALUES FOR THE EQU STATEMENTS,
                ; THE I/O MAY BE AUTOMATICALLY RECONFIGURED TO FIT MOST
                ; SITUATIONS. THE TRUE AND FALSE ONES CONTROL CONDITIONAL
                ; ASSEMBLIES OF DIFFERENT SECTIONS OF I/O ROUTINES TO FIT
                ; DIFFERENT INTERFACE REQUIREMENTS.
                ;
FFFF =          TRUE    EQU  0FFFFH             ;DEFINE VALUE OF TRUE.
0000 =
B>



                ;-----------------------------------------------------
                ; CP/M BASIC INPUT/OUTPUT OPERATING SYSTEM (BIOS)
                ; TARBELL ELECTRONICS
                ; 2.X VERSION OF 11-4-80
                ; Copyright (c) 1980 Tarbell Electronics
                ;-----------------------------------------------------
                ; This bios module is the CPM V2.X Auto Select Bios.
                ; This bios reads single or double density disk.
                ; The Double density disk contains 51 sectors/track,
                ; 77 tracks. Track 0 = single density, Tracks 1 - 76
                ; are double density at 51 sectors per track.
                ; Note:  If you leave DMACNTL false, you must have a CPU
                ; which runs at 4 MHz to run double density.
                ; This bios now supports double sided single/double density.
                ; THIS SECTION DEFINES THE I/O PORTS AND STATUS BITS.
                ; BY SETTING THE PROPER VALUES FOR THE EQU STATEMENTS,
                ; THE I/O MAY BE AUTOMATICALLY RECONFIGURED TO FIT MOST
                ; SITUATIONS. THE TRUE AND FALSE ONES CONTROL CONDITIONAL
                ; ASSEMBLIES OF DIFFERENT SECTIONS OF I/O ROUTINES TO FIT
                ; DIFFERENT INTERFACE REQUIREMENTS.
                ;
FFFF =          TRUE    EQU  0FFFFH             ;DEFINE VALUE OF TRUE.
0000 =          FALSE   EQU  NOT TRUE           ;DEFINE VALUE OF FALSE.
                ;
                ;*****************************************************
                ;*** THIS BEGINS THE AREA WHICH REQUIRES CHANGES ***
                ;***       FOR DIFFERENT CONSOLE I/O SYSTEMS       ***
                ;*****************************************************
                ;
0040 =          MSIZE   EQU  64         ;MEMORY SIZE IN KBYTES.
0000 =          INTRP   EQU  FALSE      ;TRUE IF INTERRUPTS ALLOWED.
0000 =          TARBELL EQU  FALSE      ;TRUE IF USING THE TARBELL Z-80 CPU.
0000 =          IOBASE  EQU  0          ;BASE IO ADDR FOR TARBELL CPU (0 or 10
0000 =          TIMER   EQU  FALSE      ;TRUE IF USING CPU TIMER (Tarbell CPU
0000 =          STD     EQU  FALSE      ;TRUE IF STANDARD I/O.
```

```
FFFF =          MSIO2   EQU  TRUE          ;TRUE IF MITS 2SIO.
0000 =          VDB8024 EQU  FALSE         ;TRUE IF USING VDB-8024 BOARD.
0000 =          DELTA   EQU  FALSE         ;TRUE IF USING DELTA PRODUCTS CPU.
0000 =          ISIO2   EQU  FALSE         ;TRUE IF IMSAI SIO-2.
0000 =          TUART   EQU  FALSE         ;TRUE IF CROMEMCO TUART.
0000 =          VIDEO   EQU  FALSE         ;TRUE IF USING A MEMORY MAPPED VIDEO.
0000 =          OTHER   EQU  FALSE         ;TRUE IF SOMETHING ELSE.
0000 =          SOLOS   EQU  FALSE         ;TRUE IF PROC TECH SOLOS.
0000 =          DUBSID  EQU  FALSE         ;TRUE FOR DOUBLE SIDED DRIVES (1 logica
FFFF =          DMACNTL EQU  TRUE          ;TRUE IF USING DMA CONTROL.
0004 =          NDISK   EQU  4             ;DEFINES THE NUMBER DRIVES IN SYSTEM.
                ;
                        IF  VIDEO          ;IF USING A VIDEO BOARD
                OUTADDR EQU  0             ;PUT OUTPUT ADDRESS HERE
                        ENDIF
                ;
                        IF NOT SOLOS AND NOT TARDEL     ;IF NOT PROC TECH SOLO
0000 =          CSTAT   EQU  0             ;CONSOLE STATUS PORT.
0000 =          CCOM    EQU  0             ;CONSOLE COMMAND PORT.
0001 =          CDATA   EQU  1             ;CONSOLE DATA PORT.
0002 =          LSTAT   EQU  2             ;LIST STATUS PORT.
0002 =          LCOM    EQU  2             ;LIST COMMAND PORT.
0003 =          LDATA   EQU  3             ;LIST DATA PORT.
                        ENDIF
                ;
0000 =          CONUL   EQU  FALSE         ;CONSOLE NULLS?
0010 =          CNULL   EQU  16            ;CONSOLE NULL COUNT.
0000 =          LSTNUL  EQU  FALSE         ;LIST DEVICE NULLS?
0000 =          LNULL   EQU  0             ;LIST NULL COUNT.
0000 =          LSTPAG  EQU  FALSE         ;LIST DEVICE PAGING?
0042 =          LINCNT  EQU  66            ;LINES PER PAGE.
0008 =          HLAB    EQU  8             ;8 FOR HD LD AT BEG OF SEEK.
0001 =          STPRAT  EQU  1             ;RATE 0=3ms,1=6MS, 2=10MS, 3=20MS.
0000 =          DUAL    EQU  FALSE         ;TRUE IF DUAL HEADED (2 HEADS MOVING TO
                ;
                ;*************************************************************
                ;*** THIS IS THE END OF THE AREA WHICH NORMALLY NEED ***
                ;***      BE CHANGED FOR MOST CONSOLE I/O SYSTEMS      ***
                ;*************************************************************
                ;
0000 =          RDYLO   EQU  STD OR SOLOS OR OTHER  ;STATUS READY WHEN LOW.
FFFF =          RDYHI   EQU  NOT RDYLO
0000 =          TARDEL  EQU  TARBELL OR DELTA   ;IF USING TARBELL OR DELTA CPU.
                ;
                        IF   TARDEL        ;IF USING TARBELL OR DELTA CPU
                CCOM    EQU  IOBASE+1      ;CONSOLE COMMAND PORT
                CSTAT   EQU  IOBASE+1      ;CONSOLE STATUS PORT ( CHAN A.)
                CDATA   EQU  IOBASE+0      ;CONSOLE DATA PORT
                LCOM    EQU  IOBASE+3      ;LIST COMMAND PORT
                LSTAT   EQU  IOBASE+3      ;LIST STATUS PORT (CHAN B.)
                LDATA   EQU  IOBASE+2      ;LIST DATA PORT
                        ENDIF
                ;
                        IF      TIMER AND TARBELL       ;MUST BE USING TARBELL
                ;
                ; TIMER EQUATES
                ;
                TCH0    EQU     IOBASE+4           ;TIMER CHAN 0 ADDRESS
                TCH1    EQU     IOBASE+5           ;TIMER CHAN 1 ADDRESS
                TCH2    EQU     IOBASE+6           ;TIMER CHAN 2 ADDRESS
                TCMND   EQU     IOBASE+7           ;TIMER COMMAND PORT
                IMASK   EQU     IOBASE+8           ;INTERRUPT MASKING PORT
                CNTR0   EQU     00000000B          ;counter 0
                CNTR1   EQU     01000000B          ;counter 1
                CNTR2   EQU     10000000B          ;counter 2
                RLWORD  EQU     00110000B          ;read/load lsb 1st, msb 2nd.
```

```
                RLHBYTE EQU     00100000B       ;read/load msb only.
                RLLBYTE EQU     00010000B       ;read/load lsb only.
                CNTRLT  EQU     00000000B       ;counter latching operation.
                BINARY  EQU     00000000B       ;select binary operation.
                BCD     EQU     00000001B       ;select BCD operation.
                MODE0   EQU     00000000B       ;interrupt on terminal count.
                MODE1   EQU     00000010B       ;programmable One-shot.
                MODE2   EQU     00000100B       ;rate generator.
                MODE3   EQU     00000110B       ;square wave rate generator.
                MODE4   EQU     00001000B       ;software triggered strobe.
                MODE5   EQU     00001010B       ;hardware triggered strobe.
                        ENDIF
                ;
                        IF      SOLOS           ;IF PROC TECH SOLOS,
                CSTAT   EQU     0FAH            ;CONSOLE STATUS PORT.
                KBD     EQU     0C02EH          ;SOLOS KEYBOARD.
                CLRSCR  EQU     0C0D5H          ;CLEAR SCREEN.
                SCRN    EQU     0C054H          ;SOLOS OUTPUT.
                        ENDIF
                ;
                        IF NOT SOLOS            ;IF NOT PROC TECH SOLOS,
00E0 =          DMAP    EQU     0E0H            ;DMA BASE ADDRESS.
00F8 =          DISK    EQU     0F8H            ;DISK BASE ADDRESS.
                        ENDIF
                ;
                        IF SOLOS                ;IF PROC TECH SOLOS,
                DMAP    EQU     060H            ;DMA BASE ADDRESS.
                DISK    EQU     078H            ;DIFFERENT DISK PORTS.
                        ENDIF
                ;
00E0 =          ADR0    EQU     DMAP+0          ;DMA ADDRESS REG PORT.
00E1 =          WCT0    EQU     DMAP+1          ;DMA WORD COUNT REG PORT.
00E8 =          CMND    EQU     DMAP+8          ;DMA COMMAND PORT.
00F8 =          DCOM    EQU     DISK            ;DISK COMMAND PORT.
00F8 =          DSTAT   EQU     DISK            ;DISK STATUS PORT.
00F9 =          TRACK   EQU     DISK+1          ;DISK TRACK PORT.
00FA =          SECTP   EQU     DISK+2          ;DISK SECTOR PORT.
00FB =          DDATA   EQU     DISK+3          ;DISK DATA PORT.
00FC =          WAIT    EQU     DISK+4          ;DISK WAIT PORT.
00FC =          DCONT   EQU     DISK+4          ;DISK CONTROL PORT.
00FD =          DMACHK  EQU     DISK+5          ;DMA CHECK PORT.
000A =          RTCNT   EQU     10              ;RETRY COUNT.
                ;
                        IF      STD             ;IF STANDARD I/O,
                CKBR    EQU     00000001B       ;KEYBOARD READY BIT.
                CPTR    EQU     10000000B       ;CONS OUTPUT RDY BIT.
                        ENDIF
                ;
                        IF      MSIO2           ;IF MITS 2SIO,
0001 =          CKBR    EQU     00000001B       ;KEYBOARD READY BIT.
0002 =          CPTR    EQU     00000010B       ;PRINT READY BIT.
                        ENDIF
                ;
                        IF      VDB8024         ;IF VDB-8024 BOARD.
                CKBR    EQU     00000010B       ;KEYBOARD READY BIT.
                CPTR    EQU     00000100B       ;CONS OUTPUT RDY BIT.
                        ENDIF
                ;
                        IF      ISIO2
                CKBR    EQU     00000010B       ;KEYBOARD READY BIT.
                CPTR    EQU     00000001B       ;PRINT READY BIT.
                        ENDIF
                ;
                        IF      TARDEL
                CKBR    EQU     00000010B       ;KEYBOARD READY BIT.
                CPTR    EQU     00000001B       ;PRINT READY BIT.
```

```
                        ENDIF
                ;
                        IF    TUART                 ;IF CROMEMCO TUART,
                CKBR    EQU   01000000B             ;KEYBOARD READY BIT.
                CPTR    EQU   10000000B             ;PRINT READY BIT.
                        ENDIF
                ;
                        IF    SOLOS                 ;IF PROC TECH SOLOS,
                CKBR    EQU   00000001B             ;KEYBOARD READY BIT.
                CPTR    EQU   10000000B             ;DUMMY EQU.
                        ENDIF
                ;
                        IF    OTHER                 ;IF SOMETHING ELSE,
                CKBR    EQU   00000010B             ;KEYBOARD READY BIT.
                CPTR    EQU   10000000B             ;PRINTER READY BIT.
                        ENDIF
                ;
0002 =          LRBIT   EQU   CPTR                  ;LISTER READY BIT.
                ;
0003 =          IOBYTE  EQU   3                     ;ADDRESS OF I/O BYTE.
B000 =          CBASE   EQU   (MSIZE-20)*1024       ;BIAS FOR LARGER THAN 20K.
E400 =          CPMB    EQU   CBASE+3400H           ;START OF CPM 2.0
EC06 =          BDOS    EQU   CPMB+806H             ;START OF BDOS 2.0.
FA00 =          BIOS    EQU   CPMB+1600H            ;START OF CBIOS IO.
0004 =          CDISK   EQU   4                     ;LOCATION 4 IS CURRENT DISK.
0011 =          NSECTS  EQU   17                    ;NUMBER OF SECTORS IN IT.
                ;
E408                    ORG   CPMB+8
                ;
E408 0D0A546172SMSG:    DB    0DH,0AH,'Tarbell '
E412 3634               DB    MSIZE/10+'0',MSIZE MOD 10+'0'
E414 4B2043504D         DB    'K CPM 2.2',0DH,0AH
E41F 4175746F2D         DB    'Auto-Select '
                        IF    DUBSID
                        DB    'Double Sided '
                        ENDIF
E42B 766572206F         DB    'ver of 11-4-80',0
                ;
                ; BOOT
                ; THIS SECTION IS EXECUTED WHENEVER RESET AND RUN
                ; IS PUSHED, AFTER THE COLDSTART LOADER READS IN
                ; THE CPM SYSTEM.
                ;
E43A 318000     BOOT:   LXI   SP,80H                ;SET STACK POINTER.
                ;
                        IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
                        EI                          ;ENABLE THEM HERE.
                        ENDIF
                ;
                        IF    MSIO2                 ;IF MITS 2SIO,
E43D 3E03               MVI   A,3                   ;INITIALIZE 2SIO.
E43F D300               OUT   CCOM
E441 D302               OUT   LCOM
E443 3E11               MVI   A,11H
E445 D300               OUT   CCOM
E447 D302               OUT   LCOM
                        ENDIF
                ;
                        IF    TARDEL OR ISIO2
                        LXI   H,IOINIT              ;point to 8251 init. bytes
                        MVI   B,4                   ;there are 4 of them
                INITIO: MOV   A,M                   ;set a byte
                        OUT   CCOM                  ;out to command port of console
                        OUT   LCOM                  ;out to command port of lister
                        INX   H                     ;bump pointer
                        DCR   B                     ;decrease count
```

```
                        JNZ   INITIO            ;loop till done.
                        ENDIF
                ;
                        IF    TUART             ;IF CROMEMCO TUART,
                        MVI   A,1               ;SET A = 1.
                        OUT   54H               ;SELECT DEVICE A.
                        OUT   52H               ;RESET DEVICE B.
                        LXI   H,BAUDRS          ;GET ADR OF BAUD RATE TABLE.
                        MVI   A,11H             ;OCTUPLE THE CLOCK.
                IT1:    OUT   02H               ;& RESET CURRENT DEV.
                        MOV   A,M               ;GET BAUD RATE FROM TABLE.
                        INX   H                 ;INCREMENT POINTER.
                        OUT   0                 ;SET BAUD RATE.
                        CALL  CONIN             ;READ KEYBOARD.
                        CALL  CONIN             ;READ KEYBOARD AGAIN.
                        CPI   0DH               ;IF NOT CARRIAGE-RETURN,
                        MVI   A,1               ;SLOW THE CLOCK.
                        JNZ   IT1               ;UNTIL A CARRIAGE-RETURN.
                        ENDIF
                ;
                        IF    SOLOS             ;IF PROC TECH SOLOS,
                        CALL  CLRSCR            ;CLEAR SCREEN.
                        ENDIF
                ;
                        IF DMACNTL
E449 21B4FA             LXI   H,RWDMA           ;POINT TO DMA ROUTINE
E44C 2201FA             SHLD  DMAENT+1          ;MODIFY BOOT JMP ADDRESS.
                        ENDIF
                ;
                        IF    TIMER AND TARBELL ;IF USING TARBELL CPU
                        MVI   A,CNTR0+RLWORD+MODE2+BINARY       ;INIT 8253
                        OUT   TCMND             ;SEND IT TO COMMAND PORT
                        LXI   B,33333           ;TIME CONSTANT FOR 60 HZ
                        MOV   A,C
                        OUT   TCH0              ;LS BYTE OF COUNT
                        MOV   A,B
                        OUT   TCH0              ;MS BYTE OF COUNT
                        ENDIF
                ;
E44F C3C0F9             JMP   BOOTF             ;FINISH BOOT
                ;
                        IF    TARDEL OR ISIO2
                IOINIT: DB    0AAH,040H,0CEH,037H
                        ENDIF
                ;
                        IF    TUART             ;IF CROMEMCO TUART,
                BAUDRS: DB    94H,0CEH,0A2H,92H,88H,84H,82H,1
                        ENDIF
                ;
F9C0                    ORG   BIOS-64           ;HIDE REST OF BOOT HERE.
F9C0 AF         BOOTF:  XRA   A                 ;CLEAR SCRATCH AREA.
F9C1 320300             STA   IOBYTE            ;CLEAR I/O BYTE.
F9C4 320400             STA   CDISK             ;SELECT DRIVE ZERO
F9C7 0611               MVI   B,ENDZ-STARTZ     ;GET LENGTH OF ZERO AREA.
F9C9 2127FD             LXI   H,STARTZ          ;GET SCRATCH ADDRESS.
F9CC 77         BOOTL:  MOV   M,A               ;PUT ZERO IN MEMORY.
F9CD 23                 INX   H                 ;INCREMENT POINTER.
F9CE 05                 DCR   B                 ;DECREMENT COUNTER.
F9CF C2CCF9             JNZ   BOOTL             ;LOOP TILL DONE.
F9D2 DB01               IN    CDATA             ;CLEAR CONSOLE STATUS.
F9D4 2103E4             LXI   H,SMSG            ;POINT TO SIGN ON.
F9D7 7E         PMSG:   MOV   A,M               ;GET A BYTE OF THE MESSAGE
F9D8 23                 INX   H                 ;BUMP MEMORY POINTER.
F9D9 B7                 ORA   A                 ;IS IT A ZERO?
F9DA CA1DFB             JZ    GOCPM             ;YES, WE ARE DONE, JMP TO CPM.
F9DD 4F                 MOV   C,A               ;NOPE, PRINT MORE MESSAGE.
```

```
F9DE CD5CFB             CALL  CONOT             ;USE THE CONOT ROUTINE.
F9E1 C3D7F9             JMP   PMSG              ;AND LOOP TILL DONE.
                ;
FA00                    ORG   BIOS              ;START OF CBIOS STRUCTURE.
                ;
                ; I/O JUMP VECTOR
                ; THIS IS WHERE CPM CALLS WHENEVER IT NEEDS TO DO ANY INPUT/OUTI
                ; OPERATION. USER PROGRAMS MAY USE THESE ENTRY POINTS ALSO, BU
                ; THAT THE LOCATION OF THIS VECTOR CHANGES WITH THE MEMORY SIZE.
                ;
FA00 C33AE4     DMAENT: JMP   BOOT              ;FROM SBOOT LOADER,CHANGED FOR
FA03 C3EFFA     WBOOTE: JMP   WBOOT             ;FROM WARM BOOT.
FA06 C344FB             JMP   CONST             ;CHECK CONSOLE KB STATUS.
FA09 C351FB             JMP   CONIN             ;READ CONSOLE CHARACTER.
FA0C C35CFB             JMP   CONOT             ;WRITE CONSOLE CHARACTER.
FA0F C319FD             JMP   LIST              ;WRITE LISTING CHAR.
FA12 C313FD             JMP   PUNCH             ;WRITE PUNCH CHAR.
FA15 C313FD             JMP   READER            ;READ READER CHAR.
FA18 C3F4FB             JMP   HOME              ;MOVE DISK TO TRACK ZERO.
FA1B C367FB             JMP   SELDSK            ;SELECT DISK DRIVE.
FA1E C3FAFB             JMP   SETTRK            ;SEEK TO TRACK IN REG A.
FA21 C363FC             JMP   SETSEC            ;SET SECTOR NUMBER.
FA24 C378FC             JMP   SETDMA            ;SET DISK STARTING ADR.
FA27 C39DFC             JMP   READ              ;READ SELECTED SECTOR.
FA2A C3F2FC             JMP   WRITE             ;WRITE SELECTED SECTOR.
FA2D C30CFD             JMP   PRSTAT            ;LIST STATUS CHECK.
FA30 C36DFC             JMP   SECTRAN           ;SECTOR TRANSLATE ROUTINE.
                ;
                ; THIS SECTION DEFINES THE THE DISK PARAMETERS
                ;
FA33 =          DPBASE  EQU   $                 ;BASE OF DISK PARAMETER BLOCK
FA33 97FA0000   DPE0:   DW    XLT0,0000H        ;TRANSLATE TABLE
FA37 00000000           DW    0000H,0000H       ;SCRATCH AREA
FA3B 38FD76FA           DW    DIRBUF,SDTAB+3    ;DIR BUFF, PARM BLOCK
FA3F D7FDB8FD           DW    CSV0,ALV0         ;CHECK, ALLOC VECTORS
                ;
FA43 97FA0000   DPE1:   DW    XLT1,0000H
FA47 00000000           DW    0000H,0000H
FA4B 38FD76FA           DW    DIRBUF,DPB1
FA4F 0EFEEFFD           DW    CSV1,ALV1
                ;
FA53 97FA0000   DPE2:   DW    XLT2,0000H
FA57 00000000           DW    0000H,0000H
FA5B 38FD76FA           DW    DIRBUF,DPB2
FA5F 45FE26FE           DW    CSV2,ALV2
                ;
FA63 97FA0000   DPE3:   DW    XLT3,0000H
FA67 00000000           DW    0000H,0000H
FA6B 38FD76FA           DW    DIRBUF,DPB3
FA6F 7CFE5DFE           DW    CSV3,ALV3
                ;
                ;THE FOLLOWING DESCRIBES THE DISK PHYSICAL NATURE, SUCH AS
                ;SECTORS/TRACK,DIRECTORY SIZE, ETC...
                ;THE FOLLOWING TABLE DEFINES A SINGLE DENSITY DRIVE.
                ;
FA73 =          SDTAB:  EQU   $                 ;ONE OF 4 DISK PARM. BLOCKS
FA73 00                 DB    00H               ;LOG BYTE SINGLE DENSITY
FA74 97FA               DW    XLT0              ;USE SINGLE DENSITY TRANSLATE T
FA76 1A00               DW    26                ;SECTORS/TRACK
FA78 03                 DB    3                 ;BLOCK SHIFT
FA79 07                 DB    7                 ;BLOCK MASK
FA7A 00                 DB    0                 ;EXTNT MASK
FA7B F200               DW    242               ;DISK SIZE - 1
FA7D 3F00               DW    63                ;DIRECTORY MAX.
FA7F C0                 DB    192               ;ALLOC0
FA80 00                 DB    0                 ;ALLOC1
```

```
FA81 1000                       DW    16                ;CHECK SIZE
FA83 0200                       DW    2                 ;NUMBER OF SYSTEM TRACKS
                        ;
                                IF    DUBSIDE           ;is using double sided drives.
                        ;
                        ; Defines a Single density/ Double sided disk
                        ;
                                DB    02H               ;log byte doub sided
                                DW    XLT0
                                DW    26
                                DB    4
                                DB    15
                                DB    0
                                DW    242
                                DW    95                ;allow 95 entrys for dir.
                                DB    192
                                DB    0
                                DW    24
                                DW    2
                                ENDIF
                        ;
                        ;THE FOLLOWING TABLE DEFINES A DOUBLE DENSITY DRIVE.
                        ;
FA85 01                 DDTAB:  DB    01H               ;log byte doub den/sing sided
FA86 0000                       DW    0                 ;NO SECTOR TRANSLATE TABLE.
FA88 3300                       DW    51                ;51 SECTORS.
FA8A 04                         DB    4                 ;BLOCK SHIFT.
FA8B 0F                         DB    15                ;BLOCK MASK.
FA8C 00                         DB    0                 ;EXTENT MASK.
FA8D ED00                       DW    237               ;DISK SIZE -1
FA8F 5F00                       DW    95                ;DIRECTORY MAX.
FA91 C0                         DB    192               ;ALLOC0
FA92 00                         DB    0                 ;ALLOC1
FA93 1800                       DW    24                ;CHECK SIZE
FA95 0200                       DW    2                 ;NUMBER OF SYSTEM TRACKS.
                        ;
                                IF    DUBSIDE           ;if using double sided drives.
                        ;
                        ; Defines a Double density/Doub sided drive
                        ;
                                DB    03H               ;log byte and dub sided
                                DW    0
                                DW    51
                                DB    5
                                DB    31
                                DB    0
                                DW    237
                                DW    95
                                DB    192
                                DB    0
                                DW    24
                                DW    2
                                ENDIF
                        ;
                        ;SECTOR TRANSLATION TABLE
                        ;
FA97 =                  XLT0    EQU   $                 ;START OF TRANS. TABLE
FA97 0107 0D13 19               DB    1,7,13,19,25
FA9C 050B 1117 03               DB    5,11,17,23,3
FAA1 090F 1502 08               DB    9,15,21,2,8
FAA6 0E14 1A06 0C               DB    14,20,26,6,12
FAAB 1218 040A 10               DB    18,24,4,10,16,22
                        ;
FA76 =                  DPB1    EQU   SDTAB+3           ;EQUIVALENT PARAMETERS
FA97 =                  XLT1    EQU   XLT0              ;SAME TRANSLATE TABLE
                        ;
```

```
FA76 =          DPB2    EQU  SDTAB+3
FA97 =          XLT2    EQU  XLT0
                ;
FA76 =          DPB3    EQU  SDTAD+3
FA97 =          XLT3    EQU  XLT0
                ;
                ;DISK SET UP ROUTINE. THIS ROUTINE IS COMMON TO BOTH THE
                ;READ AND WRITE ROUTINES FOR DMA OPERATION. THIS ROUTINE
                ;MAY BE USED STAND ALONE BY PASSING PARAMETERS TO IT AND
                ;JUMPING TO WBOOT-3 HEX. THIS JUMP VECTOR IS CHANGED WHEN
                ;CP/M IS BOOTED UP.
                ;
                ;ENTERY POINT = RWDMA:
                ;
                ;USER MUST SET UP DMAADD FOR MEMORY ADDRESS AND
                ;USER MUST SET UP DISK SECTOR WITH 'SETSEC' ENTRY
                ;THE TRACK TO READ OR WRITE MUST BE SET UP USING 'SETTRK'
                ;BEFORE USING RWDMA ROUTINE EXTERNALLY.
                ;
                ;ENTRY PARAMETERS:
                ;B    = FLOPPY DISK (1793) READ/WRITE COMMAND BYTE
                ;C    = FLOPPY DISK (1793) FORCE INTERRUPT COMMAND BYTE
                ;D    = DMA (8257) READ/WRITE COMMAND + HIGH BYTE COUNT
                ;E    = DMA (8257) LOW BYTE COUNT (80 HEX = 128 BYTES)
                ;
                ;EXIT VALUES
                ;B,C = FLOPPY COMMANDS
                ;D,E = DMA COMMAND + BYTE COUNT.
                ;H,L = (H,L + D,E)
                ;A    = FLOPPY DISK STATUS BYTE
                ;
                ;STACK USAGE IS 1 LEVEL DEEP.
                ;
                        IF DMACNTL              ;IF USING DMA CONTROL
                ;
FAB1 3234FD     DMARW:  STA  ERCNT              ;SAVE ERROR COUNT.
FAB4 3A24FD     RWDMA:  LDA  SECT               ;GET SECTOR TO READ/WRITE
FAB7 D3FA               OUT  SECTP              ;AND SEND IT FLOPPY CHIP.
FAB9 2A25FD             LHLD DMAADD             ;GET CPM DMA ADDRESS.
FABC AF         DMARWE: XRA  A                  ;CLEAR ACCUM.
FABD D3E8               OUT  CMND               ;RESET DMA CHIP.
FABF 79                 MOV  A,C                ;FORCE INTERRUPT COMMAND BYTE
FAC0 D3F8               OUT  DCOM               ;SEND IT TO CONTROLLER.
FAC2 7B                 MOV  A,E                ;BYTE COUNT TO TRANSFER
FAC3 3D                 DCR  A                  ;COUNT = COUNT - 1.
FAC4 D3C1               OUT  WCT0               ;SEND IT TO DMA CHIP.
FAC6 7A                 MOV  A,D                ;GET READ/WRITE CODE.
FAC7 D3E1               OUT  WCT0               ;AND TELL DMA CHIP WHAT TO DO.
FAC9 7D                 MOV  A,L                ;GET LOW ADDRESS BYTE
FACA D3E0               OUT  ADR0               ;AND SEND IT TO DMA CHIP.
FACC 7C                 MOV  A,H                ;GET HIGH ADDRESS BYTE
FACD D3E0               OUT  ADR0               ;AND SEND IT TO DMA CHIP.
FACF 3E41               MVI  A,41H              ;SET UP FOR REQUEST CH. 0
FAD1 D3E8               OUT  CMND               ;SEND IT TO CONTROLLER.
FAD3 CD7EFC             CALL HDLD               ;CHECK HEAD LOAD BIT.
FAD6 B0                 ORA  B                  ;'OR' IN THE READ/WRITE BITS.
FAD7 D3F8               OUT  DCOM               ;TELL FLOPPY CHIP WHAT TO DO.
                ;
                ;ADJUST H,L FOR 128 BYTE INCREASE.
                ;
FAD9 D5                 PUSH D                  ;SAVE D,E
FADA 7A                 MOV  A,D                ;GET DMA COMMAND BYTE
FADB E63F               ANI  3FH                ;STRIP OFF COMMAND BITS 7 & 6
FADD 57                 MOV  D,A                ;NOW SET FOR H,L ADJUST.
FADE 19                 DAD  D                  ;ADD IT TO H,L.
FADF D1                 POP  D                  ;RESTORE D,E
```

```
                        ;
                        ;GENERAL PURPOSE WAIT ROUTINE.
                        ;
FAE0 3E20               CNTLOOP:MVI   A,20H              ;COUNT VALUE.
FAE2 3D                 SLOOP:  DCR   A
FAE3 C2E2FA                     JNZ   CNTLOOP           ;LOOP TILL = ZERO.
FAE6 DBFD                       IN    DMACHK            ;CHECK FOR OPERATION DONE.
FAE8 07                         RLC                     ;BY LOOKING AT BIT 7.
FAE9 DAE6FA                     JC    SLOOP             ;LOOP TILL BIT 7 = 0.
FAEC DBF8                       IN    DSTAT             ;CHECK AND RETURN DISK STATUS.
FAEE C9                         RET                     ;RETURN TO CALLER.
                                ENDIF
                        ;
                        ;Warm-boot - Read the CCP back into memory. BDOS and BIOS
                        ;assumed still in memory. If they are not, a cold start will
                        ;have to be done to bring them back into memory.
                        ;
FAEF 318000             WBOOT:  LXI   SP,80H            ;SET STACK POINTER.
                        ;
                                IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
                                EI                      ;ALLOW THEM HERE.
                                ENDIF
                        ;
                                IF    LSTPAG            ;IF LIST DEVICE PAGING,
                                XRA   A                 ;RESET LINE-FEED COUNT.
                                STA   LFCNT
                                ENDIF
                        ;
FAF2 0E00                       MVI   C,0               ;SELECT DISK 0.
FAF4 CD67FB                     CALL  SELDSK
FAF7 CDF4FB                     CALL  HOME              ;MOVE TO TRACK ZERO.
FAFA 210000                     LXI   H,0               ;clear h,l
FAFD 222FFD                     SHLD  DRVFLG            ;clear drive flags
FB00 2231FD                     SHLD  DRVFLG+2
FB03 0611                       MVI   B,NSECTS          ;GET # SECTORS FOR CPM READ.
FB05 0E02                       MVI   C,2               ;TRACK (B)=0, SECTOR (C)=2.
                        ;
                                IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
                                DI                      ;DISABLE THEM HERE.
                                ENDIF
                        ;
FB07 2100E4             RBLK1:  LXI   H,CPMB            ;GET STARTING ADDRESS.
FB0A 2225FD                     SHLD  DMAADD            ;SET STARTING ADDRESS.
FB0D CD68FC                     CALL  SETSEC            ;READ STARTING AT SECTOR IN C
FB10 C5                         PUSH  B
FB11 CD9DFC                     CALL  READ              ;READ A SECTOR BACK.
FB14 C1                         POP   B
FB15 C23EFB                     JNZ   RDERR             ;IF ERROR, PRINT MESSAGE.
FB18 0C                         INR   C                 ;INCREMENT SECTOR NUMBER.
FB19 05                         DCR   B                 ;DECREMENT SECTOR COUNT.
FB1A C20AFB                     JNZ   RBLK1             ;NOT ZERO, KEEP READING
                        ;
                                IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
                                EI                      ;ALLOW THEM AGAIN HERE.
                                ENDIF
                        ;
                        ; SET UP JUMPS INTO CP/M IN LOWER MEMORY.
                        ;
FB1D 3EC3               GOCPM:  MVI   A,0C3H            ;PUT JMP TO WBOOT
FB1F 320000                     STA   0                 ;ADR AT ZERO.
FB22 2103FA                     LXI   H,WBOOTE          ;WARMBOOT ENTRY POINT
FB25 220100                     SHLD  1                 ;SET IT.
FB28 320500                     STA   5                 ;SET JUMP INSTRUCTION.
FB2B 2106EC                     LXI   H,BDOS            ;PUT JUMP TO BDOS
FB2E 220600                     SHLD  6                 ;AT ADR 5,6,7.
FB31 218000                     LXI   H,80H             ;SET DEFAULT DMA ADR.
```

```
FB34 2225FD             SHLD DMAADD             ;SAVE IT.
FB37 3A0400             LDA  CDISK              ;GET DISK NUMBER TO
FB3A 4F                 MOV  C,A                ;PASS TO CCP IN C.
FB3B C300E4             JMP  CPMB               ;JUMP TO CCP.
                ;
FB3E CDB7FC     RDERR:  CALL RECOV              ;We have an error in booting.
FB41 C3E1FA             JMP  WBOOT              ;DO A WARM BOOT.
                ;
                ; CHECK CONSOLE INPUT STATUS.
                ;
FB44 CD4CFB     CONST:  CALL STATCON            ;CHECK CONSOLE STATUS PORT.
FB47 3E00       CONST1: MVI  A,0                ;SET A=0 FOR RETURN.
                ;
                        IF   RDYLO              ;IF STATUS READY LOW,
                        RNZ                     ;NOT READY WHEN NOT 0.
                        ENDIF
                ;
                        IF   RDYHI              ;IF STATUS READY HIGH,
FB49 C8                 RZ                      ;NOT READY WHEN ZERO.
                        ENDIF
                ;
FB4A 2F                 CMA                     ;IF READY A=FF.
FB4B C9                 RET                     ;RETURN FROM CONST.
                ;
                ; STATCON - CHECK KEYBOARD STATUS
                ;
                        IF   NOT SOLOS
FB4C DB00       STATCON:IN   CSTAT              ;IN STATUS PORT
FB4E E601               ANI  CKBR               ;MASK READY BIT.
FB50 C9                 RET
                        ENDIF
                ;
                ; READ A CHARACTER FROM CONSOLE.
                ;
                CONIN:
                        IF   NOT SOLOS          ;IF NOT PROC TECH SOLOS,
FB51 CD4CFB             CALL STATCON            ;READ CONSOLE STATUS.
                        ENDIF
                ;
                        IF   SOLOS              ;IF PROC TECH SOLOS,
                        CALL KBD                ;READ SOL KEYBOARD.
                        JZ   CONIN              ;READY WHEN NOT ZERO.
                        ENDIF
                ;
                        IF   RDYLO AND NOT SOLOS
                        JNZ  CONIN              ;LOOP UNTIL LOW.
                        ENDIF
                ;
                        IF   RDYHI              ;IF READY WHEN HIGH,
FB54 CA51FB             JZ   CONIN              ;LOOP UNTIL HIGH.
                        ENDIF
                ;
                        IF   NOT SOLOS          ;IF NOT PROC TECH SOLOS,
FB57 DB01               IN   CDATA              ;READ A CHARACTER.
                        ENDIF
                ;
FB59 E67F               ANI  7FH                ;MAKE MOST SIG. BIT = 0.
FB5B C9                 RET                     ;RETURN FROM CONIN.
                ;
                ; WRITE A CHARACTER TO THE CONSOLE DEVICE.
                ;
                CONOT:
                        IF CONUL                ;IF NULLS REQUIRED,
                        MVI  A,0AH              ;IF IT'S A LF,
                        CMP  C                  ;THEN HOP OUT
                        JZ   CONULL             ;TO NULL ROUTINE.
```

```
                        ENDIF
                CONOT1:
                        IF NOT SOLOS AND NOT VIDEO
FB5C DB00               IN   CSTAT              ;READ CONSOLE STATUS.
FB5E E602               ANI  CPTR               ;IF NOT READY,
                        ENDIF
                ;
                        IF RDYLO AND NOT SOLOS AND NOT VIDEO
                        JNZ  CONOT1             ;LOOP UNTIL LOW.
                        ENDIF
                ;
                        IF RDYHI                ;IF READY WHEN HIGH,
FB60 CA5CFB             JZ   CONOT1             ;LOOP UNTIL HIGH.
                        ENDIF
                ;
                        IF NOT SOLOS AND NOT VIDEO
FB63 79                 MOV  A,C                ;GET CHARACTER.
FB64 D301               OUT  CDATA              ;PRINT IT.
FB66 C9                 RET                     ;RETURN.
                        ENDIF
                ;
                ;THIS ROUTINE CALLES YOUR VIDEO DRIVER ROUTINE WHICH MUST
                ;BE IN ROM. ALL REGISTERS MUST BE SAVED AND RESTORED BY YOUR
                ;VIDEO DRIVER IN ORDER TO BE COMPATIABLE WITH CPM. CPM PASSES
                ;THE CHAR. TO BE OUTPUT IN THE  C REGISTER. MAKE ANY CHANGES
                ;IN THIS ROUTINE TO PASS THE CHAR FROM REG C TO THE REGISTER
                ;YOUR VIDEO DRIVER EXPECTS IT TO BE IN.
                ;
                        IF  VIDEO               ;IF USING A VIDEO DRIVER IN RO
                        MOV  A,C                ;GET THE CPM CHAR INTO REG A
                        CALL OUTADDR            ;CALL YOUR VIDEO DRIVER.
                        RET                     ;RETURN TO CPM.
                        ENDIF
                ;
                        IF CONUL
                CONULL: PUSH B                  ;SAVE B&C.
                        MVI  B,CNULL+1          ;GET NULL COUNT.
                CONUL1: CALL CONOT1             ;PRINT CR.
                        MVI  C,0                ;GET NULL CHAR.
                        DCR  B                  ;DECREMENT COUNTER.
                        JNZ  CONUL1             ;DO NEXT NULL.
                        POP  B                  ;RESTORE B&C.
                        MOV  A,C                ;RESTORE A.
                        RET                     ;RETURN.
                        ENDIF
                ;
                        IF   SOLOS              ;IF PROC TECH SOLOS,
                        PUSH B                  ;SAVE B&C.
                        MOV  B,C                ;PUT CHAR IN B REG.
                        CALL SCRN               ;OUTPUT CHAR TO SOLOS.
                        POP  B                  ;RESTORE B&C.
                        MOV  A,C                ;PUT CHAR IN A.
                        RET                     ;RETURN FROM CONOT.
                        ENDIF
                ;
                ; SELECT DISK NUMBER ACCORDING TO REGISTER C.
                ;
FB67 210000     SELDSK: LXI  H,0                ;SET UP FOR ERROR CODE
FB6A 79                 MOV  A,C                ;GET NEW DRIVE.
FB6B FE04               CPI  NDISK              ;CALLING UNDEFINED DRIVE ?
FB6D D0                 RNC                     ;IF NO CY, H,L TELLS CPM YES.
FB6E 2127FD             LXI  H,DISKNO           ;GET OLD DRIVE NUMBER.
FB71 7E                 MOV  A,M                ;GET OLD DISK NUMBER.
                ;
                        IF   DUAL               ;IF DUAL DRIVE,
                        ANI  0FEH               ;CLEAR OUT BIT 0.
```

```
                        ENDIF
                ;
FB72 5F                 MOV   E,A                       ;PUT OLD DISK NO. IN D&E.
FB73 1600               MVI   D,0
FB75 212BFD             LXI   H,TRTAB                   ;GET ADDRESS OF TRACK TABLE.
FB78 E5                 PUSH  H                         ;SAVE ADDRESS OF TRTAB.
FB79 19                 DAD   D                         ;ADD DISK NO. TO ADDRESS.
FB7A DBF9               IN    TRACK                     ;READ 1771 TRACK REGISTER.
FB7C 77                 MOV   M,A                       ;PUT INTO TABLE.
FB7D 79                 MOV   A,C                       ;GET NEW DISK NUMBER.
                ;
                        IF    DUAL                      ;IF A DUAL DRIVE,
                        ANI   0FEH                      ;CLEAR BIT 0.
                        ENDIF
                ;
FB7E 5F                 MOV   E,A                       ;PUT NEW DISK NO. IN D&E.
FB7F E1                 POP   H                         ;RESTORE ADDRESS OF TRTAB.
FB80 19                 DAD   D                         ;ADD DISK NO. TO ADDRESS.
FB81 7E                 MOV   A,M                       ;GET NEW TRACK NUMBER.
FB82 D3F9               OUT   TRACK                     ;PUT INTO 1771 TRACK REG.
FB84 79                 MOV   A,C                       ;UPDATE OLD DISK NUMBER.
FB85 3227FD             STA   DISKNO
FB88 87                 ADD   A                         ;PUT BITS 1&2 AT 4&5.
FB89 87                 ADD   A
FB8A 87                 ADD   A
FB8B 87                 ADD   A
FB8C 3236FD             STA   LATCH                     ;SAVE NEW LATCH CODE.
FB8F 212FFD     DENSITY:LXI   H,DRVFLG                 ;POINT TO DRIVE DEN. FLAG
FB92 0600               MVI   B,0                       ;CLEAR REG B.
FB94 09                 DAD   B                         ;INDEX INTO DRIVE FLAG LOC.
FB95 7E                 MOV   A,M                       ;GET THE FLAG BYTE
FB96 B7                 ORA   A                         ;LOGGED IN?
FB97 FAC0FB             JM    LOGED                     ;YES, IT'S LOGGED.
FB9A E5                 PUSH  H                         ;NO, SAVE FLAG ADDRESS.
FB9B 3A36FD             LDA   LATCH                     ;GET LATCH CODE
FB9E D3FC               OUT   DCONT                     ;CHANGE LATCH OR DENSITY
                ;
                ;READ TRACK 0 SECTOR 1 FOR DENSITY BYTE AT 7E HEX.
                ;
FBA0 3E01               MVI   A,1                       ;SECTOR 1.
FBA2 3224FD             STA   SECT                      ;SAVE THE SECTOR VALUE.
FBA5 CDF4FB             CALL  HOME                      ;HOME THE DRIVE.
FBA8 2A25FD             LHLD  DMAADD                    ;GET CP/M DMA ADDRESS VALUE
FBAB E5                 PUSH  H                         ;SAVE IT ON THE STACK.
FBAC 2194FE             LXI   H,DBUFF                   ;POINT TO THE DMA BUFFER.
FBAF 2225FD             SHLD  DMAADD                    ;SET UP READ DMA ADDRESS.
                ;
                ;READ THE DATA USING READ ROUTINE.
                ;
FBB2 CD9DFC             CALL  READ                      ;CBIOS READ ROUTINE.
                ;
                ;GET DENSITY BYTE VALUE AND DETERMINE DRIVE STATUS.
                ;
FBB5 E1                 POP   H                         ;RESTORE DMA ADDRESS FROM THE S
FBB6 2225FD             SHLD  DMAADD                    ; AND RESTORE THE CP/M DMA ADDF
FBB9 E1                 POP   H                         ;RESTORE DENSITY FLAG ADDRESS
FBBA 3A12FF             LDA   DBUFF+7EH                 ;INDEX INTO DBUFF TO LOCATION I
FBBD F680               ORI   80H                       ;set logged in bit
FBBF 77                 MOV   M,A                       ;place it in flag table.
FBC0 011200     LOGED:  LXI   B,18                      ;index value throush drive tab
FBC3 E612               ANI   12H                       ;MASK DENSITY AND SIDE BITS OU
FBC5 B7                 ORA   A                         ;SINGLE DENSITY?
FBC6 2173FA             LXI   H,SDTAB                   ;point to start of tables
FBC9 CACDFB             JZ    DENSIT1                   ;yes, overlay param. block
                ;
                        IF    DUBSID
```

```
                        DAD   B                 ;no, add offset to next table
                        CPI   2                 ;single den doub sided?
                        JZ    DENSIT1           ;yes
                        DAD   B                 ;no
                        CPI   10H               ;doub den single sided?
                        JZ    DENSIT1           ;yes
                        ENDIF
                ;
FBCC 09                 DAD   B                 ;no, must be doub den , doub si
FBCD EB         DENSIT1:XCHG                    ;drive table pointer --> d,e
FBCE 1A                 LDAX  D                 ;get los and drive type byte.
FBCF 13                 INX   D                 ;bump pointer
FBD0 3233FD             STA   DENS              ;set current drive density.
FBD3 D5                 PUSH  D                 ;save drive table pointer.
FBD4 CDE7FB             CALL  PARINDX           ;compute parameter overlay area
FBD7 D1                 POP   D                 ;restore drive table pointer.
FBD8 010802             LXI   B,0208H           ;B = 2, C = 8 (count values).
FBDB 1A         MOVE:   LDAX  D                 ;GET XLTO BYTE.
FBDC 77                 MOV   M,A               ;AND PUT IT INTO DW TABLE FOR I
FBDD 13                 INX   D                 ;BUMP
FBDE 23                 INX   H                 ; POINTERS
FBDF 05                 DCR   B                 ;DECREASE COUNT.
FBE0 C2DBFB             JNZ   MOVE              ; AND LOOP TILL ZERO.
FBE3 09                 DAD   B                 ;NOW ADD INDEX INTO DPB0 AREA.
FBE4 73                 MOV   M,E               ;GET LOW POINTER BYTE.
FBE5 23                 INX   H                 ;BUMP POINTER.
FBE6 72                 MOV   M,D               ;GET HIGH POINTER BYTE.
                ;
                ;SELECT DRIVE AS A FUNCTION OF H,L
                ;
FBE7 2A27FD     PARINDX:LHLD  DISKNO            ;LOAD DISK NUMBER AND ZERO BYTE
FBEA 1133FA             LXI   D,DPBASE          ;POINT TO DISK PARM START.
FBED 29                 DAD   H                 ;*2
FBEE 29                 DAD   H                 ;*4
FBEF 29                 DAD   H                 ;*8
FBF0 29                 DAD   H                 ;*16
FBF1 19                 DAD   D                 ;COMPUTE INDEX FOR THE DRIVE
FBF2 AF                 XRA   A                 ;SET A = 0.
FBF3 C9                 RET                     ;RETURN FROM SELDSK.
                ;
                ; MOVE DISK TO TRACK ZERO.
                ;
FBF4 0E00       HOME:   MVI   C,0               ;SEEK TO TRACK ZERO.
FBF6 3E01               MVI   A,STPRAT          ;RESTORE COMMAND
FBF8 D3F8               OUT   DCOM              ;TELL CONTROLLER.
                ;
                ; SET TRACK NUMBER TO WHATEVER IS IN REGISTER C.
                ; ALSO PERFORM MOVE TO THE CORRECT TRACK (SEEK).
                ;
                SETTRK:
FBFA 2A36FD             LHLD  LATCH             ;get new and old latch.
FBFD 7C                 MOV   A,H               ;get latch value.
FBFE E6B7               ANI   0B7H              ;strip density and side bits.
FC00 67                 MOV   H,A               ;restore it.
                ;
                        IF    DUBSID            ;if using double sided drive.
                        LDA   DENS              ;check if double sided.
                        RRC
                        RRC                     ;look at bit 1.
                        JNC   NOTSID            ;if bit 1 = 0, it's single side
                        MOV   A,C               ;it's doub sided, so set track
                        RRC                     ;divide by 2.
                        MOV   B,A               ;save it in reg b.
                        MOV   A,L               ;get old latch value.
                        JC    SIDE2             ;change side if odd track.
                        ANI   0BFH              ;clear side bit from latch.
```

```
                        JMP   SETLAT            ;so set the latch.
                    ;
                    SIDE2:  ORI   40H           ;turn on side select bit.
                    SETLAT: STA   CLATCH        ;save it for later.
                        ANI   0BFH              ;clear side bit.
                        CALL  OLDLAT            ;check for drive change.
                        MOV   A,B               ;restore doub sided trk number.
                        ANI   7FH               ;clear bit 7.
                        MOV   C,A               ;trk number now in reg c.
                        JMP   TRKSET            ;check for density of track soi
                        ENDIF
                    ;
                        IF    NOT DUBSID        ;if not using double sided driv
FC01 C30FFC             JMP   NOTSID            ;Jump around subroutine.
                        ENDIF
                    ;
FC04 BC             OLDLAT: CMP   H             ;new = old?
FC05 3EFF               MVI   A,0FFH            ;if not, set = ff
FC07 C20BFC             JNZ   SFLAG
FC0A 2F                 CMA                     ;new = old, set = 0.
FC0B 3229FD         SFLAG:  STA   HLSF          ;save head load select flag.
FC0E C9                 RET
                    ;
FC0F 7D             NOTSID: MOV   A,L           ;set latch value.
FC10 3237FD             STA   CLATCH            ;save it
FC13 CD04FC             CALL  OLDLAT            ;check for drive change.
FC16 3A33FD         TRKSET: LDA   DENS          ;CHECK DRIVE DENSITY FLAG.
FC19 0F                 RRC                     ;IS BIT 0 = 0?
FC1A D22BFC             JNC   TRKSD             ;YES, WE ARE SINGLE DENSITY.
FC1D 79                 MOV   A,C               ;NO, RESTORE TRACK NUMBER.
FC1E FE01               CPI   1                 ;IS IT TRACK 1?
FC20 DA2BFC             JC    TRKSD             ;IF LESS THAN, SET SINGLE DENS
FC23 3A37FD             LDA   CLATCH            ;GET CURRENT LATCH VALUE.
FC26 F608               ORI   8                 ;SET FOR DOUBLE DENSITY.
FC28 C330FC             JMP   TRKDD
                    ;
FC2B 3A37FD         TRKSD:  LDA   CLATCH        ;GET CURRENT LATCH VALUE.
FC2E E6F7               ANI   0F7H              ;TURN OFF BIT 4 (SINGLE DENSIT
FC30 3237FD         TRKDD:  STA   CLATCH        ;SAVE NEW LATCH VALUE.
FC33 D3FC               OUT   DCONT             ;SELECT DISK AND MAKE DENSITY C
FC35 79                 MOV   A,C               ;RESTORE TRACK VALUE
FC36 3223FD             STA   TRK               ;UPDATE OLD WITH NEW.
                    ;
                    ; MOVE THE HEAD TO THE TRACK IN REGISTER A.
                    ;
FC39 C5             SEEK:   PUSH  B             ;SAVE B&C.
FC3A 47                 MOV   B,A               ;SAVE DESTINATION TRACK.
FC3B 3E0A               MVI   A,RTCNT           ;GET RETRY COUNT.
FC3D 3235FD         SRETRY: STA   SERCNT        ;STORE IN ERROR COUNTER.
FC40 DBF9               IN    TRACK             ;READ PRESENT TRACK NO.
FC42 B8                 CMP   B                 ;SAME AS NEW TRACK NO.?
FC43 C248FC             JNZ   NOTHR             ;JUMP IF NOT THERE.
FC46 C1             THERE:  POP   B             ;RESTORE B&C.
FC47 C9                 RET                     ;RETURN FROM SEEK.
FC48 78             NOTHR:  MOV   A,B           ;RESTORE A FROM B.
FC49 D3FB               OUT   DDATA             ;TRACK TO DATA REGISTER.
FC4B 3E1D               MVI   A,14H+STPRAT+HLAB ;GET STEP RATE, 00
FC4D D3F8               OUT   DCOM              ;SEEK WITH VERIFY.
                    ;
                        IF NOT DMACNTL
                        IN    WAIT              ;WAIT FOR INTRQ.
                        IN    DSTAT             ;READ STATUS.
                        ENDIF
                    ;
                        IF DMACNTL
FC4F CDE6FA             CALL  SLOOP             ;NO WAIT STATUS CHECK.
```

```
                        ENDIF
                ;
FC52 E691               ANI   91H               ;LOOK AT BITS.
FC54 CA46FC             JZ  THERE               ;OK IF ZERO.
FC57 3A35FD             LDA   SERCNT            ;GET ERROR COUNT.
FC5A 3D                 DCR   A                 ;DECREMENT COUNT.
FC5B C23DFC             JNZ   SRETRY            ;RETRY SEEK.
FC5E C1                 POP   B                 ;RESTORE B&C.
FC5F C5                 PUSH  B                 ;SAVE
FC60 CDB7FC             CALL  RECOV             ;IF SEEK RETRY = 10 CHECK
FC63 C1                 POP   B
FC64 79                 MOV   A,C               ;RECOVER TRACK NUMBER.
FC65 C339FC             JMP   SEEK              ; FOR CNTL-C FOR ABORT.
                ;
                ; SET DISK SECTOR NUMBER.
                ;
FC68 79         SETSEC:   MOV   A,C             ;GET SECTOR NUMBER.
FC69 3224FD               STA   SECT            ;PUT AT SECT # ADDRESS.
FC6C C9                   RET                   ;RETURN FROM SETSEC.
                ;
                ;TRANSLATE THE SECTOR GIVEN B,C USING
                ;THE TRANSLATE TABLE;GIVEN BY D,E
                ;
                SECTRAN:
FC6D 69                 MOV   L,C               ;GET PHYSICAL SECTOR NUMBER
FC6E 2C                 INR   L                 ;BUMP IT BY ONE.
FC6F 7A                 MOV   A,D               ;ARE WE USING NO XLAT TABLE?
FC70 B3                 ORA   E                 ;IT WILL BE ZERO IF NOT.
FC71 C8                 RZ                      ;RETURN IF IT IS ZERO.
FC72 EB                 XCHG                    ;H,L = TRANS
FC73 09                 DAD   B                 ;H,L = TRANS (SECTOR)
FC74 6E                 MOV   L,M               ;L   = TRANS (SECTOR)
FC75 2600               MVI   H,0               ;CLEAR REG H
FC77 C9                 RET                     ;H,L = TRANSLATED SECTOR
                ;
                ; SET DISK DMA ADDRESS.
                ;
FC78 60         SETDMA: MOV   H,B               ;MOVE B&C TO H&L.
FC79 69                 MOV   L,C
FC7A 2225FD             SHLD  DMAADD            ;PUT AT DMA ADR ADDRESS.
FC7D C9                 RET                     ;RETURN FROM SETDMA.
                ;
                ; HDLD - GET HEAD-LOAD BIT IF REQUIRED.
                ;
FC7E 3A29FD     HDLD:   LDA   HLSF              ;GET HEAD-LOAD FLAG.
FC81 B7                 ORA   A                 ;IS A = ZERO?
FC82 CA94FC             JZ    HDLD1             ;HOP IF SO.
FC85 2F                 CMA                     ;SET A = 0.
FC86 3229FD             STA   HLSF              ;SET FLAG = 0 IF NOT.
                ;
                ;IF CHANGING TO A NEW DRIVE, PERFORM A SEEK TO
                ;THE SAME TRACK TO UNLOAD THE HEAD ON NEW DRIVE.
                ;
FC89 DBF9               IN    TRACK             ;GET PRESENT TRACK
FC8B D3FB               OUT   DDATA             ;TELL CONTROLLER.
FC8D 3E15               MVI   A,14H+STPRAT
FC8F D3F8               OUT   DCOM
                ;
                        IF    NOT DMACNTL
                        IN    WAIT              ;WAIT FOR INTRQ.
                        ENDIF
                ;
                        IF    DMACNTL
FC91 CDE6FA             CALL  SLOOP             ;CHECK DMA STATUS PORT.
                        ENDIF
                ;
```

```
FC94 DBF8       HDLD1:  IN    DSTAT                 ;READ 1771 STATUS.
FC96 E620               ANI   20H                   ;LOOK AT HL BIT.
FC98 3E04               MVI   A,4
FC9A C8                 RZ                          ;RETURN IF HEAD IS NOT LOADED.
FC9B 97                 SUB   A                     ;HEAD IS ALREADY LOADED.
FC9C C9                 RET                         ;RETURN FROM HDLD.
                ;
                ; READ THE SECTOR AT SECT, FROM THE PRESENT TRACK,
                ; USE STARTING ADDRESS AT DMAADD.
                ;
FC9D 3E0A       READ:   MVI   A,RTCNT               ;GET RETRY COUNT.
                RRETRY:
                        IF DMACNTL
FC9F 01D080             LXI   B,80D0H               ;FLOPPY READ, FORCE INTERRUPT
FCA2 118040             LXI   D,4080H               ;DMA READ, DMA COUNT BYTE
FCA5 CDB1FA             CALL  DMARW                 ;ENTER COMMON READ/WRITE ROUTIN
                        ENDIF
                ;
                        IF NOT DMACNTL
                        MVI   B,80H                 ;FLOPPY READ COMMAND BYTE.
                        CALL  DSKSET                ;SET UP DISK CONTROLLER.
                        ORA   B                     ;'OR' IN THE READ COMMAND.
                READE:  OUT   DCOM                  ;SEND COMMAND TO 1771.
                RLOOP:  IN    WAIT                  ;WAIT FOR DRQ OR INTRQ.
                        ORA   A                     ;SET FLAGS.
                        JP    RDDONE                ;DONE IF INTRQ.
                        IN    DDATA                 ;READ A DATA BYTE FROM DISK.
                        MOV   M,A                   ;PUT BYTE INTO MEMORY.
                        INX   H                     ;INCREMENT MEMORY POINTER.
                        JMP   RLOOP                 ;KEEP READING.
                RDDONE: IN    DSTAT                 ;READ DISK STATUS.
                        ENDIF
                ;
                        IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
                        EI                          ;ALLOW AGAIN HERE.
                        ENDIF
                ;
FCA8 E69D               ANI   9DH                   ;LOOK AT ERROR BITS.
FCAA C8                 RZ                          ;RETURN IF NONE.
FCAB CDC5FC             CALL  ERCHK                 ;CHECK FOR SEEK ERROR.
FCAE C29FFC             JNZ   RRETRY                ;TRY TO READ AGAIN.
FCB1 CDB7FC             CALL  RECOV                 ;CHECK FOR ABORT OR CONTINUE
FCB4 C39DFC             JMP   READ                  ;IF NOT CNTL-C, TRY TO READ AGA
                ;
                ;RECOV
                ;THIS ROUTINE IS CALLED BY ANY READ,WRITE,SEEK ROUTINE IF THE R
                ;COUNT GOES TO 10. IF IT DOES,THIS ROUTINE CALLS CONIN FOR A K
                ;BE PUSHED. IF THE KEY IS A CNTL-C, THEN A WARMBOOT IS EXECUTED
                ;ANY OTHER KEY IS PUSHED, THEN A RETURN IS MADE BACK TO THE CAL
                ;AND THAT ROUTINE IS RETRIED FOR 10 MORE TIMES.
                ;
                RECOV:
FCB7 0E65               MVI   C,'e'                 ;ERROR CODE
FCB9 CD5CFB             CALL  CONOT                 ;PRINT IT
FCBC CD51FB             CALL  CONIN                 ;CHECK FOR PUSHED KEY.
FCBF FE03               CPI   03H                   ;IS IT A CNTL-C ?
FCC1 C0                 RNZ                         ;RETURN TO CALLER IF NOT.
FCC2 C3EFFA             JMP   WBOOT                 ;YES, DO WARMBOOT.
                ;
                ; ERCHK - CHECK FOR RECORD NOT FOUND ERROR.
                ;
FCC5 E610       ERCHK:  ANI   10H                   ;IF RECORD NOT FOUND,
FCC7 C2CFFC             JNZ   CHKSK                 ;DO A CHECK ON SEEK.
FCCA 3A34FD     CHKOK:  LDA   ERCNT                 ;GET RETRYS ALLOWED
FCCD 3D                 DCR   A                     ;DECREASE IT,
FCCE C9                 RET                         ;AND RETURN WITH NUMBER.
```

```
                        ;
                        ;CHECK FOR SEEK TO CORRECT TRACK,
                        ;AND CHANGE IF NECESSARY.
                        ;
                                IF NOT DMACNTL
                        CHKSK:  MVI   A,0C4H            ;SEND COMMAND TO 1771
                                OUT   DCOM              ;TO READ ADDRESS.
                                IN    WAIT              ;WAIT FOR DRQ OR INTRQ.
                                IN    DDATA             ;READ THE TRACK ADDRESS.
                                PUSH  PSW               ;SAVE IT ON THE STACK.
                        CHKS2:  IN    DMACHK            ;WAIT FOR INTRQ.
                                ORA   A                 ;SET FLAGS.
                                JP    CHKS3             ;DONE WITH READ ADR OP.
                                IN    DDATA             ;READ ANOTHER BYTE.
                                JMP   CHKS2             ;DO IT AGAIN.
                        CHKS3:  IN    DSTAT             ;READ DISK STATUS.
                                ENDIF
                        ;
                                IF    DMACNTL
FCCF 21F9F9             CHKSK:  LXI   H,BIOS-7          ;POINT TO UNUSED SPACE
FCD2 01D0C4                     LXI   B,0C4D0H          ;READ ADDRESS, FORCE INTERRUPT
FCD5 110640                     LXI   D,04006H          ;DMA READ, COUNT BYTE
FCD8 CDBCFA                     CALL  DMARWE            ;READ THE ID USING DMA CONTROL
FCDB B7                         ORA   A                 ;SET FLAGS.
FCDC CAE5FC                     JZ    CHKS4             ;READ ADR OK IF 0.
FCDF CDF4FB                     CALL  HOME              ;OTHERWISE, HOME FIRST.
FCE2 C3E9FC                     JMP   CHKS5
                                ENDIF
                        ;
                                IF    NOT DMACNTL
                        CHKS4:  POP   PSW               ;UPDATE TRACK REGISTER.
                                ENDIF
                        ;
                                IF    DMACNTL
FCE5 DBFA               CHKS4:  IN    SECTP             ;GET THE TRACK BYTE
                                ENDIF
                        ;
FCE7 D3F9                       OUT   TRACK
FCE9 3A23FD             CHKS5:  LDA   TRK               ;GET REQUIRED TRACK NO.
FCEC CD39FC                     CALL  SEEK              ;MOVE THE HEAD TO IT.
FCEF C3CAFC                     JMP   CHKOK             ;EXIT FROM ERROR CHECK.
                        ;
                                IF    NOT DMACNTL
                        DSKSET: STA   ERCNT             ;STORE IN ERROR CTR.
                                MVI   A,0D0H            ;CAUSE INTERRUPT.
                                OUT   DCOM
                                XTHL                    ;SOME
                                XTHL                    ; DELAY
                                ENDIF
                        ;
                                IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
                                DI                      ;DISABLE THEM HERE.
                                ENDIF
                        ;
                                IF    NOT DMACNTL
                                LHLD  DMAADD            ;GET STARTING ADDR.
                                LDA   SECT              ;GET SECTOR NUMBER.
                                OUT   SECTP             ;SET SECTOR INTO 1771.
                                CALL  HDLD              ;GET HEAD-LOAD BIT?
                                RET                     ;RETURN TO CALLER
                                ENDIF
                        ;
                        ; WRITE THE SECTOR AT SECT, ON THE PRESENT TRACK,
                        ; USE STARTING ADDRESS AT DMAADD.
                        ;
FCF2 3E0A               WRITE:  MVI   A,RTCNT           ;GET RETRY COUNT.
```

```
                WRETRY:
                        IF      DMACNTL
FCF4 01D0A0             LXI     B,0A0DOH        ;FLOPPY WRITE, FORCE INTERRUPT.
FCF7 118080             LXI     D,08080H        ;DMA WRITE, DMA COUNT BYTE.
FCFA CDB1FA             CALL    DMARW           ;ENTER COMMON READ/WRITE ROUTINE
                        ENDIF
                ;
                        IF      NOT DMACNTL
                        MVI     B,0A0H          ;FLOPPY WRITE COMMAND BYTE.
                        CALL    DSKSET          ;SET UP FLOPPY CONTROLLER.
                        ORA     B               ;'OR' IN WRITE COMMAND.
                WRITE2: OUT     DCOM
                WLOOP:  IN      WAIT            ;WAIT FOR READY.
                        ORA     A               ;SET FLAGS.
                        JP      WDONE           ;HOP OUT WHEN DONE.
                        MOV     A,M             ;GET BYTE FROM MEM.
                        OUT     DDATA           ;WRITE ONTO DISK.
                        INX     H               ;INCREMENT MEM PTR.
                        JMP     WLOOP           ;KEEP WRITING.
                WDONE:  IN      DSTAT           ;READ DISK STATUS.
                        ENDIF
                ;
                        IF INTRP AND NOT DMACNTL;IF INTERRUPTS ALLOWED,
                        EI                      ;ENABLE AGAIN HERE.
                        ENDIF
                ;
FCFD E6FD               ANI     0FDH            ;LOOK AT THESE BITS.
FCFF C8                 RZ                      ;RETURN IF NO ERR.
FD00 CDC5FC             CALL    ERCHK           ;CHECK/CORRECT SEEK ERR.
FD03 C2F4FC             JNZ     WRETRY          ;TRY TO WRITE AGAIN.
FD06 CDB7FC             CALL    RECOV           ;CHECK FOR ABORT
FD09 C3F2FC             JMP     WRITE           ;RETRY WRITE AGAIN.
                ;
                ;LIST STATUS CHECK ROUTINE
                ;
FD0C CD14FD     PRSTAT: CALL    PSTAT           ;CHECK PRINTER STATUS PORT.
                ;
FD0F 3E00               MVI     A,0             ;RETURN STATUS ACTIVITY.
                ;
                        IF      TARDLL OR RDYLO
                        RNZ
                        ENDIF
                ;
                        IF      RDYH1
FD11 C8                 RZ
                        ENDIF
                ;
FD12 2F                 CMA                     ;INVERT IT
                ;
                ; PUNCH AND READER ARE NOT SUPPORTED.
                ;
                PUNCH:
FD13 C9         READER: RET
                ;
                ;PSTAT - PRINTER STATUS CHECK ROUTINE.
                ;
FD14 DB02       PSTAT:  IN      LSTAT           ;READ PRINTER STATUS PORT.
                ;
                        IF      NOT TARDEL
FD16 E602               ANI     LRBIT
                        ENDIF
                ;
                        IF      TARDEL
                        ANI     81H             ;MASK READY BITS
                        XRI     81H
                        ENDIF
```

```
                        ;
FD18 C9                         RET                             ;RETURN TO CALLER
                        ;
                        ; WRITE A CHARACTER ON LISTING DEVICE.
                        ;
                        LIST:
                                IF    LSTNUL                    ;IF NULLS OR PAGING,
                                MVI   A,0DH                     ;IF IT'S A CR,
                                CMP   C                         ;THEN HOP OUT TO
                                JZ    LINUL                     ;NULL ROUTINE.
                                ENDIF
                        ;
                                IF    LSTPAG                    ;IF PAGING
                                MVI   A,0AH                     ;GET A LINEFEED
                                CMP   C                         ;DOES IT MATCH?
                                JZ    LINUL3
                                MOV   A,C
                                CPI   0CH
                                RZ
                                ENDIF
                        ;
FD19 CD14FD             LTBSY:  CALL PSTAT                      ;READ LISTER STATUS.
                        ;
                                IF    TARDEL OR RDYLO
                                JNZ   LTBSY                     ;LOOP TILL LOW.
                                ENDIF
                        ;
                                IF    NOT TARDEL AND RDYHI
FD1C CA19FD                     JZ    LTBSY                     ;LOOP TILL HIGH.
                                ENDIF
                        ;
FD1F 79                         MOV   A,C                       ;GET DATA BYTE.
FD20 D303                       OUT   LDATA                     ;PRINT IT.
FD22 C9                         RET                             ;RETURN FROM LIST.
                        ;
                                IF    LSTNUL                    ;IF LIST NULLS
                        LINUL:  PUSH B                          ;SAVE B&C.
                                MVI   B,(LNULL AND 0FFH)+1      ;GET NULL COUNT
                        LINUL1: CALL LTBSY                      ;PRINT (CR FIRST).
                                MVI   C,0                       ;GET NULL CHAR.
                                DCR   B                         ;DECREMENT COUNTER.
                                JNZ   LINUL1                    ;DO NEXT NULL.
                                JMP   LINUL2                    ;EXIT THE ROUTINE.
                                ENDIF
                        ;
                                IF    LSTPAG                    ;IF LIST DEV. PAGING,
                        LINUL3: PUSH B                          ;SAVE B,C PAIR
                                LDA   LFCNT                     ;GET LINE-FEED COUNT.
                                INR   A                         ;INCREMENT IT.
                                STA   LFCNT                     ;SAVE IT BACK.
                                CPI   LINCNT-(LINCNT/11)        ;END OF PAGE?
                                MVI   B,1                       ;SET UP FOR 1 LF.
                                JNZ   NOTEOP                    ;HOP IF NOT END.
                                XRA   A                         ;SET LF COUNT = 0.
                                STA   LFCNT
                                MVI   B,(LINCNT/11)+1           ;BETWEEN PAGES.
                        NOTEOP: MVI   C,0AH                     ;GET LINE FEED CODE.
                        LSTPA1: CALL LTBSY                      ;PRINT LINE-FEED.
                                DCR   B                         ;DECREMENT LF COUNTER.
                                JNZ   LSTPA1                    ;DO NEXT LINE FEED?
                                ENDIF
                        ;
                                IF    LSTNUL OR LSTPAG          ;IF NULLS OR PAGING,
                        LINUL2: POP   B                         ;RESTORE B&C.
                                MOV   A,C                       ;RESTORE A.
                                RET                             ;RETURN FROM LIST.
```

```
                        ENDIF
                ;
FD22 =          ENDPROG EQU     $-1             ;ENDING ADDRESS.
                ;
                ;NOTE:  AS THERE ARE ONLY SIX (6) SECTORS AVAILABLE FOR CBIOS
                ;THE SECOND SYSTEM TRACK (1), THE LAST ADDRESS BEFORE THIS POIN
                ;SHOULD BE NO GREATER THAN THE CBIOS STARTING ADDRESS + 037F (H
                ;THIS WILL NORMALLY BE XD7F (HEX).
                ;
                ; BIOS SCRATCH AREA.
                ;
FD23            TRK:    DS   1                  ;CURRENT TRACK NUMBER.
FD24            SECT:   DS   1                  ;CURRENT SECTOR NUMBER.
FD25            DMAADD: DS   2                  ;DISK TRANSFER ADDRESS.
                ;
                ; THE NEXT SEVERAL BYTES, BETWEEN STARTZ AND
                ; ENDZ, ARE SET TO ZERO AT COLD BOOT TIME.
                ;
                STARTZ:                         ;START OF ZEROED AREA.
                ;
FD27            DISKNO: DS   2                  ;DISK NUMBER
                ;
                ; SPECIAL FLAGS.
                ;
FD29            HLSF:   DS   1                  ;HEAD-LOAD SELECT FLAG.
FD2A            LFCNT:  DS   1                  ;PAGING LINE-FEED COUNT.
                ;
                ; TRTAB - DISK TRACK TABLE - PRESENT POSITION OF
                ;       HEADS FOR UP TO 4 DRIVES.
                ;
FD2B            TRTAB:  DS   4
FD2F            DRVFLG: DS   4                  ;DRIVE DENSITY FLAGS.
FD33            DENS:   DS   1                  ;CURRENT DRIVE DENSITY VALUE.
FD34            ERCNT:  DS   1                  ;ERROR COUNT FOR RETRIES.
FD35            SERCNT: DS   1                  ;SEEK RETRY COUNTER.
FD36            LATCH:  DS   1                  ;NEW CODE FOR LATCH.
FD37            CLATCH: DS   1                  ;CURRENT CODE IN LATCH.
                ;
FD38 =          ENDZ    EQU  $
                ;
FD38 =          BEGDAT  EQU  $
FD38            DIRBUF: DS   128                ;DIRECTORY BUFFER
FDB8            ALV0:   DS   31
FDD7            CSV0:   DS   24
FDEF            ALV1:   DS   31
FE0E            CSV1:   DS   24
FE26            ALV2:   DS   31
FE45            CSV2:   DS   24
FE5D            ALV3:   DS   31
FE7C            CSV3:   DS   24
FE94 =          ENDDAT  EQU  $
015C =          DATSIZ  EQU  $-BEGDAT           ;TOTAL SIZE OF DISK PARM STORAG
                ;
FE94            DBUFF:  DS   128                ;128 BYTE DENSITY SELECT BUFFER
                ;
FD22                    ORG  ENDPROG            ;SHOW ACTUAL ENDING ADDRESS OF
FD22                    END
```

BIOS

```
B>TYPE 2DBOOT64.PRN

                ;
                ; TARBELL ELECTRONICS CP/M COLDSTART LOADER
                ; VERSION OF 7-31-80.
                ;-----------------------------------------------------------------
                ; Modified for DMA Control - 1-5-80.
                ; Modified for reading larger bios from TRK 1  - 6-5-80.
                ; Modified for Tarbell CPU Card  -  7-3-80.
                ; G.W.Mulchin
                ; Tarbell Electronics
                ;-----------------------------------------------------------------
                ; Copyright (c) 1980 Tarbell Electronics
                ;
                ;
                ;*****************************************************************
                ;*
                ;*                          ** NOTE **
                ;*                          ============
                ;*
                ;*      The equate for Double Density (DOUBDEN) must only be
                ;* set true for a disk which is formatted in double density onl
                ;* and one which you wish to put an operating system on to.
                ;* Otherwise, leave it false if you are building an operating
                ;* system on to a single density formatted disk.
                ;*
                ;*****************************************************************
                ;
                ;
                ;
                ; THIS PROGRAM IS LOADED AT LOCATION ZERO BY THE BOOTSTRAP PROG
                ; AND EXECUTED. ITS PURPOSE IS TO LOAD AND EXECUTE THE CP/M DI
                ; OPERATING SYSTEM AT THE TOP OF THE MEMORY IN USE.
                ;
0000 =          FALSE   EQU  0          ;DEFINE VALUE OF FALSE.
FFFF =          TRUE    EQU  NOT FALSE  ;DEFINE VALUE OF TRUE.
                ;
                ;********** THIS IS THE AREA TO MAKE CHANGES IN **************
                ;********** FOR DIFFERENT SYSTEM CONFIGURATIONS ***************
                ;
0040 =          MSIZE   EQU  64         ;MEMORY SIZE IN DECIMAL KB.
0000 =          TARBELL EQU  FALSE      ;TRUE IF USING TARBBELL CPU.
0000 =          DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED SYSTEMS.
0000 =          DELTA   EQU  FALSE      ;TRUE IF USING DELTA CPU CARD
0000 =          DOUBDEN EQU  FALSE      ;TRUE IF DOUB. DEN DISK.
FFFF =          DMACNTL EQU  TRUE       ;TRUE IF USING DMA CONTROL
0000 =          BASE    EQU  0          ;TARBELL I/O PORTS (00 or 10 HEX)
001A =          SPT     EQU  26         ;NUMBER OF SECTORS PER TRACK.
001A =          DDS     EQU  26         ;sectors in trk 1 , (Range = 26 to 51)
00F8 =          DISK    EQU  0F8H       ;DISK PORT BASE ADDRESS.
                ;
                ;*****************************************************************
                ;
                        IF   TARBELL
                IO      EQU  BASE       ;i/o ports on tarbell cpu.
                MMENB   EQU  IO+10      ;memory management enable port.
                MEMMAG  EQU  BASE+32    ;memory management port.
                        ENDIF

00E0 =          ADRO    EQU  0E0H       ;DMA ADDRESS PORT.
```

```
00E1 =          WCTO    EQU  0E1H          ;DMA WORD COUNT PORT.
00E8 =          CMND    EQU  0E8H          ;DMA COMMAND PORT.
00F8 =          DCOM    EQU  DISK          ;COMMAND PORT.
00F8 =          DSTAT   EQU  DISK          ;STATUS PORT.
00F9 =          TRACK   EQU  DISK+1        ;TRACK PORT.
00FA =          SECT    EQU  DISK+2        ;SECTOR PORT.
00FB =          DATA    EQU  DISK+3        ;DATA PORT.
00FC =          WAIT    EQU  DISK+4        ;WAIT PORT.
00FC =          DCONT   EQU  DISK+4        ;CONTROL PORT.
00FD =          DMACHK  EQU  DISK+5        ;DMA CHECK PORT.
00FF =          PANEL   EQU  0FFH          ;front panel machines.
B000 =          CBASE   EQU  (MSIZE-20)*1024
E400 =          CPMB    EQU  CBASE+3400H;START OF CP/M.
FA00 =          BOOTE   EQU  CPMB+1600H ;COLD BOOT ENTRY POINT.
0019 =          SDS     EQU  25            ;always 25 sectors to read in trk 1.
0033 =          NSECTS  EQU  SDS + DDS     ;SECTORS OF CP/M.
000A =          RTCNT   EQU  10            ;NUMBER OF RETRYS.
                ;
0000                    ORG  0             ;START OF LOADER.
                ;
                BOOT:
                        IF   TARBELL       ;IF USING TARBELL CPU
                        LXI  D,1000H       ;COUNT=16, DATA BYTE = 0
                        MVI  C,MEMMAG AND 0FFH
                MLOOP:  MOV  A,E           ;GET ADDRESS VALUE
                        ORA  C             ;MAKE I/O PORT VALUE
                        STA  MOUT+1        ;MODIFY PORT ON THE FLY
                        MOV  A,E           ;GET INIT VALUE.
                        CMA                ;FLIP IT FOR RAM ON CPU
                MOUT:   OUT  BASE          ;PUT IT TO RAM ON CPU
                        INR  E             ;BUMP DATA VALUE
                        DCR  D             ;DECREASE COUNT
                        JNZ  MLOOP         ;LOOP 16 TIMES.
                        OUT  MMENB         ;ENABLE MEMORY MANAGEMENT.
                        ENDIF
                ;
                        IF   DELTA         ;IF USING DELTA CPU.
                        MVI  A,1           ;GET A 1 IN REG A.
                        OUT  9             ; AND DISABLE CPU ROM SLOT.
                        ENDIF
                ;
0000 1E0A               MVI  E,RTCNT       ;GET RETRY COUNT.
0002 310001     BLOOP:  LXI  SP,100H       ;SET STACK POINTER.
0005 2100E4             LXI  H,CPMB        ;CP/M STARTS HERE.
0008 1633               MVI  D,NSECTS      ;NUMBER OF SECTORS TO READ.
000A 0E02               MVI  C,2           ;SECTOR NUMBER.
000C 0604       RNTRK:  MVI  B,4           ;FOR HEAD LOAD.
000E CD2900     RNSEC:  CALL READ          ;READ FIRST SECTOR.
0011 15                 DCR  D             ;IF DONE,
0012 CA00FA             JZ   BOOTE         ;GO TO CP/M.
0015 0600               MVI  B,0           ;FOR NO HEAD LOAD.
0017 0C                 INR  C             ;INCREMENT SECTOR COUNT.
0018 79                 MOV  A,C           ;DONE WITH
0019 FE1B       SECCMP: CPI  SPT+1         ;THIS TRACK?
001B DA0E00             JC   RNSEC         ;IF NOT, READ NEXT SECTOR.
                ;
                        IF   DOUBDEN AND NOT DUBSID
                        MVI  A,DDS + 1     ;number of sectors to read on trk 2.
                        STA  SECCMP+1      ;modify sector compare value.
                        MVI  A,8           ;GET SET DOUBLE DENSITY CODE
                        OUT  WAIT          ;SET LATCH FOR D.DENSITY
                        ENDIF
                ;
                        IF   NOT DUBSID
001E 3E5B               MVI  A,5BH         ;STEP COMMAND.
0020 D3F8               OUT  DCOM          ;ISSUE IT.
```

```
0022 DBFC                IN    WAIT        ;WAIT UNTIL DONE.
                         ENDIF
                ;
                         IF    DUBSID      ;IF DOUBLE SIDED SYSTEM.
                         MVI   A,40H       ;SIDE SELECT COMMAND.
                         OUT   DCONT       ;ISSUE IT.
                         ENDIF
                ;
0024 0E01                MVI   C,1         ;SECTOR NUMBER.
0026 C30C00              JMP   RNTRK       ;READ NEXT TRACK.
                ;
                READ:
                         IF    DMACNTL     ;IF USING DMA CONTROL.
0029 3E41                MVI   A,41H       ;SET UP FOR CHAN 0 REQ.
002B D3E8                OUT   CMND
002D 3E7F                MVI   A,7FH       ;COUNT FOR 128 BYTES
002F D3E1                OUT   WCT0
0031 3E40                MVI   A,40H       ;READ COMMAND
0033 D3E1                OUT   WCT0
0035 7D                  MOV   A,L         ;GET LOW ADDRESS BYTE
0036 D3E0                OUT   ADR0
0038 7C                  MOV   A,H         ;HIGH ADDRESS BYTE
0039 D3E0                OUT   ADR0
                         ENDIF
                ;
003B 79                  MOV   A,C         ;SECTOR IN A.
003C D3FA                OUT   SECT        ;SET SECTOR REGISTER.
003E 3E88                MVI   A,88H       ;COMMAND FOR READ.
0040 B0                  ORA   B           ;GET HEAD LOAD BIT.
0041 D3F8                OUT   DCOM        ;ISSUE COMMAND.
                ;
                         IF    NOT DMACNTL;IF NOT USING DMA CONTROL.
                RLOOP:   IN    WAIT        ;WAIT FOR DRQ.
                         ORA   A           ;SET FLAGS.
                         JP    CHECK       ;JUMP IF DONE.
                         IN    DATA        ;READ DATA.
                         MOV   M,A         ;PUT IN MEMORY.
                         INX   H           ;INCREMENT POINTER.
                         JMP   RLOOP       ;LOOP UNTIL DONE.
                         ENDIF
                ;
                         IF   DMACNTL
0043 DBFD       SLOPP:   IN    DMACHK      ;CHECK DMA STATUS
0045 07                  RLC               ; BIT 7
0046 DA4300              JC    SLOPP       ;LOOP IF CARRY
0049 C5                  PUSH  B           ;SAVE B,C PAIR
004A 018000              LXI   B,128       ;COUNT SET FOR 128 BYTES
004D 09                  DAD   B           ;ADJUST H,L BY 128 BYTES
004E C1                  POP   B           ;RESTORE BC
                         ENDIF
                ;
004F DBF8       CHECK:   IN    DSTAT       ;READ STATUS.
0051 E69D                ANI   9DH         ;LOOK AT ERROR BITS.
0053 C8                  RZ                ;OK IF ZERO.
0054 1D                  DCR   E           ;DECREMENT RETRY COUNT.
0055 C20200              JNZ   BLOOP       ;TRY AGAIN IF NOT ZERO.
0058 2F                  CMA               ;flip for front panel
0059 D3FF                OUT   PANLL       ;show error code from floppy.
005B C35B00     HERE:    JMP   HERE        ;LOOP.
                ;
007D                     ORG   7DH         ;PUT JUMP HERE.
                ;
                         IF    DOUBDEN AND NOT DUBSID     ;IF RUNNING DOUBLE DENS
                         RST   0           ;DO RESTART 0
                         DB    0DDH        ;THIS BYTE MUST BE HERE IF DOUB DEN.
                         DB    0           ;THIS BYTE UNUSED
```

```
                        ENDIF
                ;
                        IF    DOUBDEN AND DUBSID
                        RST   0
                        DB    0DFH
                        DB    0
                        ENDIF
                ;
                        IF    NOT DOUBDEN AND NOT DUBSID
 007D C7                RST   0             ;DO WARM BOOT WITH RST INST.
 007E E5                DB    0E5H
 007F 00                DB    0
                        ENDIF
                ;
                        IF    NOT DOUBDEN AND DUBSID
                        RST   0
                        DB    0E7H
                        DB    0
                        ENDIF
                ;
 0080                   END   BOOT        ;END OF BOOT.

B>D100  CPM64.COM
0100 C3 2C 01 43 4F 50 59 52 49 47 48 54 20 28 43 29 .,.COPYRIGHT (C)
0110 20 44 49 47 49 54 41 4C 20 52 45 53 45 41 52 43  DIGITAL RESEARC
0120 48 2C 20 31 39 37 39 20 20 20 20 20 31 00 08 11 H, 1979     1...
0130 5D 00 1A FE 20 CA 91 01 FE 3F CA 91 01 21 00 00 ]... ....?...!..
0140 1A 13 FE 20 CA 5F 01 B7 CA 5F 01 D6 30 FE 0A D2 ... ._....._..O...
0150 72 01 29 E5 29 29 C1 09 4F 06 00 09 C3 40 01 7C r.).))..O....@.|
0160 B7 C2 72 01 7D FE 10 DA 72 01 2E 00 67 29 29 C3 ..r.}...r...g)).
0170 A5 01 11 7B 01 CD 75 03 C3 00 00 0D 0A 49 4E 56 ...{..u......INV
0180 41 4C 49 44 20 4D 45 4D 4F 52 59 20 53 49 5A 45 ALID MEMORY SIZE
0190 24 21 00 00 24 CA A1 01 7E 2F 77 BE 2F 77 CA 94 $!..$...~/w./w..
01A0 01 7C E6 FC 67 E5 2A 06 00 22 7A 03 E1 E5 7C 0F .|..g.*.."z...|.
01B0 0F E6 3F C2 B8 01 3E 40 47 21 8D 03 3E 30 77 23 ..?...>@G!..>0w#
-D
01C0 77 21 8E 03 34 7E FE 3A DA CF 01 36 30 2B 34 05 w!..4~.:...60+4.
01D0 C2 C1 01 11 7E 03 CD 75 03 21 01 08 4E 23 46 C5 ....~..u.!..N#F.
01E0 21 00 09 11 8F 03 78 B1 CA 2C 02 0B C5 0E 0F E5 !.....x..,......
01F0 1A BE C2 1A 02 13 23 0D CA 20 02 C3 F0 01 01 AF ......#.. ......
0200 3D C2 00 02 21 F3 76 22 7A 03 21 77 03 36 CD 11 =...!.v"z.!w.6..
0210 88 02 21 05 00 19 EB C3 75 03 E1 23 C1 C3 E3 01 ..!.....u..#....
0220 E1 C1 2B 11 8E 03 1A 77 2B 1B 1A 77 01 7A 03 0A ..+....w+..w.z..
0230 FE 06 3E 00 C2 5A 02 02 C1 E1 C5 78 C6 03 47 7D ..>..Z.....x..G}
0240 91 6F 7C 98 67 22 7C 03 EB 21 00 09 C1 C5 3A 6D .o|.g"|..!....:m
0250 00 FE 20 CA 63 02 09 C3 70 02 21 FF 01 22 07 03 .. .c...p.!.."..
0260 C3 06 03 78 B1 CA 70 02 0B 7E 12 13 23 C3 63 02 ...x..p..~..#.c.
0270 C1 E5 2A 7C 03 EB 21 00 01 19 3A 6D 00 FE 20 CA ..*|..!...:m.. .
-D
0280 85 02 11 00 09 78 B1 CA C0 02 C3 A4 02 0D 0A 53 .....x.........S
0290 59 4E 43 52 4F 4E 49 5A 41 54 49 4F 4E 20 45 52 YNCRONIZATION ER
02A0 52 4F 52 24 0B 7D E6 07 C2 B0 02 E3 7E 23 E3 6F ROR$.}......~#.o
02B0 7D 17 6F D2 BC 02 1A 84 12 C3 BC 02 13 C3 85 02 }.o.............
02C0 D1 11 00 12 2A 7A 03 0E 06 1A BE C2 5A 02 23 13 ....*z......Z.#.
02D0 0D C2 C9 02 3A 6D 00 FE 20 CA 6D 03 06 30 21 00 ....:m.. .m...!.
02E0 09 7E B7 C2 09 03 23 05 C2 E1 02 EB 2A 01 08 01 .~....#.....*...
02F0 80 FF 09 44 4D 21 00 09 78 B1 CA 09 03 0B 1A 77 ...DM!..x......w
0300 13 23 C3 F8 02 01 C3 06 03 21 01 08 4E 23 46 21 .#.......!..N#F!
0310 00 09 09 44 21 5F 03 3E 30 77 23 77 05 CA 31 03 ...D!_.>0w#w..1.
0320 21 60 03 34 7E FE 3A DA 1C 03 36 30 2B 34 C3 1C !`.4~.:...60+4..
0330 03 2A 8D 03 22 65 03 11 40 03 CD 75 03 C3 00 00 .*.."e..@..u....
-D
0340 0D 0A 52 45 41 44 59 20 46 4F 52 20 22 53 59 53 ..READY FOR "SYS
0350 47 45 4E 22 20 4F 52 0D 0A 22 53 41 56 45 20 33 GEN" OR.."SAVE 3
0360 34 20 43 50 4D 36 34 2E 43 4F 4D 22 24 11 00 17 4 CPM64.COM"$...
0370 2A 7C 03 19 E9 0E 09 C3 05 00 00 AC 00 E3 0D 0A *|.............
```

```
0380 43 4F 4E 53 54 52 55 43 54 49 4E 47 20 36 34 6B CONSTRUCTING 64k
0390 20 43 50 2F 4D 20 76 65 72 73 20 32 2E 32 24 00  CP/M vers 2.2$.
03A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
03B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
03C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
03D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
03E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
03F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-D
0400 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0410 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0420 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0430 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0440 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0450 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0460 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0470 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0480 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0490 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
04A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
04B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-D
04C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
04D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
04E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
04F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0500 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0510 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0520 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0530 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0540 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0550 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0560 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0570 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-D
0580 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0590 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
05A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
05B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
05C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
05D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
05E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
05F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0600 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0610 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0620 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0630 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-D
0640 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0650 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0660 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0670 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0680 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0690 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
06A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
06B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
06C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
06D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
06E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
06F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-D
0700 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0710 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0720 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0730 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0740 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

```
0750 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0760 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0770 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0780 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0790 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
07A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
07B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-D
07C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
07D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
07E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
07F0 00 00 00 00 00 00 00 00 A1 20 5E 02 00 1A 3D 03 ......... ^...=.
0800 FA 00 1A 1D 3E FA CD A6 1D C9 11 00 00 0E 03 CD ....>...........
0810 05 00 C9 11 00 00 0E 01 CD 05 00 C9 21 AB 1E 71 ............!..q
0820 3A AB 1E E6 7F 5F 16 00 0E 02 CD 05 00 C9 0E 0D :...._..........
0830 CD 1C 08 0E 0A CD 1C 08 C9 21 AD 1E 70 2B 71 CD .........!..p+q.
0840 2E 08 2A AC 1E EB 0E 09 CD 05 00 C9 11 00 00 0E ..*.............
0850 0C CD 05 00 C9 11 00 00 0E 0D CD 05 00 C9 21 AF ..............!.
0860 1E 71 2A AF 1E 26 00 EB 0E 0E CD 05 00 C9 21 B1 .q*..&........!.
0870 1E 70 2B 71 2A B0 1E EB 0E 0F CD 05 00 32 AE 1E .p+q*........2..
-D
0880 C9 21 B3 1E 70 2B 71 2A B2 1E EB 0E 10 CD 05 00 .!..p+q*........
0890 32 AE 1E C9 21 B5 1E 70 2B 71 2A B4 1E EB 0E 11 2...!..p+q*.....
08A0 CD 05 00 32 AE 1E C9 11 00 00 0E 12 CD 05 00 32 ...2...........2
08B0 AE 1E C9 21 B7 1E 70 2B 71 2A B6 1E EB 0E 13 CD ...!..p+q*......
08C0 05 00 C9 21 B9 1E 70 2B 71 2A B8 1E EB 0E 14 CD ...!..p+q*......
08D0 05 00 C9 21 BB 1E 70 2B 71 2A BA 1E EB 0E 15 CD ...!..p+q*......
08E0 05 00 C9 21 BD 1E 70 2B 71 2A BC 1E EB 0E 16 CD ...!..p+q*......
08F0 05 00 32 AE 1E C9 21 BF 1E 70 2B 71 2A BE 1E EB ..2...!..p+q*...
0900 31 00 01 DB 79 DB 7B DB FF E6 02 C2 07 30 D3 7F 1...Y.{......0..
0910 06 02 21 42 30 7D D3 79 7C D3 7A DB 78 E6 04 CA ..!B0}.Y|.z.x...
0920 1B 30 DB 79 E6 03 FE 02 D2 00 30 DB 7B 17 DC 0F .0.Y......0.{...
0930 FF 1F E6 1E C2 00 30 11 07 00 19 05 C2 15 30 C3 ......0.......0.
-D
0940 00 FA 80 04 19 00 02 00 E4 80 04 18 01 01 80 F0 ................
0950 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0960 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0970 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0980 C3 5C E7 C3 58 E7 7F 00 20 20 20 20 20 20 20 20 .\..X...
0990 20 20 20 20 20 20 20 20 43 4F 50 59 52 49 47 48         COPYRIGH
09A0 54 20 28 43 29 20 31 39 37 39 2C 20 44 49 47 49 T (C) 1979, DIGI
09B0 54 41 4C 20 52 45 53 45 41 52 43 48 20 20 00 00 TAL RESEARCH  ..
09C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
09D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
09E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
09F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-D
0A00 00 00 00 00 00 00 00 00 08 E4 00 00 5F 0E 02 C3 ............_...
0A10 05 00 C5 CD 8C E4 C1 C9 3E 0D CD 92 E4 3E 0A C3 ........>....>..
0A20 92 E4 3E 20 C3 92 E4 C5 CD 98 E4 E1 7E B7 C8 23 ..> ........~..#
0A30 E5 CD 8C E4 E1 C3 AC E4 0E 0D C3 05 00 5F 0E 0E ............._..
0A40 C3 05 00 CD 05 00 32 EE EB 3C C9 0E 0F C3 C3 E4 ......2..<......
0A50 AF 32 ED EB 11 CD EB C3 CB E4 0E 10 C3 C3 E4 0E .2..............
0A60 11 C3 C3 E4 0E 12 C3 C3 E4 11 CD EB C3 DF E4 0E ................
0A70 13 C3 05 00 CD 05 00 B7 C9 0E 14 C3 F4 E4 11 CD ................
0A80 EB C3 F9 E4 0E 15 C3 F4 E4 0E 16 C3 C3 E4 0E 17 ................
0A90 C3 05 00 1E FF 0E 20 C3 05 00 CD 13 E5 87 87 87 ...... .........
0AA0 87 21 EF EB B6 32 04 00 C9 3A EF EB 32 04 00 C9 .!...2...:..2...
0AB0 FE 61 D8 FE 7B D0 E6 5F C9 3A AB EB B7 CA 96 E5 .a..{..._.:......
-D
0AC0 3A EF EB B7 3E 00 C4 BD E4 11 AC EB CD CB E4 CA :...>...........
0AD0 96 E5 3A B8 EB 3D 32 CC EB 11 AC EB CD F9 E4 C2 ..:..=2.........
0AE0 96 E5 11 07 E4 21 80 00 06 80 CD 42 E8 21 BA EB .....!.....B.!..
0AF0 36 00 23 35 11 AC EB CD DA E4 CA 96 E5 3A EF EB 6.#5.........:..
0B00 B7 C4 BD E4 21 08 E4 CD AC E4 CD C2 E5 CA A7 E5 ....!...........
0B10 CD DD E5 C3 82 E7 CD DD E5 CD 1A E5 0E 0A 11 06 ................
```

```
0B20 E4 CD 05 00 CD 29 E5 21 07 E4 46 23 78 B7 CA BA .....).!..F#x...
0B30 E5 7E CD 30 E5 77 05 C3 AB E5 77 21 08 E4 22 88 .~.0.w....w!.."
0B40 E4 C9 0E 0B CD 05 00 B7 C8 0E 01 CD 05 00 B7 C9 ................
0B50 0E 19 C3 05 00 11 80 00 0E 1A C3 05 00 21 AB EB .............!..
0B60 7E B7 C8 36 00 AF CD BD E4 11 AC EB CD EF E4 3A ~..6...........:
0B70 EF EB C3 BD E4 11 28 E7 21 00 EC 06 06 1A BE C2 ......(.!.......
-D
0B80 CF E7 13 23 05 C2 FD E5 C9 CD 98 E4 2A 8A E4 7E ...#........*..~
0B90 FE 20 CA 22 E6 B7 CA 22 E6 E5 CD 8C E4 E1 23 C3 . ."..."......#.
0BA0 0F E6 3E 3F CD 8C E4 CD 98 E4 CD DD E5 C3 82 E7 ..>?............
0BB0 1A B7 C8 FE 20 DA 09 E6 C8 FE 3D C8 FE 5F C8 FE .... .....=..._..
0BC0 2E C8 FE 3A C8 FE 3B C8 FE 3C C8 FE 3E C8 C9 1A ...:..;..<..>...
0BD0 B7 C8 FE 20 C0 13 C3 4F E6 83 6F D0 24 C9 3E 00 ... ...O..o.$.>.
0BE0 21 CD EB CD 59 E6 E5 E5 AF 32 F0 EB 2A 88 E4 EB !...Y....2..*...
0BF0 CD 4F E6 EB 22 8A E4 EB E1 1A B7 CA 89 E6 DE 40 .O.."...........@
0C00 47 13 1A FE 3A CA 90 E6 1B 3A EF EB 77 C3 96 E6 G...:....:..w...
0C10 78 32 F0 EB 70 13 06 03 CD 30 E6 CA B9 E6 23 FE x2..p....0....#.
0C20 2A C2 A9 E6 36 3F C3 AB E6 77 13 05 C2 98 E6 CD *...6?...w......
0C30 30 E6 CA C0 E6 13 C3 AF E6 23 36 20 05 C2 B9 E6 0........#6 ....
-D
0C40 06 03 FE 2E C2 E9 E6 13 CD 30 E6 CA E9 E6 23 FE .........0....#.
0C50 2A C2 D9 E6 36 3F C3 DB E6 77 13 05 C2 C8 E6 CD *...6?...w......
0C60 30 E6 CA F0 E6 13 C3 DF E6 23 36 20 05 C2 E9 E6 0........#6 ....
0C70 06 03 23 36 00 05 C2 F2 E6 EB 22 88 E4 E1 01 0B ..#6......".....
0C80 00 23 7E FE 3F C2 09 E7 04 0D C2 01 E7 78 B7 C9 .#~.?........x..
0C90 44 49 52 20 45 52 41 20 54 59 50 45 53 41 56 45 DIR ERA TYPESAVE
0CA0 52 45 4E 20 55 53 45 52 00 16 00 00 0F 08 21 10 REN USER......!.
0CB0 E7 0E 00 79 FE 06 D0 11 CE EB 06 04 1A BE C2 4F ...y...........O
0CC0 E7 13 23 05 C2 3C E7 1A FE 20 C2 54 E7 79 C9 23 ..#..<... .T.y.#
0CD0 05 C2 4F E7 0C C3 33 E7 AF 32 07 E4 31 AB EB C5 ..O...3..2..1...
0CE0 79 1F 1F 1F 1F E6 0F 5F CD 15 E5 CD B8 E4 32 AB y......_......2.
0CF0 EB C1 79 E6 0F 32 EF EB CD BD E4 3A 07 E4 B7 C2 ..y..2.....:....
-D
0D00 98 E7 31 AB EB CD 98 E4 CD D0 E5 C6 41 CD 8C E4 ..1.........A...
0D10 3E 3E CD 8C E4 CD 39 E5 11 80 00 CD D8 E5 CD D0 >>....9.........
0D20 E5 32 EF EB CD 5E F6 C4 09 E6 3A F0 EB B7 C2 A5 .2...^....:.....
0D30 EA CD 2E E7 21 C1 E7 5F 16 00 19 19 7E 23 66 6F ....!..._....~#fo
0D40 E9 77 E8 1F E9 5D E9 AD E9 10 EA 8E EA A5 EA 21 .w...].........!
0D50 F3 76 22 00 E4 21 00 E4 E9 01 DF E7 C3 A7 E4 52 .v"..!.........R
0D60 45 41 44 20 45 52 52 4F 52 00 01 F0 E7 C3 A7 E4 EAD ERROR.......
0D70 4E 4F 20 46 49 4C 45 00 CD 5E E6 3A F0 EB B7 C2 NO FILE..^.:....
0D80 09 E6 21 CE EB 01 0B 00 7E FE 20 CA 33 E8 23 D6 ..!.....~. .3.#.
0D90 30 FE 0A D2 09 E6 57 78 E6 E0 C2 09 E6 78 07 07 0.....Wx.....x..
0DA0 07 80 DA 09 E6 80 DA 09 E6 82 DA 09 E6 47 0D C2 .............G..
0DB0 03 E8 C9 7E FE 20 C2 09 E6 23 0D C2 33 E8 78 C9 ...~. ...#..3.x.
-D
0DC0 06 03 7E 12 23 13 05 C2 42 E8 C9 21 80 00 81 CD ..~.#...B..!....
0DD0 59 E6 7E C9 AF 32 CD EB 3A F0 EB B7 C8 3D 21 EF Y.~..2..:....=!.
0DE0 EB BE C8 C3 BD E4 3A F0 EB B7 C8 3D 21 EF EB BE ......:....=!...
0DF0 C3 3A EF EB C3 BD E4 CD 5E E6 CD 54 E8 21 CE EB .:......^..T.!..
0E00 7E FE 20 C2 8F E8 06 0B 36 3F 23 05 C2 88 E8 1E ~. .....6?#.....
0E10 00 D5 CD E9 E4 CC EA E7 CA 1B E9 3A EE EB 0F 0F ...........:....
0E20 0F E6 60 4F 3E 0A CD 4B E8 17 DA 01 E9 D1 7B 1C ..`O>..K......{.
0E30 D5 E6 03 F5 C2 CC E8 CD 98 E4 C5 CD D0 E5 C1 C6 ................
0E40 41 CD 92 E4 3E 3A CD 92 E4 C3 D4 E8 CD A2 E4 3E A...>:.........>
0E50 3A CD 92 E4 CD A2 E4 06 01 78 CD 4B E8 E6 7F FE :.........x.K....
0E60 20 C2 F9 E8 F1 F5 FE 03 C2 F7 E8 3E 09 CD 4B E8  ..........>..K.
0E70 E6 7F FE 20 CA 0E E9 3E 20 CD 92 E4 04 78 FE 0C ... ...> ....x..
-D
0E80 D2 0E E9 FE 09 C2 D9 E8 CD A2 E4 C3 D9 E8 F1 CD ................
0E90 C2 E5 C2 1B E9 CD E4 E4 C3 98 E8 D1 C3 86 EB CD ................
0EA0 5E E6 FE 0B C2 42 E9 01 52 E9 CD A7 E4 CD 39 E5 ^....B..R.....9.
0EB0 21 07 E4 35 C2 82 E7 23 7E FE 59 C2 82 E7 23 22 !..5...#~.Y...#"
0EC0 88 E4 CD 54 E8 11 CD EB CD EF E4 3C CC EA E7 C3 ...T.......<....
0ED0 86 EB 41 4C 4C 20 28 59 2F 4E 29 3F 00 CD 5E E6 ..ALL (Y/N)?..^.
0EE0 C2 09 E6 CD 54 E8 CD D0 E4 CA A7 E9 CD 98 E4 21 ....T..........!
```

```
0EF0 F1 EB 36 FF 21 F1 EB 7E FE 80 DA 87 E9 E5 CD FE ..6.!..~........
0F00 E4 E1 C2 A0 E9 AF 77 34 21 80 00 CD 59 E6 7E FE ......w4!...Y.~.
0F10 1A CA 86 EB CD 8C E4 CD C2 E5 C2 86 EB C3 74 E9 ..............t.
0F20 3D CA 86 EB CD D9 E7 CD 66 E8 C3 09 E6 CD F8 E7 =.......f.......
0F30 F5 CD 5E E6 C2 09 E6 CD 54 E8 11 CD EB D5 CD EF ..^.....T.......
-D
0F40 E4 D1 CD 09 E5 CA FB E9 AF 32 ED EB F1 6F 26 00 .........2...o&.
0F50 29 11 00 01 7C B5 CA F1 E9 2B E5 21 80 00 19 E5 )...|....+.!....
0F60 CD D8 E5 11 CD EB CD 04 E5 D1 E1 C2 FB E9 C3 D4 ................
0F70 E9 11 CD EB CD DA E4 3C C2 01 EA 01 07 EA CD A7 .......<........
0F80 E4 CD D5 E5 C3 86 EB 4E 4F 20 53 50 41 43 45 00 .......NO SPACE.
0F90 CD 5E E6 C2 09 E6 3A F0 EB F5 CD 54 E8 CD E9 E4 .^....:....T....
0FA0 C2 79 EA 21 CD EB 11 DD EB 06 10 CD 42 E8 2A 88 .y.!........B.*.
0FB0 E4 EB CD 4F E6 FE 3D CA 3F EA FE 5F C2 73 EA EB ...O..=.?..._.s..
0FC0 23 22 88 E4 CD 5E E6 C2 73 EA F1 47 21 F0 EB 7E #"...^..s..G!..~
0FD0 B7 CA 59 EA B8 70 C2 73 EA 70 AF 32 CD EB CD E9 ..Y..p.s.p.2....
0FE0 E4 CA 6D EA 11 CD EB CD 0E E5 C3 86 EB CD EA E7 ..m.............
0FF0 C3 86 EB CD 66 E8 C3 09 E6 01 82 EA CD A7 E4 C3 ....f...........
-D
1000 86 EB 46 49 4C 45 20 45 58 49 53 54 53 00 CD F8 ..FILE EXISTS....
1010 E7 FE 10 D2 09 E6 5F 3A CE EB FE 20 CA 09 E6 CD ......_:... ....
1020 15 E5 C3 89 EB CD F5 E5 3A CE EB FE 20 C2 C4 EA ........:... ....
1030 3A F0 EB B7 CA 89 EB 3D 32 EF EB CD 29 E5 CD BD :.......=2...)...
1040 E4 C3 89 EB 11 D6 EB 1A FE 20 C2 09 E6 D5 CD 54 ......... .....T
1050 E8 D1 21 83 EB CD 40 E8 CD D0 E4 CA 6B EB 21 00 ..!...@.....k.!.
1060 01 E5 EB CD D8 E5 11 CD EB CD F9 E4 C2 01 ED E1 ................
1070 11 80 00 19 11 00 E4 7D 93 7C 9A D2 71 EB C3 E1 .......}.|..q...
1080 EA E1 3D C2 71 EB CD 66 E8 CD 5E E6 21 F0 EB E5 ..=.q..f..^.!...
1090 7E 32 CD EB 3E 10 CD 60 E6 E1 7E 32 DD EB AF 32 ~2..>..`..~2...2
10A0 ED EB 11 5C 00 21 CD EB 06 21 CD 42 E8 21 08 E4 ...\.!...!.B.!..
10B0 7E B7 CA 3E EB FE 20 CA 3E EB 23 C3 30 EB 06 00 ~..>.. .>.#.0...
-D
10C0 11 81 00 7E 12 B7 CA 4F EB 04 23 13 C3 43 EB 78 ...~...O..#..C.x
10D0 32 80 00 CD 98 E4 CD D5 E5 CD 1A E5 CD 00 01 31 2..............1
10E0 AB EB CD 29 E5 CD BD E4 C3 82 E7 CD 66 E8 C3 09 ...)........f...
10F0 E6 01 7A EB CD A7 E4 C3 86 EB 42 41 44 20 4C 4F ..z.......BAD LO
1100 41 44 00 43 4F 4D CD 66 E8 CD 5E E6 3A CE EB D6 AD.COM.f..^.:...
1110 20 21 F0 EB B6 C2 09 E6 C3 82 E7 00 00 00 00 00  !..............
1120 00 00 00 00 00 00 00 00 00 00 00 00 00 24 24 24 .............$$$
1130 20 20 20 20 20 53 55 42 00 00 00 00 00 00 00 00      SUB........
1140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-D
1180 00 16 00 00 0F 08 C3 11 EC 99 EC A5 EC AB EC B1 ................
1190 EC EB 22 43 EF EB 7B 32 D6 F9 21 00 00 22 45 EF ..."C..{2..!.."E.
11A0 39 22 0F EF 31 41 EF AF 32 E0 F9 32 DE F9 21 74 9"..1A..2..2..!t
11B0 F9 E5 79 FE 29 D0 4B 21 47 EC 3F 16 00 19 19 5E ..y.).K!G.?.....^
11C0 23 56 2A 43 EF EB E9 03 FA C8 EE 90 ED CE EE 12 #V*C............
11D0 FA 0F FA D4 EE ED EE F3 EE F8 EE E1 ED FE EE 7E ...............~
11E0 F8 83 F8 45 F8 9C F8 A5 F8 AB F8 C8 F8 D7 F8 E0 ...E............
11F0 F8 E6 F8 EC F8 F5 F8 FE F8 04 F9 0A F9 11 F9 2C ...............,
1200 F1 17 F9 1D F9 26 F9 2D F9 41 F9 47 F9 4D F9 0E .....&.-.A.G.M..
1210 F8 53 F9 04 EF 04 EF 9B F9 21 CA EC CD E5 EC FE .S.......!......
1220 03 CA 00 00 C9 21 D5 EC C3 B4 EC 21 E1 EC C3 B4 .....!.....!....
1230 EC 21 DC EC CD E3 EC C3 00 00 42 64 6F 73 20 45 .!........Bdos E
-D
1240 72 72 20 4F 6E 20 20 3A 20 24 42 61 64 20 53 65 rr On  : $Bad Se
1250 63 74 6F 72 24 53 65 6C 65 63 74 24 46 69 6C 65 ctor$Select$File
1260 20 52 2F 4F 24 E5 CD C9 ED 3A 42 EF C6 41 32 C6  R/O$.....:B..A2.
1270 EC 01 BA EC CD D3 ED C1 CD D3 ED 21 0E EF 7E 36 ...........!..~6
1280 00 B7 C0 C3 09 FA CD FB EC CD 14 ED D8 F5 4F CD ..............O.
1290 90 ED F1 C9 FE 0D C3 FE 0A C8 FE 09 C8 FE 08 C8 ................
12A0 FE 20 C9 3A 0E EF B7 C2 45 ED CD 06 FA E6 01 C8 . .:....E.......
12B0 CD 09 FA FE 13 C2 42 ED CD 09 FA FE 03 CA 00 00 ......B.........
```

```
12C0 AF C9 32 0E EF 3E 01 C9 3A 0A EF B7 C2 62 ED C5  ..2..>..:....b..
12D0 CD 23 ED C1 C5 CD 0C FA C1 C5 3A 0D EF B7 C4 0F  .#........:.....
12E0 FA C1 79 21 0C EF FE 7F C8 34 FE 20 D0 35 7E B7  ..y!.....4. .5~.
12F0 C8 79 FE 08 C2 79 ED 35 C9 FE 0A C0 36 00 C9 79  .y...y.5....6..y
-D
1300 CD 14 ED D2 90 ED F5 0E 5E CD 48 ED F1 F6 40 4F  ........^.H...@O
1310 79 FE 09 C2 48 ED 0E 20 CD 48 ED 3A 0C EF E6 07  y...H.. .H.:....
1320 C2 96 ED C9 CD AC ED 0E 20 CD 0C FA 0E 08 C3 0C  ........ .......
1330 FA 0E 23 CD 48 ED CD C9 ED 3A 0C EF 21 0B EF BE  ..#.H....:..!...
1340 D0 0E 20 CD 48 ED C3 B9 ED 0E 0D CD 48 ED 0E 0A  .. .H.......H...
1350 C3 48 ED 0A FE 24 C8 03 C5 4F CD 90 ED C1 C3 D3  .H...$...O......
1360 ED 3A 0C EF 32 0B EF 2A 43 EF 4E 23 E5 06 00 C5  .:..2..*C.N#....
1370 E5 CD FB EC E6 7F E1 C1 FE 0D CA C1 EE FE 0A CA  ................
1380 C1 EE FE 08 C2 16 EE 78 B7 CA EF ED 05 3A 0C EF  .......x.....:..
1390 32 0A EF C3 70 EE FE 7F C2 26 EE 78 B7 CA EF ED  2...p....&.x....
13A0 7E 05 2B C3 A9 EE FE 05 C2 37 EE C5 E5 CD C9 ED  ~.+......7......
13B0 AF 32 0B EF C3 F1 ED FE 10 C2 48 EE E5 21 0D EF  .2........H..!..
-D
13C0 3E 01 96 77 E1 C3 EF ED FE 18 C2 5F EE E1 3A 0B  >..w......_..:.
13D0 EF 21 0C EF BE D2 E1 ED 35 CD A4 ED C3 4E EE FE  .!......5....N..
13E0 15 C2 6B EE CD B1 ED E1 C3 E1 ED FE 12 C2 A6 EE  ..k.............
13F0 C5 CD B1 ED C1 E1 E5 C5 78 B7 CA 8A EE 23 4E 05  ........x....#N.
1400 C5 E5 CD 7F ED E1 C1 C3 78 EE E5 3A 0A EF B7 CA  ........x..:....
1410 F1 ED 21 0C EF 96 32 0A EF CD A4 ED 21 0A EF 35  ..!...2.....!..5
1420 C2 99 EE C3 F1 ED 23 77 04 C5 E5 4F CD 7F ED E1  ......#w...O....
1430 C1 7E FE 03 78 C2 BD EE FE 01 CA 00 00 B9 DA EF  .~..x...........
1440 ED E1 70 0E 0D C3 48 ED CD 06 ED C3 01 EF CD 15  ..p...H.........
1450 FA C3 01 EF 79 3C CA E0 EE 3C CA 06 FA C3 0C FA  ....y<...<......
1460 CD 06 FA B7 CA 91 F9 CD 09 FA C3 01 EF 3A 03 00  .............:..
1470 C3 01 EF 21 03 00 71 C9 EB 4D 44 C3 D3 ED CD 23  ...!..q..MD....#
-D
1480 ED 32 45 EF C9 3E 01 C3 01 EF 00 00 00 00 00 00  .2E..>..........
1490 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
14A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
14B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
14C0 00 00 00 00 00 00 00 21 0B EC 5E 23 56 EB E9 0C  .......!..^#V...
14D0 0D C8 1A 77 13 23 C3 50 EF 3A 42 EF 4F CD 1B FA  ...w.#.P.:B.O...
14E0 7C B5 C8 5E 23 56 23 22 B3 F9 23 23 22 B5 F9 23  |..^#V#"..##"..#
14F0 23 22 B7 F9 23 23 EB 22 D0 F9 21 B9 F9 0E 08 CD  #"..##."..!.....
1500 4F EF 2A BB F9 EB 21 C1 F9 0E 0F CD 4F EF 2A C6  O.*...!.....O.*.
1510 F9 7C 21 DD F9 36 FF B7 CA 9D EF 36 00 3E FF B7  .|!..6.....6.>..
1520 C9 CD 18 FA AF 2A B5 F9 77 23 77 2A B7 F9 77 23  .....*..w#w*..w#
1530 77 C9 CD 27 FA C3 BB EF CD 2A FA B7 C8 21 09 EC  w..'.....*...!..
-D
1540 C3 4A EF 2A EA F9 0E 02 CD EA F0 22 E5 F9 22 EC  .J.*.......".."."
1550 F9 21 E5 F9 4E 23 46 2A B7 F9 5E 23 56 2A B5 F9  .!..N#F*..^#V*..
1560 7E 23 66 6F 79 93 78 9A D2 FA EF E5 2A C1 F9 7B  ~#foy.x.....*..{
1570 95 5F 7A 9C 37 E1 2B C3 E4 EF E5 2A C1 F9 19 DA  ._z.W.+....*....
1580 0F F0 79 95 78 9C DA 0F F0 EB E1 23 C3 FA EF E1  ..y.x......#....
1590 C5 D5 E5 EB 2A CE F9 19 44 4D CD 1E FA D1 2A B5  ....*...DM....*.
15A0 F9 73 23 72 D1 2A B7 F9 73 23 72 C1 79 93 4F 78  .s#r.*..s#r.y.Ox
15B0 9A 47 2A D0 F9 EB CD 30 FA 4D 44 C3 21 FA 21 C3  .G*....0.MD.!.!.
15C0 F9 4E 3A E3 F9 B7 1F 0D C2 45 F0 47 3E 08 96 4F  .N:......E.G>..O
15D0 3A E2 F9 0D CA 5C F0 B7 17 C3 53 F0 80 C9 2A 43  :....\....S...*C
15E0 EF 11 10 00 19 09 3A DD F9 B7 CA 71 F0 6E 26 00  ......:....q.n&.
15F0 C9 09 5E 23 56 EB C9 CD 3E F0 4F 06 00 CD 5E F0  ..^#V...>.O...^.
-D
1600 22 E5 F9 C9 2A E5 F9 7D B4 C9 3A C3 F9 2A E5 F9  "...*..}..:..*..
1610 29 3D C2 90 F0 22 E7 F9 3A C4 F9 4F 3A E3 F9 A1  )=..."..:..O:...
1620 B5 6F 22 E5 F9 C9 2A 43 EF 11 0C 00 19 C9 2A 43  .o"...*C......*C
1630 EF 11 0F 00 19 EB 21 11 00 19 C9 CD AE F0 7E 32  ......!.......~2
1640 E3 F9 EB 7E 32 E1 F9 CD A6 F0 3A C5 F9 A6 32 E2  ...~2.....:...2.
1650 F9 C9 CD AE F0 3A D5 F9 FE 02 C2 DE F0 AF 4F 3A  .....:........O:
1660 E3 F9 81 77 EB 3A E1 F9 77 C9 0C 0D C8 7C B7 1F  ...w.:..w....|..
1670 67 7D 1F 6F C3 EB F0 0E 80 2A B9 F9 AF 86 23 0D  g}.o.....*....#.
1680 C2 FD F0 C9 0C 0D C8 29 C3 05 F1 C5 3A 42 EF 4F  .......).....:B.O
```

```
1690 21 01 00 CD 04 F1 C1 79 B5 6F 78 B4 67 C9 2A AD !.....y.ox.g.*.
16A0 F9 3A 42 EF 4F CD EA F0 7D E6 01 C9 21 AD F9 4E .:B.O...}....!..N
16B0 23 46 CD 0B F1 22 AD F9 2A C8 F9 23 EB 2A B3 F9 #F...".."*..#.*..
-D
16C0 73 23 72 C9 CD 5E F1 11 09 00 19 7E 17 D0 21 0F s#r..^.....~..!.
16D0 EC C3 4A EF CD 1E F1 C8 21 0D EC C3 4A EF 2A B9 ..J.....!...J.*.
16E0 F9 3A E9 F9 85 6F D0 24 C9 2A 43 EF 11 0E 00 19 .:...o.$.*C.....
16F0 7E C9 CD 69 F1 36 00 C9 CD 69 F1 F6 80 77 C9 2A ~..i.6...i...w.*
1700 EA F9 EB 2A B3 F9 7B 96 23 7A 9E C9 CD 7F F1 D8 ...*..{.#z......
1710 13 72 2B 73 C9 7B 95 6F 7A 9C 67 C9 0E FF 2A EC .r+s.{.oz.g...*.
1720 F9 EB 2A CC F9 CD 95 F1 D0 C5 CD F7 F0 2A BD F9 ..*..........*..
1730 EB 2A EC F9 19 C1 0C CA C4 F1 BE C8 CD 7F F1 D0 .*..............
1740 CD 2C F1 C9 77 C9 CD 9C F1 CD E0 F1 0E 01 CD B8 .,..w...........
1750 EF C3 DA F1 CD E0 F1 CD B2 EF 21 B1 F9 C3 E3 F1 ..........!.....
1760 21 B9 F9 4E 23 46 C3 24 FA 2A B9 F9 EB 2A B1 F9 !..N#F.$.*...*..
1770 0E 80 C3 4F EF 21 EA F9 7E 23 BE C0 3C C9 21 FF ...O.!..~#..<.!.
-D
1780 FF 22 EA F9 C9 2A C8 F9 EB 2A EA F9 23 22 EA F9 ."...*...*..#"..
1790 CD 95 F1 D2 19 F2 C3 FE F1 3A EA F9 E6 03 06 05 .........:......
17A0 87 05 C2 20 F2 32 E9 F9 B7 C0 C5 CD C3 EF CD D4 ... .2..........
17B0 F1 C1 C3 9E F1 79 E6 07 3C 5F 57 79 0F 0F 0F E6 .....y..<_Wy....
17C0 1F 4F 78 87 87 87 87 87 B1 4F 78 0F 0F 0F E6 1F .Ox......Ox.....
17D0 47 2A BF F9 09 7E 07 1D C2 56 F2 C9 D5 CD 35 F2 G*...~...V....5.
17E0 E6 FE C1 B1 0F 15 C2 64 F2 77 C9 CD 5E F1 11 10 .......d.w..^...
17F0 00 19 C5 0E 11 D1 0D C8 D5 3A DD F9 B7 CA 88 F2 .........:......
1800 C5 E5 4E 06 00 C3 8E F2 0D C5 4E 23 46 E5 79 B0 ..N.......N#F.y.
1810 CA 9D F2 2A C6 F9 7D 91 7C 98 D4 5C F2 E1 23 C1 ...*..}.|..\..#.
1820 C3 75 F2 2A C6 F9 0E 03 CD EA F0 23 44 4D 2A BF .u.*.......#DM*.
1830 F9 36 00 23 0B 78 B1 C2 B1 F2 2A CA F9 EB 2A BF .6.#.x....*...*.
-D
1840 F9 73 23 72 CD A1 EF 2A B3 F9 36 03 23 36 00 CD .s#r...*..6.#6..
1850 FE F1 0E FF CD 05 F2 CD F3 F1 C8 CD 5E F1 3E E5 ............^.>.
1860 BE CA D2 F2 3A 41 EF BE C2 F6 F2 23 7E D6 24 C2 ....:A.....#~.$.
1870 F6 F2 3D 32 45 EF 0E 01 CD 6B F2 CD 8C F1 C3 D2 ..=2E....k......
1880 F2 3A D4 F9 C3 01 EF C5 F5 3A C5 F9 2F 47 79 A0 .:.......:../Gy.
1890 4F F1 A0 91 E6 1F C1 C9 3E FF 32 D4 F9 21 D3 F9 O.......>.2..!..
18A0 71 2A 43 EF 22 D9 F9 CD FE F1 CD A1 EF 0E 00 CD q*C."...........
18B0 05 F2 CD F5 F1 CA 94 F3 2A D9 F9 EB 1A FE E5 CA ........*.......
18C0 4A F3 D5 CD 7F F1 D1 D2 94 F3 CD 5E F1 3A D8 F9 J..........^.:..
18D0 4F 06 00 79 B7 CA 83 F3 1A FE 3F CA 7C F3 78 FE O..y......?.|.x.
18E0 0D CA 7C F3 FE 0C 1A CA 73 F3 96 E6 7F C2 2D F3 ..|.....s.....-.
18F0 C3 7C F3 C5 4E CD 07 F3 C1 C2 2D F3 13 23 04 0D .|..N.....-..#..
-D
1900 C3 53 F3 3A EA F9 E6 03 32 45 EF 21 D4 F9 7E 17 .S.:....2E.!..~.
1910 D0 AF 77 C9 CD FE F1 3E FF C3 01 EF CD 54 F1 0E ..w....>.....T..
1920 0C CD 18 F3 CD F5 F1 C8 CD 44 F1 CD 5E F1 36 E5 .........D..^.6.
1930 0E 00 CD 6B F2 CD C6 F1 CD 2D F3 C3 A4 F3 50 59 ...k.....-....PY
1940 79 B0 CA D1 F3 0B D5 C5 CD 35 F2 1F D2 EC F3 C1 y........5......
1950 D1 2A C6 F9 7B 95 7A 9C D2 F4 F3 13 C5 D5 42 4B .*..{.z.......BK
1960 CD 35 F2 1F D2 EC F3 D1 C1 C3 C0 F3 17 3C CD 64 .5...........<.d
1970 F2 E1 D1 C9 79 B0 C2 C0 F3 21 00 00 C9 0E 00 1E ....y....!......
1980 20 D5 06 00 2A 43 EF 09 EB CD 5E F1 C1 CD 4F EF  ...*C....^...O.
1990 CD C3 EF C3 C6 F1 CD 54 F1 0E 0C CD 18 F3 2A 43 .......T......*C
19A0 EF 7E 11 10 00 19 77 CD F5 F1 C8 CD 44 F1 0E 10 .~....w.....D...
19B0 1E 0C CD 01 F4 CD 2D F3 C3 27 F4 0E 0C CD 18 F3 ......-..'......
-D
19C0 CD F5 F1 C8 0E 00 1E 0C CD 01 F4 CD 2D F3 C3 40 ............-..@
19D0 F4 0E 0F CD 18 F3 CD F5 F1 C8 CD A6 F0 7E F5 E5 .............~..
19E0 CD 5E F1 EB 2A 43 EF 0E 20 D5 CD 4F EF CD 78 F1 .^..*C.. ..O..x.
19F0 D1 21 0C 00 19 4E 21 0F 00 19 46 E1 F1 77 79 BE .!...N!...F..wy.
1A00 78 CA 8B F4 3E 00 DA 8B F4 3E 80 2A 43 EF 11 0F x....>....>.*C...
1A10 00 19 77 C9 7E 23 B6 2B C0 1A 77 13 23 1A 77 1B ..w.~#.+..w.#.w.
1A20 2B C9 AF 32 45 EF 32 EA F9 32 EB F9 CD 1E F1 C0 +..2E.2..2......
1A30 CD 69 F1 E6 80 C0 0E 0F CD 18 F3 CD F5 F1 C8 01 .i..............
1A40 10 00 CD 5E F1 09 EB 2A 43 EF 09 0E 10 3A DD F9 ...^...*C....:..
1A50 B7 CA F8 F4 7E B7 1A C2 DB F4 77 B7 C2 E1 F4 7E ....~.....w....~
```

```
1A60 12 BE C2 1F F5 C3 FD F4 CD 94 F4 EB CD 94 F4 EB ................
1A70 1A BE C2 1F F5 13 23 1A BE C2 1F F5 0D 13 23 0D ......#.......#.
-D
1A80 C2 CD F4 01 EC FF 09 EB 09 1A BE DA 17 F5 77 01 ..............w.
1A90 03 00 09 EB 09 7E 12 3E FF 32 D2 F9 C3 10 F4 21 .....~.>.2.....!
1AA0 45 EF 35 C9 CD 54 F1 2A 43 EF E5 21 AC F9 22 43 E.5..T.*C..!.."C
1AB0 EF 0E 01 CD 18 F3 CD F5 F1 E1 22 43 EF C8 EB 21 ..........."C...!
1AC0 0F 00 19 0E 11 AF 77 23 0D C2 46 F5 21 0D 00 19 ......w#..F.!...
1AD0 77 CD 3C F1 CD FD F3 C3 78 F1 AF 32 D2 F9 CD A2 w.......x..2....
1AE0 F4 CD F5 F1 C8 2A 43 EF 01 0C 00 09 7E 3C E6 1F .....*C.....~<..
1AF0 77 CA 83 F5 47 3A C5 F9 A0 21 D2 F9 A6 CA 8E F5 w...G:...!......
1B00 C3 AC F5 01 02 00 09 34 7E E6 0F CA B6 F5 0E 0F .......4~.......
1B10 CD 18 F3 CD F5 F1 C2 AC F5 3A D3 F9 3C CA B6 F5 .........:..<...
1B20 CD 24 F5 CD F5 F1 CA B6 F5 C3 AF F5 CD 5A F4 CD .$..........Z..
1B30 BB F0 AF C3 01 EF CD 05 EF C3 78 F1 3E 01 32 D5 ..........x.>.2.
-D
1B40 F9 3E FF 32 D3 F9 CD BB F0 3A E3 F9 21 E1 F9 BE .>.2.....:..!...
1B50 DA E6 F5 FE 80 C2 FB F5 CD 5A F5 AF 32 E3 F9 3A .........Z..2..:
1B60 45 EF B7 C2 FB F5 CD 77 F0 CD 84 F0 CA FB F5 CD E.......w........
1B70 8A F0 CD D1 EF CD B2 EF C3 D2 F0 C3 05 EF 3E 01 ..............>.
1B80 32 D5 F9 3E 00 32 D3 F9 CD 54 F1 2A 43 EF CD 47 2..>.2...T.*C..G
1B90 F1 CD BB F0 3A E3 F9 FE 80 D2 05 EF CD 77 F0 CD ....:.........w..
1BA0 84 F0 0E 00 C2 6E F6 CD 3E F0 32 D7 F9 01 00 00 .....n..>.2.....
1BB0 B7 CA 3B F6 4F 0B CD 5E F0 44 4D CD BE F3 7D B4 ..;.O..^.DM...}.
1BC0 C2 48 F6 3E 02 C3 01 EF 22 E5 F9 EB 2A 43 EF 01 .H.>....".....*C..
1BD0 10 00 09 3A DD F9 B7 3A D7 F9 CA 64 F6 CD 64 F1 ...:...:...d..d.
1BE0 73 C3 6C F6 4F 06 00 09 09 73 23 72 0E 02 3A 45 s.l.O....s#r..:E
1BF0 EF B7 C0 C5 CD 8A F0 3A D5 F9 3D 3D C2 BB F6 C1 .......:..==....
-D
1C00 C5 79 3D 3D C2 BB F6 E5 2A B9 F9 57 77 23 14 F2 .y==....*..Ww#..
1C10 8C F6 CD E0 F1 2A E7 F9 0E 02 22 E5 F9 C5 CD D1 .....*....".....
1C20 EF C1 CD B8 EF 2A E5 F9 0E 00 3A C4 F9 47 A5 B8 .....*....:..G..
1C30 23 C2 9A F6 E1 22 E5 F9 CD DA F1 CD D1 EF C1 C5 #....".........
1C40 CD B8 EF C1 3A E3 F9 21 E1 F9 BE DA D2 F6 77 34 ....:..!......w4
1C50 0E 02 0D 0D C2 DF F6 F5 CD 69 F1 E6 7F 77 F1 FE .........i...w..
1C60 7F C2 00 F7 3A D5 F9 FE 01 C2 00 F7 CD D2 F0 CD ....:...........
1C70 5A F3 21 45 EF 7E B7 C2 FE F6 3D 32 E3 F9 36 00 Z.!E.~....=2..6.
1C80 C3 D2 F0 AF 32 D5 F9 C5 2A 43 EF EB 21 21 00 19 ....2...*C..!!..
1C90 7E E6 7F F5 7E 17 23 7E 17 E6 1F 4F 7E 1F 1F 1F ~...~.#~...O~...
1CA0 1F E6 0F 47 F1 23 6E 2C 2D 2E 06 C2 8B F7 21 20 ...G.#n,-.....!
1CB0 00 19 77 21 0C 00 19 79 96 C2 47 F7 21 0E 00 19 ..w!...y..G.!...
-D
1CC0 78 96 E6 7F CA 7F F7 C5 D5 CD A2 F4 D1 C1 2E 03 x...............
1CD0 3A 45 EF 3C CA 84 F7 21 0C 00 19 71 21 0E 00 19 :E.<...!...q!...
1CE0 70 CD 51 F4 3A 45 EF 3C C2 7F F7 C1 C5 2E 04 0C p.Q.:E.<........
1CF0 CA 84 F7 CD 24 F5 2E 05 3A 45 EF 3C CA 84 F7 C1 ....$...:E.<....
1D00 AF C3 01 EF E5 CD 69 F1 36 C0 E1 C1 7D 32 45 EF ......i.6...}2E.
1D10 C3 78 F1 0E FF CD 03 F7 CC C1 F3 C9 0E 00 CD 03 .x..............
1D20 F7 CC 03 F6 C9 EB 19 4E 06 00 21 0C 00 19 7E 0F .......N..!...~.
1D30 E6 80 81 4F 3E 00 88 47 7E 0F E6 0F 80 47 21 0E ...O>..G~....G!.
1D40 00 19 7E 87 87 87 87 F5 80 47 F5 E1 7D E1 B5 E6 ..~......G..}...
1D50 01 C9 0E 0C CD 18 F3 2A 43 EF 11 21 00 19 E5 72 .......*C..!...r
1D60 23 72 23 72 CD F5 F1 CA 0C F8 CD 5E F1 11 0F 00 #r#r.......^....
1D70 CD A5 F7 E1 E5 5F 79 96 23 78 9E 23 7B 9E DA 06 ....._y.#x.#{...
-D
1D80 F8 73 2B 70 2B 71 CD 2D F3 C3 E4 F7 E1 C9 2A 43 .s+p+q.-......*C
1D90 EF 11 20 00 CD A5 F7 21 21 00 19 71 23 70 23 77 .. ....!!..q#p#w
1DA0 C9 2A AF F9 3A 42 EF 4F CD EA F0 E5 EB CD 59 EF .*..:B.O......Y.
1DB0 E1 CC 47 EF 7D 1F D8 2A AF F9 4D 44 CD 0B F1 22 ..G.}..*..MD...."
1DC0 AF F9 C3 A3 F2 3A D6 F9 21 42 EF BE C8 77 C3 21 .....:..!B...w.!
1DD0 F8 3E FF 32 DE F9 2A 43 EF 7E E6 1F 3D 32 D6 F9 .>.2..*C.~..=2..
1DE0 FE 1E D2 75 F8 3A 42 EF 32 DF F9 7E 32 E0 F9 E6 ...u.:B.2..~2...
1DF0 E0 77 CD 45 F8 3A 41 EF 2A 43 EF B6 77 C9 3E 22 .w.E.:A.*C..w.>"
1E00 C3 01 EF 21 00 00 22 AD F9 22 AF F9 AF 32 42 EF ...!.."..".....2B.
1E10 21 80 00 22 B1 F9 CD DA F1 C3 21 F8 CD 72 F1 CD !.."......!..r..
1E20 51 F8 C3 51 F4 CD 51 F8 C3 A2 F4 0E 00 EB 7E FE Q..Q..Q.......~.
```

```
1E30 3F CA C2 F8 CD A6 F0 7E FE 3F C4 72 F1 CD 51 F8 ?......~.?.r..Q.
-D
1E40 0E 0F CD 18 F3 C3 E9 F1 2A D9 F9 22 43 EF CD 51 ........*.."C..Q
1E50 F8 CD 2D F3 C3 E9 F1 CD 51 F8 CD 9C F3 C3 01 F3 ..-.....Q.......
1E60 CD 51 F8 C3 BC F5 CD 51 F8 C3 FE F5 CD 72 F1 CD .Q.....Q.....r..
1E70 51 F8 C3 24 F5 CD 51 F8 CD 16 F4 C3 01 F3 2A AF Q..$..Q.......*.
1E80 F9 C3 29 F9 3A 42 EF C3 01 EF EB 22 B1 F9 C3 DA ..).:B.....".....
1E90 F1 2A BF F9 C3 29 F9 2A AD F9 C3 29 F9 CD 51 F8 .*...).*...)..Q.
1EA0 CD 3B F4 C3 01 F3 2A BB F9 22 45 EF C9 3A D6 F9 .;....*.."E..:..
1EB0 FE FF C2 3B F9 3A 41 EF C3 01 EF E6 1F 32 41 EF ...;.:A......2A.
1EC0 C9 CD 51 F8 C3 93 F7 CD 51 F8 C3 9C F7 CD 51 F8 ..Q.....Q.....Q.
1ED0 C3 D2 F7 2A 43 EF 7D 2F 5F 7C 2F 2A AF F9 A4 57 ...*C.)/_|/*...W
1EE0 7D A3 5F 2A AD F9 EB 22 AF F9 7D A3 6F 7C A2 67 )._*..."..).o|.g
1EF0 22 AD F9 C9 3A DE F9 B7 CA 91 F9 2A 43 EF 36 00 "...:......*C.6.
-D
1F00 3A E0 F9 B7 CA 91 F9 77 3A DF F9 32 D6 F9 CD 45 :......w:..2...E
1F10 F8 2A 0F EF F9 2A 45 EF 7D 44 C9 CD 51 F8 3E 02 .*...*E.)D..Q.>.
1F20 32 D5 F9 0E 00 CD 07 F7 CC 03 F6 C9 E5 00 00 00 2...............
1F30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1F40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1F50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1F60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1F70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1F80 C3 B3 FA C3 C3 FA C3 61 FB C3 64 FB C3 6A FB C3 .......a..d..j..
1F90 6D FB C3 72 FB C3 75 FB C3 78 FB C3 7D FB C3 A7 m..r..u..x..)...
1FA0 FB C3 AC FB C3 BB FB C3 C1 FB C3 CA FB C3 70 FB ..............p.
1FB0 C3 B1 FB 82 FA 00 00 00 00 00 00 6E FC 73 FA 0D ...........n.s..
-D
1FC0 FD EE FC 82 FA 00 00 00 00 00 00 6E FC 73 FA 3C ...........n.s.<
1FD0 FD 1D FD 82 FA 00 00 00 00 00 00 6E FC 73 FA 6B ...........n.s.k
1FE0 FD 4C FD 82 FA 00 00 00 00 00 00 6E FC 73 FA 9A .L.........n.s..
1FF0 FD 7B FD 1A 00 03 07 00 F2 00 3F 00 C0 00 10 00 .{........?.....
2000 02 00 01 07 0D 13 19 05 0B 11 17 03 09 0F 15 02 ................
2010 08 0E 14 1A 06 0C 12 18 04 0A 10 16 0D 0A 0A 36 ...............6
2020 34 6B 20 43 50 2F 4D 20 76 65 72 73 20 32 2E 32 4k CP/M vers 2.2
2030 0D 0A 00 31 00 01 21 9C FA CD D3 FB AF 32 04 00 ...1..!......2..
2040 C3 0F FB 31 80 00 0E 0A C5 01 00 E4 CD DB FB 0E ...1............
2050 00 CD 7D FB 0E 00 CD A7 FB 0E 02 CD AC FB C1 06 ..}.............
2060 2C C5 CD C1 FB C2 49 FB 2A 6C FC 11 80 00 19 44 ,.....I.*l.....D
2070 4D CD BB FB 3A 6B FC FE 1A DA 05 FB 3A 6A FC 3C M...:k......:j.<
-D
2080 4F CD A7 FB AF 3C 4F CD AC FB C1 05 C2 E1 FA F3 O....<O.........
2090 3E 12 D3 FD AF D3 FC 3E 7E D3 FC AF D3 F3 01 80 >.......>~.......
20A0 00 CD BB FB 3E C3 32 00 00 21 03 FA 22 01 00 32 ....>.2..!.."...2
20B0 05 00 21 06 EC 22 06 00 32 38 00 21 00 F8 22 39 ..!.."..28.!..."9
20C0 00 3A 04 00 4F FB C3 00 E4 C1 0D CA 52 FB C5 C3 .:..O.......R...
20D0 C9 FA 21 5B FB CD D3 FB C3 0F FF 3F 62 6F 6F 74 ..![........?boot
20E0 00 C3 12 F8 CD 03 F8 E6 7F C9 C3 09 F8 C3 0F F8 ................
20F0 AF C9 C3 0C F8 C3 06 F8 0E 00 C3 A7 FB 21 00 00 .............!..
2100 79 FE 04 D0 E6 02 32 66 FC 79 E6 01 B7 CA 92 FB y.....2f.y......
2110 3E 30 47 21 68 FC 7E E6 CF B0 77 69 26 00 29 29 >0G!h.~...wi&.))
2120 29 29 11 33 FA 19 C9 21 6A FC 71 C9 21 6B FC 71 )).3...!j.q.!k.q
2130 C9 06 00 EB 09 7E 32 6B FC 6F C9 69 60 22 6C FC .....~2k.o.i`"l.
-D
2140 C9 0E 04 CD E0 FB CD F0 FB C9 0E 06 CD E0 FB CD ................
2150 F0 FB C9 7E B7 C8 E5 4F CD 6A FB E1 23 C3 D3 FB ...~...O.j..#...
2160 21 68 FC 7E E6 F8 B1 77 E6 20 21 6B FC B6 77 C9 !h.~...w. !k..w.
2170 0E 0A CD 3F FC CD 4C FC 3A 66 FC B7 3E 67 06 FC ...?..L.:f..>g..
2180 C2 0B FC D3 79 78 D3 7A C3 10 FC D3 89 78 D3 8A ....yx.z.....x..
2190 CD 59 FC E6 04 CA 10 FC CD 3F FC FE 02 CA 32 FC .Y.......?....2.
21A0 B7 C2 38 FC CD 4C FC 17 DA 32 FC 1F E6 FE C2 38 ..8..L...2.....8
21B0 FC C9 CD 4C FC C3 38 FC 0D C2 F2 FB 3E 01 C9 3A ...L..8.....>..:
21C0 66 FC B7 C2 49 FC DB 79 C9 DB 89 C9 3A 66 FC B7 f...I..y....:f..
21D0 C2 56 FC DB 7B C9 DB 8B C9 3A 66 FC B7 C2 63 FC .V..{....:f...c.
21E0 DB 78 C9 DB 88 C9 00 80 04 01 02 01 80 00 00 00 .x..............
21F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

```
-D
2200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2290 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
22A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
22B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-D
22C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
22D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
22E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
22F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2300 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2310 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2320 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2330 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2340 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2350 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2360 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2370 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-F100,3000,00
-
-
-I2AB10S64.HEX
-R2380
NEXT  PC
2300 0000
-D940
0940 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0950 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0960 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0970 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0980 00 00 00 00 00 00 00 00 0D 0A 54 61 72 62 65 6C ..........Tarbel
0990 6C 20 36 34 4B 20 43 50 4D 20 32 2E 32 0D 0A 41 l 64K CPM 2.2..A
09A0 75 74 6F 2D 53 65 6C 65 63 74 20 76 65 72 20 6F uto-Select ver o
09B0 66 20 31 31 2D 34 2D 38 30 00 31 80 00 3E 03 D3 f 11-4-80.1..>..
09C0 00 D3 02 3E 11 D3 00 D3 02 21 B4 FA 22 01 FA C3 ...>.....!..".."...
09D0 C0 F9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
09E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
09F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-
-
-
-D1F00
1F00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1F10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1F20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1F30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1F40 AF 32 03 00 32 04 00 06 11 21 27 FD 77 23 05 C2 .2..2....!'.w#..
1F50 CC F9 DB 01 21 08 E4 7E 23 B7 CA 1D FB 4F CD 5C ....!..~#....O.\
1F60 FB C3 D7 F9 00 00 00 00 00 00 00 00 00 00 00 00 ................
1F70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1F80 C3 3A E4 C3 EF FA C3 44 FB C3 51 FB C3 5C FB C3 .:.....D..Q..\..
1F90 19 FD C3 13 FD C3 13 FD C3 F4 FB C3 67 FB C3 FA ............g...
1FA0 FB C3 68 FC C3 78 FC C3 9D FC C3 F2 FC C3 0C FD ..h..x..........
1FB0 C3 6D FC 97 FA 00 00 00 00 00 00 38 FD 76 FA D7 .m.........8.v..
-D
1FC0 FD B8 FD 97 FA 00 00 00 00 00 00 38 FD 76 FA 0E ...........8.v..
1FD0 FE EF FD 97 FA 00 00 00 00 00 00 38 FD 76 FA 45 ...........8.v.E
```

2.2 Auto bios SD

```
1FF0 FE 5D FE 00 97 FA 1A 00 03 07 00 F2 00 3F 00 C0 .]...........?..
2000 00 10 00 02 00 01 00 00 33 00 04 0F 00 ED 00 5F ........3......_
2010 00 C0 00 18 00 02 00 01 07 0D 13 19 05 08 11 17 ................
2020 03 09 0F 15 02 08 0E 14 1A 06 0C 12 18 04 0A 10 ................
2030 16 32 34 FD 3A 24 FD D3 FA 2A 25 FD AF D3 E8 79 .24.:$...*%....y
2040 D3 F8 7B 3D D3 E1 7A D3 E1 7D D3 E0 7C D3 E0 3E ..{=..z..}..|..>
2050 41 D3 E8 CD 7E FC B0 D3 F8 D5 7A E6 3F 57 19 D1 A...~.....z.?W..
2060 3E 20 3D C2 E2 FA DB FD 07 DA E6 FA DB F8 C9 31 > =.............1
2070 80 00 0E 00 CD 67 FB CD F4 FB 21 00 00 22 2F FD .....g....!.."/.
-D
2080 22 31 FD 06 11 0E 02 21 00 E4 22 25 FD CD 68 FC "1.....!.."%..h.
2090 C5 CD 9D FC C1 C2 3E FB 0C 05 C2 0A FB 3E C3 32 ......>......>.2
20A0 00 00 21 03 FA 22 01 00 32 05 00 21 06 EC 22 06 ..!.."..2..!.."..
20B0 00 21 80 00 22 25 FD 3A 04 00 4F C3 00 E4 CD B7 .!.."%.:..O.....
20C0 FC C3 EF FA CD 4C FB 3E 00 C8 2F C9 DB 00 E6 01 .....L.>../.....
20D0 C9 CD 4C FB CA 51 FB DB 01 E6 7F C9 DB 00 E6 02 ..L..Q..........
20E0 CA 5C FB 79 D3 01 C9 21 00 00 79 FE 04 D0 21 27 .\.y...!..y...!'
20F0 FD 7E 5F 16 00 21 2B FD E5 19 DB F9 77 79 5F E1 .~_..!+.....wy_.
2100 19 7E D3 F9 79 32 27 FD 87 87 87 87 32 36 FD 21 .~..y2'.....26.!
2110 2F FD 06 00 09 7E B7 FA C0 FB E5 3A 36 FD D3 FC /....~.....:6...
2120 3E 01 32 24 FD CD F4 FB 2A 25 FD E5 21 94 FE 22 >.2$....*%..!.."
2130 25 FD CD 9D FC E1 22 25 FD E1 3A 12 FF F6 80 77 %.....'%..:....w
-D
2140 01 12 00 E6 12 B7 21 73 FA CA CD FB 09 EB 1A 13 ......!s........
2150 32 33 FD D5 CD E7 FB D1 01 08 02 1A 77 13 23 05 23..........w.#.
2160 C2 DB FB 09 73 23 72 2A 27 FD 11 33 FA 29 29 29 ....s#r*'..3.)))
2170 29 19 AF C9 0E 00 3E 01 D3 F8 2A 36 FD 7C E6 B7 ).....>...*6.|..
2180 67 C3 0F FC BC 3E FF C2 0B FC 2F 32 29 FD C9 7D g....>..../2)..}
2190 32 37 FD CD 04 FC 3A 33 FD 0F D2 2B FC 79 FE 01 27....:3...+.y..
21A0 DA 2B FC 3A 37 FD F6 08 C3 30 FC 3A 37 FD E6 F7 .+.:7....0.:7...
21B0 32 37 FD D3 FC 79 32 23 FD C5 47 3E 0A 32 35 FD 27...y2#..G>.25.
21C0 DB F9 B8 C2 48 FC C1 C9 78 D3 FB 3E 1D D3 F8 CD ....H...x..>....
21D0 E6 FA E6 91 CA 46 FC 3A 35 FD 3D C2 3D FC C1 C5 .....F.:5.=.=....
21E0 CD B7 FC C1 79 C3 39 FC 79 32 24 FD C9 69 2C 7A ....y.9.y2$..i,z
21F0 B3 C8 EB 09 6E 26 00 C9 60 69 22 23 FD C9 3A 29 ....n&..`i"#..:)
-D
2200 FD B7 CA 94 FC 2F 32 29 FD DB F9 D3 FB 3E 15 D3 ...../2).....>..
2210 F8 CD E6 FA DB F8 E6 20 3E 04 C3 97 C9 3E 0A 01 ....... >....>..
2220 D0 80 11 80 40 CD B1 FA E6 9D C8 CD C5 FC C2 9F ....@...........
2230 FC CD B7 FC C3 9D FC 0E 65 CD 5C FB CD 51 FB FE ........e.\..Q..
2240 03 C0 C3 EF FA E6 10 C2 CF FC 3A 34 FD 3D C9 21 ..........:4.=.!
2250 F9 F9 01 D0 C4 11 06 40 CD BC FA B7 CA E5 FC CD .......@........
2260 F4 FB C3 E9 FC DB FA D3 F9 3A 23 FD CD 39 FC C3 .........:#..9..
2270 CA FC 3E 0A 01 D0 A0 11 30 80 CD B1 FA E6 FD C8 ..>.....0.......
2280 CD C5 FC C2 F4 FC CD B7 FC C3 F2 FC CD 14 FD 3E ...............>
2290 00 C8 2F C9 DB 02 E6 02 C9 CD 14 FD CA 19 FD 79 ../............y
22A0 D3 03 C9 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
22B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-D
22C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
22D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
22E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
22F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2300 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2310 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2320 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2330 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2340 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2350 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2360 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
2370 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-
-
-F100,3000,00
-I2DBOOT64.HEX 2.2 S.D. Boot
-R200
```

```
NEXT  PC
2300 0000
-^]#
 D900
0900 1E 0A 31 00 01 21 00 E4 16 33 0E 02 06 04 CD 29 ..1..!...3.....)
0910 00 15 CA 00 FA 06 00 0C 79 FE 1B DA 0E 00 3E 5B ........y.....>[
0920 D3 F8 DB FC 0E 01 C3 0C 00 3E 41 D3 E8 3E 7F D3 .........>A..>..
0930 E1 3E 40 D3 E1 7D D3 E0 7C D3 E0 79 D3 FA 3E 88 .>@..}..|..y..>.
0940 B0 D3 F8 DB FD 07 DA 43 00 C5 01 80 00 09 C1 DB .......C........
0950 F8 E6 9D C8 1D C2 02 00 2F D3 FF C3 3B 00 00 00 ......../...;...
0960 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0970 00 00 00 00 00 00 00 00 00 00 00 00 00 C7 E5 00 ................
0980 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0990 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
09A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
09B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-D
09C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
09D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
09E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
09F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0A00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0A10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0A20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0A30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0A40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0A50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0A60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0A70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

```
                ; CP/M BASIC INPUT/OUTPUT OPERATING SYSTEM (BIOS)
                ; TARBELL ELECTRONICS
                ; 2.2 VERSION OF 2-19-80
                ; CHANGED SIGN-ON FOR CPM 2.2 2-19-80
                ;
                ; THIS MODULE CONTAINS ALL THE INPUT/OUTPUT
                ; ROUTINES FOR THE CP/M SYSTEM, INCLUDING
                ; THE DISK ROUTINES.
                ;
                ; THIS SECTION DEFINES THE I/O PORTS AND
                ; STATUS BITS.  BY SETTING THE PROPER VALUES
                ; FOR THE EQU STATEMENTS, THE I/O MAY BE
                ; AUTOMATICALLY RECONFIGURED TO FIT MOST
                ; SITUATIONS.  THE TRUE AND FALSE ONES
                ; CONTROL CONDITIONAL ASSEMBLIES OF DIFFERENT
                ; SECTIONS OF I/O ROUTINES TO FIT DIFFERENT
                ; INTERFACE REQUIREMENTS.

FFFF =          TRUE    EQU  OFFFFH     ;DEFINE VALUE OF TRUE.
0000 =          FALSE   EQU  NOT TRUE   ;DEFINE VALUE OF FALSE.

                ;*****************************************************
                ;*** THIS BEGINS THE AREA WHICH REQUIRES CHANGES ***
                ;***       FOR DIFFERENT CONSOLE I/O SYSTEMS       ***
                ;*****************************************************

0014 =          MSIZE   EQU  20         ;MEMORY SIZE IN KBYTES.
0000 =          INTRP   EQU  FALSE      ;TRUE IF INTERRUPTS ALLOWED.

0000 =          STD     EQU  FALSE      ;TRUE IF STANDARD I/O.
FFFF =          MSIO2   EQU  TRUE       ;TRUE IF MITS 2SIO.
0000 =          DELTA   EQU  FALSE      ;TRUE IF USING DELTA PRODUCTS CPU.
0000 =          ISIO2   EQU  FALSE      ;TRUE IF IMSAI SIO-2.
0000 =          TUART   EQU  FALSE      ;TRUE IF CROMEMCO TUART.
0000 =          VIDEO   EQU  FALSE      ;TRUE IF USING A MEMORY MAPPED VIDEO.
0000 =          OTHER   EQU  FALSE      ;TRUE IF SOMETHING ELSE.
0000 =          SOLOS   EQU  FALSE      ;TRUE IF PROC TECH SOLOS.
0000 =          DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED DRIVES.
0000 =          SPOOL   EQU  FALSE      ;TRUE IF USING KLH SPOOLER.
0004 =          NDISK   EQU  4          ;DEFINES THE NUMBER DRIVES IN SYSTEM.

                        IF  VIDEO       ;IF USING A VIDEO BOARD
                OUTADDR EQU  0          ;PUT OUTPUT ADDRESS HERE
                        ENDIF

                        IF NOT SOLOS    ;IF NOT PROC TECH SOLOS,
0000 =          CSTAT   EQU  0          ;CONSOLE STATUS PORT.
                        ENDIF

0000 =          CCOM    EQU  0          ;CONSOLE COMMAND PORT.
0001 =          CDATA   EQU  1          ;CONSOLE DATA PORT.
0000 =          CONUL   EQU  FALSE      ;CONSOLE NULLS?
0000 =          CNULL   EQU  0          ;CONSOLE NULL COUNT.

0002 =          LSTAT   EQU  2          ;LIST STATUS PORT.
0002 =          LCOM    EQU  2          ;LIST COMMAND PORT.
0003 =          LDATA   EQU  3          ;LIST DATA PORT.
0000 =          LSTNUL  EQU  FALSE      ;LIST DEVICE NULLS?
0000 =          LNULL   EQU  0          ;LIST NULL COUNT.
0000 =          LSTPAG  EQU  FALSE      ;LIST DEVICE PAGING?
0042 =          LINCNT  EQU  66         ;LINES PER PAGE.
```

```
0008 =          HLAB    EQU  8          ;8 FOR HD LD AT BEG OF SEEK.
0001 =          STPRAT  EQU  1          ;RATE 1=6MS, 2=10MS, 3=20MS.
0000 =          DUAL    EQU  FALSE      ;TRUE IF DUAL DRIVE.
0000 =          PERSCI  EQU  FALSE      ;TRUE IF FAST SEEK (PERSCI).

                ;*********************************************************
                ;*** THIS IS THE END OF THE AREA WHICH NORMALLY NEED ***
                ;***     BE CHANGED FOR MOST CONSOLE I/O SYSTEMS     ***
                ;*********************************************************

0000 =          RDYLO   EQU  STD OR SOLOS OR OTHER  ;STATUS READY WHEN LOW.
FFFF =          RDYHI   EQU  NOT RDYLO

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                CSTAT   EQU  0FAH       ;CONSOLE STATUS PORT.
                KBD     EQU  0C02EH     ;SOLOS KEYBOARD.
                CLRSCR  EQU  0C0D5H     ;CLEAR SCREEN.
                $CRN    EQU  0C054H     ;SOLOS OUTPUT.
                        ENDIF


                        IF NOT SOLOS    ;IF NOT PROC TECH SOLOS,
00F8 =          DISK    EQU  0F8H       ;DISK BASE ADDRESS.
                        ENDIF

                        IF SOLOS        ;IF PROC TECH SOLOS,
                DISK    EQU  0E8H       ;DIFFERENT DISK PORTS.
                        ENDIF

00F8 =          DCOM    EQU  DISK       ;DISK COMMAND PORT.
00F8 =          DSTAT   EQU  DISK       ;DISK STATUS PORT.
00F9 =          TRACK   EQU  DISK+1     ;DISK TRACK PORT.
00FA =          SECTP   EQU  DISK+2     ;DISK SECTOR PORT.
00FB =          DDATA   EQU  DISK+3     ;DISK DATA PORT.
00FC =          WAIT    EQU  DISK+4     ;DISK WAIT PORT.
00FC =          DCONT   EQU  DISK+4     ;DISK CONTROL PORT.

000A =          RTCNT   EQU  10         ;RETRY COUNT.

                        IF   STD        ;IF STANDARD I/O,
                CKBR    EQU  00000001B  ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B  ;CONS OUTPUT RDY BIT.
                        ENDIF

                        IF   MSIO2      ;IF MITS 2SIO,
0001 =          CKBR    EQU  00000001B  ;KEYBOARD READY BIT.
0002 =          CPTR    EQU  00000010B  ;PRINT READY BIT.
                        ENDIF

                        IF ISIO2 OR DELTA
                CKBR    EQU  00000010B  ;KEYBOARD READY BIT.
                CPTR    EQU  00000001B  ;PRINT READY BIT.
                        ENDIF

                        IF   TUART      ;IF CROMEMCO TUART,
                CKBR    EQU  01000000B  ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B  ;PRINT READY BIT.
                        ENDIF

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                CKBR    EQU  00000001B  ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B  ;DUMMY EQU.
                        ENDIF

                        IF OTHER        ;IF SOMETHING ELSE,
```

```
                CKBR    EQU  00000010B  ;KEYBOARD READY BIT.
                CPTR    EQU  10000000B  ;PRINTER READY BIT.
                        ENDIF

0002 =          LRBIT   EQU  CPTR       ;LISTER READY BIT.


0003 =          IOBYTE  EQU  3                  ;ADDRESS OF I/O BYTE.
0000 =          CBASE   EQU  (MSIZE-20)*1024    ;BIAS FOR LARGER THAN 20K.
3400 =          CPMB    EQU  CBASE+3400H        ;START OF CPM 2.0
3C06 =          BDOS    EQU  CPMB+806H          ;START OF BDOS 2.0.
4A00 =          BIOS    EQU  CPMB+1600H         ;START OF CBIOS IO.
0004 =          CDISK   EQU  4                  ;LOCATION 4 IS CURRENT DISK.
002C =          NSECTS  EQU  44                 ;NUMBER OF SECTORS IN IT.

4A00                    ORG  BIOS               ;START OF CBIOS STRUCTURE.
                ;
                ; I/O JUMP VECTOR
                ; THIS IS WHERE CPM CALLS WHENEVER IT NEEDS
                ; TO DO ANY INPUT/OUTPUT OPERATION.
                ; USER PROGRAMS MAY USE THESE ENTRY POINTS
                ; ALSO, BUT NOTE THAT THE LOCATION OF THIS
                ; VECTOR CHANGES WITH THE MEMORY SIZE.
                ;
4A00 C39C4A             JMP  BOOT               ;FROM COLD START LOADER.
4A03 C3EF4A     WBOOTE: JMP  WBOOT              ;FROM WARM BOOT.
4A06 C3484B             JMP  CONST              ;CHECK CONSOLE KB STATUS.
4A09 C3514B             JMP  CONIN              ;READ CONSOLE CHARACTER.
4A0C C35D4B             JMP  CONOT              ;WRITE CONSOLE CHARACTER.
4A0F C30B4D             JMP  LIST               ;WRITE LISTING CHAR.
4A12 C3164D             JMP  PUNCH              ;WRITE PUNCH CHAR.
4A15 C3174D             JMP  READER             ;READ READER CHAR.
4A18 C3A34B             JMP  HOME               ;MOVE DISK TO TRACK ZERO.
4A1B C3684B             JMP  SELDSK             ;SELECT DISK DRIVE.
4A1E C3A54B             JMP  SETTRK             ;SEEK TO TRACK IN REG A.
4A21 C3FB4B             JMP  SETSEC             ;SET SECTOR NUMBER.
4A24 C3064C             JMP  SETDMA             ;SET DISK STARTING ADR.
4A27 C32D4C             JMP  READ               ;READ SELECTED SECTOR.
4A2A C3B14C             JMP  WRITE              ;WRITE SELECTED SECTOR.
4A2D C3044D             JMP  PRSTAT             ;LIST STATUS CHECK.
4A30 C3004C             JMP  SECTRAN            ;SECTOR TRANSLATE ROUTINE.

                        ; THESE ENTRY POINTS ADDED BY TARBELL ELECTRONICS.

                        IF   SPOOL              ;IF USING KLH SPOOLER.
                        DB    0FFH              ;FLAG FOR SPOOLER.
                        DW    LTBSY             ;LISTER STATUS LOCATION
                        DW    LTBSY             ;FOR SPOOLER - -
                        DW    LTBSY             ;I DON'T KNOW WHY IT'S
                        DW    LTBSY             ;HERE 4 TIMES EITHER.
                        ENDIF
                ;
                ; THIS SECTION DEFINES THE THE DISK PARAMETERS
                ; NOTE:
                ; IF YOU HAVE THE MACRO ASSEMBLER (MAC) FROM
                ; DIGITAL RESEARCH, YOU MAY ELIMINATE THIS SECTION
                ; STARTING AT **AA** TO **BB** AND USE THE MACRO
                ; FILE "DISKDEF" TO CUSTOM TAILOR YOUR SYSTEM TO
                ; ALLOW DIFFERENT TYPES OF DRIVES OR MORE THAN 4
                ; DRIVES.
                ;
                ;**AA**
                ;
4A33 =          DPBASE  EQU  $                  ;BASE OF DISK PARAMETER BLOCK
4A33 824A0000   DPE0:   DW   XLT0,0000H         ;TRANSLATE TABLE
4A37 00000000           DW   0000H,0000H        ;SCRATCH AREA
```

```
4A3B 2A4D734A           DW    DIRBUF,DPB0       ;DIR BUFF, PARM BLOCK
4A3F C94DAA4D           DW    CSV0,ALV0         ;CHECK, ALLOC VECTORS
                        ;
4A43 824A0000  DPE1:    DW    XLT1,0000H
4A47 00000000           DW    0000H,0000H
4A4B 2A4D734A           DW    DIRBUF,DPB1
4A4F F84DD94D           DW    CSV1,ALV1
                        ;
4A53 824A0000  DPE2:    DW    XLT2,0000H
4A57 00000000           DW    0000H,0000H
4A5B 2A4D734A           DW    DIRBUF,DPB2
4A5F 274E084E           DW    CSV2,ALV2
                        ;
4A63 824A0000  DPE3:    DW    XLT3,0000H
4A67 00000000           DW    0000H,0000H
4A6B 2A4D734A           DW    DIRBUF,DPB3
4A6F 564E374E           DW    CSV3,ALV3
                        ;
                        ;THE FOLLOWING DESCRIBES THE DISK PHYSICAL
                        ;NATURE, SUCH AS SECTORS/TRACK,DIRECTORY SIZE.
                        ;
4A73 =         DPB0     EQU   $                 ;ONE OF 4 DISK PARM. BLOCKS
4A73 1A00               DW    26                ;SECTORS/TRACK
4A75 03                 DB    3                 ;BLOCK SHIFT
4A76 07                 DB    7                 ;BLOCK MASK
4A77 00                 DB    0                 ;EXTNT MASK
4A78 F200               DW    242               ;DISK SIZE - 1
4A7A 3F00               DW    63                ;DIRECTORY MAX.
4A7C C0                 DB    192               ;ALLOC0
4A7D 00                 DB    0                 ;ALLOC1
4A7E 1000               DW    16                ;CHECK SIZE
4A80 0200               DW    2                 ;NUMBER OF SYSTEM TRACKS
                        ;
                        ;SECTOR TRANSLATION TABLE
                        ;
4A82 =         XLT0     EQU   $                 ;START OF TRANS. TABLE
4A82 01070D1319         DB    1,7,13,19,25
4A87 050B111703         DB    5,11,17,23,3
4A8C 090F150208         DB    9,15,21,2,8
4A91 0E141A060C         DB    14,20,26,6,12
4A96 1218040A10         DB    18,24,4,10,16,22
                        ;
4A73 =         DPB1     EQU   DPB0              ;EQUIVALENT PARAMETERS
4A82 =         XLT1     EQU   XLT0              ;SAME TRANSLATE TABLE
                        ;
4A73 =         DPB2     EQU   DPB0
4A82 =         XLT2     EQU   XLT0
                        ;
4A73 =         DPB3     EQU   DPB0
4A82 =         XLT3     EQU   XLT0
                        ;
                        ;**BB**
                        ;
                        ;
                        ; BOOT
                        ; THIS SECTION IS EXECUTED WHENEVER RESET AND RUN
                        ; IS PUSHED, AFTER THE COLDSTART LOADER READS IN
                        ; THE CPM SYSTEM.
                        ;
4A9C 318000    BOOT:    LXI   SP,80H      ;SET STACK POINTER.

                        IF    INTRP       ;IF INTERRUPTS ALLOWED,
                        EI                ;ENABLE THEM HERE.
                        ENDIF

                        IF    STD         ;IF STANDARD I/O,
```

```
                        DW    0,0         ;LEAVE SPACE FOR INIT.
                        DW    0,0
                        DW    0,0
                        DW    0,0
                        ENDIF

                        IF    MSIO2       ;IF MITS 2SIO,
4A9F 3E03               MVI   A,3         ;INITIALIZE 2SIO.
4AA1 D300               OUT   CCOM
4AA3 D302               OUT   LCOM
4AA5 3E11               MVI   A,11H
4AA7 D300               OUT   CCOM
4AA9 D302               OUT   LCOM
                        ENDIF

                        IF ISIO2 OR DELTA
                        MVI   A,0AAH      ;INITIALIZE SIO 2-2.
                        OUT   CCOM
                        OUT   LCOM
                        MVI   A,40H
                        OUT   CCOM
                        OUT   LCOM
                        MVI   A,0CEH
                        OUT   CCOM
                        OUT   LCOM
                        MVI   A,37H
                        OUT   CCOM
                        OUT   LCOM
                        ENDIF

                        IF    TUART       ;IF CROMEMCO TUART,
                        MVI   A,1         ;SET A = 1.
                        OUT   54H         ;SELECT DEVICE A.
                        OUT   52H         ;RESET DEVICE B.
                        LXI   H,BAUDRS    ;GET ADR OF BAUD RATE TABLE.
                        MVI   A,11H       ;OCTUPLE THE CLOCK.
                IT1:    OUT   02H         ;& RESET CURRENT DEV.
                        MOV   A,M         ;GET BAUD RATE FROM TABLE.
                        INX   H           ;INCREMENT POINTER.
                        OUT   0           ;SET BAUD RATE.
                        CALL  CONIN       ;READ KEYBOARD.
                        CALL  CONIN       ;READ KEYBOARD AGAIN.
                        CPI   0DH         ;IF NOT CARRIAGE-RETURN,
                        MVI   A,1         ;SLOW THE CLOCK.
                        JNZ   IT1         ;UNTIL A CARRIAGE-RETURN.
                        ENDIF

                        IF    SOLOS       ;IF PROC TECH SOLOS,
                        CALL  CLRSCR      ;CLEAR SCREEN.
                        ENDIF

4AAB AF                 XRA   A           ;CLEAR SCRATCH AREA.
4AAC 320300             STA   IOBYTE      ;CLEAR I/O BYTE.
4AAF 320400             STA   CDISK       ;SELECT DRIVE ZERO
4AB2 0E09               MVI   C,ENDZ-STARTZ    ;GET LENGTH OF ZERO AREA.
4AB4 211C4D             LXI   H,STARTZ    ;GET SCRATCH ADDRESS.
4AB7 77         BOOTL:  MOV   M,A         ;PUT ZERO IN MEMORY.
4AB8 23                 INX   H           ;INCREMENT POINTER.
4AB9 0D                 DCR   C           ;DECREMENT COUNTER.
4ABA C2B74A             JNZ   BOOTL       ;LOOP TILL DONE.
4ABD 3EF2               MVI   A,0F2H      ;SET LATCH CODE = F2.
4ABF 32284D             STA   LATCH
4AC2 CDD44A             CALL  SETUP       ;SET UP JUMPS.
4AC5 DB01               IN    CDATA       ;CLEAR CONSOLE STATUS.
4AC7 21EE4C             LXI   H,SMSG      ;PRINT OPENING MESSAGE.
4ACA CDDC4C             CALL  PMSG
```

```
4ACD 3A0400    GOCPM:  LDA   CDISK       ;GET DISK NUMBER TO
4AD0 4F                MOV   C,A         ;PASS TO CCP IN C.
4AD1 C30034            JMP   CPMB        ;JUMP TO CCP.
                       ;
                       ; SET UP JUMPS INTO CP/M IN LOWER MEMORY.
                       ;
4AD4 3EC3      SETUP:  MVI   A,0C3H      ;PUT JMP TO WBOOT
4AD6 320000            STA   0           ;ADR AT ZERO.
4AD9 21034A            LXI   H,WBOOTE
4ADC 220100            SHLD  1
4ADF 320500            STA   5
4AE2 21063C            LXI   H,BDOS      ;PUT JUMP TO BDOS
4AE5 220600            SHLD  6           ;AT ADR 5,6,7.
4AE8 218000            LXI   H,80H       ;SET DEFAULT DMA ADR.
4AEB 221A4D            SHLD  DMAADD
4AEE C9                RET               ;RETURN FROM SETUP.

                       IF    TUART       ;IF CROMEMCO TUART,
                BAUDRS: DB    94H,0CEH,0A2H,92H,88H,84H,82H,1
                       ENDIF

                       ;
                       ; WARM-BOOT:  READ ALL OF CPM BACK IN
                       ; EXCEPT BIOS, THEN JUMP TO CCP.
                       ;
4AEF 318000    WBOOT:  LXI   SP,80H      ;SET STACK POINTER.

                       IF   INTRP        ;IF INTERRUPTS ALLOWED,
                       EI                ;ALLOW THEM HERE.
                       ENDIF

                       IF    LSTPAG      ;IF LIST DEVICE PAGING,
                       XRA   A           ;RESET LINE-FEED COUNT.
                       STA   LFCNT
                       ENDIF

4AF2 3A0400            LDA   CDISK       ;SAVE DISK NUMBER.
4AF5 32274D            STA   TEMP
4AF8 0E00              MVI   C,0         ;SELECT DISK ZERO.
4AFA CD684B            CALL  SELDSK
4AFD CDA34B            CALL  HOME        ;MOVE TO TRACK ZERO.
4B00 C23D4B            JNZ   RDERR       ;IF ERROR, PRINT MESSAGE.
4B03 162C              MVI   D,NSECTS    ;GET # SECTORS FOR CPM READ.
4B05 010200            LXI   B,2         ;TRACK (B)=0, SECTOR (C)=2.
4B08 210034            LXI   H,CPMB      ;GET STARTING ADDRESS.

                       IF   INTRP        ;IF INTERRUPTS ALLOWED,
                       DI                ;DISABLE THEM HERE.
                       ENDIF

4B0B C5        RDBLK:  PUSH  B           ;SAVE B&C.
4B0C 48                MOV   C,B         ;GO TO TRACK IN B.
4B0D CDA54B            CALL  SETTRK
4B10 C1                POP   B           ;RESTORE B&C.
4B11 C23D4B            JNZ   RDERR       ;IF ERROR, PRINT MESSAGE.
4B14 221A4D    RBLK1:  SHLD  DMAADD      ;SET STARTING ADDRESS.
4B17 CDFB4B            CALL  SETSEC      ;READ STARTING AT SECTOR IN C.
4B1A CD2D4C            CALL  READ
4B1D C23D4B            JNZ   RDERR       ;IF ERROR, PRINT MESSAGE.
4B20 15                DCR   D           ;DECREMENT SECTOR COUNT.
4B21 CA314B            JZ    ALDON       ;ALL DONE WHEN D=0.
4B24 0C                INR   C           ;INCREMENT SECTOR NUMBER.
4B25 79                MOV   A,C         ;IF SECTOR NUMBER
4B26 FE1B              CPI   27          ;IS NOT 27,
4B28 DA144B            JC    RBLK1       ;KEEP READING ON THIS TRACK.
4B2B 0E01              MVI   C,1         ;OTHERWISE, RESET SECTOR=1,
```

```
4B2D 04                 INR  B          ;INCREMENT TRACK NUMBER,
4B2E C30B4B             JMP  RDBLK      ;AND READ NEXT TRACK.
4B31 3A274D   ALDON:    LDA  TEMP       ;RESTORE DISK NUMBER.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        EI              ;ALLOW THEM AGAIN HERE.
                        ENDIF

4B34 320400             STA  CDISK
4B37 CDD44A             CALL SETUP      ;SET UP JUMPS.
4B3A C3CD4A             JMP  GOCPM      ;GO BACK TO CPM.
                ;
4B3D 0E42     RDERR:    MVI  C,'B'      ;GET BOOT ERROR CODE.
4B3F CD5D4B             CALL CONOT      ;PRINT IT.
4B42 CD514B             CALL CONIN      ;READ A CHAR FROM CONSOLE.
4B45 C3EF4A             JMP  WBOOT      ;DO A WARM BOOT.
                ;
                ; CHECK CONSOLE INPUT STATUS.
                ;

4B48 DB00     CONST:    IN   CSTAT      ;READ CONSOLE STATUS.
4B4A E601               ANI  CKBR       ;LOOK AT KB READY BIT.
4B4C 3E00     CONST1:   MVI  A,0        ;SET A=0 FOR RETURN.

                        IF   RDYLO      ;IF STATUS READY LOW,
                        RNZ             ;NOT READY WHEN NOT 0.
                        ENDIF

                        IF   RDYHI      ;IF STATUS READY HIGH,
4B4E C8                 RZ              ;NOT READY WHEN ZERO.
                        ENDIF

4B4F 2F                 CMA             ;IF READY A=FF.
4B50 C9                 RET             ;RETURN FROM CONST.

                ;
                ; READ A CHARACTER FROM CONSOLE.
                ;
                CONIN:

                        IF NOT SOLOS    ;IF NOT PROC TECH SOLOS,
4B51 DB00               IN   CSTAT      ;READ CONSOLE STATUS.
4B53 E601               ANI  CKBR       ;IF NOT READY,
                        ENDIF

                        IF   SOLOS      ;IF PROC TECH SOLOS,
                        CALL KBD        ;READ SOL KEYBOARD.
                        JZ   CONIN      ;READY WHEN NOT ZERO.
                        ENDIF

                        IF RDYLO AND NOT SOLOS
                        JNZ  CONIN      ;LOOP UNTIL LOW.
                        ENDIF

                        IF   RDYHI      ;IF READY WHEN HIGH,
4B55 CA514B             JZ   CONIN      ;LOOP UNTIL HIGH.
                        ENDIF

                        IF NOT SOLOS    ;IF NOT PROC TECH SOLOS,
4B58 DB01               IN   CDATA      ;READ A CHARACTER.
                        ENDIF

4B5A E67F               ANI  7FH                ;MAKE MOST SIG. BIT = 0.
4B5C C9                 RET                     ;RETURN FROM CONIN.
                ;
                ; WRITE A CHARACTER TO THE CONSOLE DEVICE.
```

```
                        ;
                        CONOT:
                                IF CONUL                ;IF NULLS REQUIRED,
                                MVI  A,0DH              ;IF IT'S A CR,
                                CMP  C                  ;THEN HOP OUT
                                JZ   CONULL             ;TO NULL ROUTINE.
                                ENDIF
                        CONOT1:
                                IF NOT SOLOS AND NOT VIDEO
4B5D DB00                       IN   CSTAT              ;READ CONSOLE STATUS.
4B5F E602                       ANI  CPTR               ;IF NOT READY,
                                ENDIF

                                IF RDYLO AND NOT SOLOS AND NOT VIDEO
                                JNZ  CONOT1             ;LOOP UNTIL LOW.
                                ENDIF

                                IF RDYHI                ;IF READY WHEN HIGH,
4B61 CA5D4B                     JZ   CONOT1             ;LOOP UNTIL HIGH.
                                ENDIF

                                IF NOT SOLOS AND NOT VIDEO
4B64 79                         MOV  A,C                ;GET CHARACTER.
4B65 D301                       OUT  CDATA              ;PRINT IT.
4B67 C9                         RET                     ;RETURN.
                                ENDIF

                        ;
                        ;THIS ROUTINE CALLES YOUR VIDEO DRIVER
                        ;ROUTINE WHICH MUST BE IN ROM.
                        ;ALL REGISTERS MUST BE SAVED AND RESTORED
                        ;BY YOUR VIDEO DRIVER IN ORDER TO BE
                        ;COMPATIABLE WITH CPM.
                        ;CPM PASSES THE CHAR. TO BE OUTPUT IN THE
                        ; C REGISTER. MAKE ANY CHANGES IN THIS
                        ;ROUTINE TO PASS THE CHAR FROM REG C TO
                        ;THE REGISTER YOUR VIDEO DRIVER EXPECTS
                        ;IT TO BE IN.
                        ;
                                IF  VIDEO               ;IF USING A VIDEO DRIVER IN ROM
                                MOV  A,C                ;GET THE CPM CHAR INTO REG A
                                CALL OUTADDR            ;CALL YOUR VIDEO DRIVER.
                                RET                     ;RETURN TO CPM.
                                ENDIF

                                IF CONUL
                        CONULL: PUSH B                  ;SAVE B&C.
                                MVI  B,CNULL+1          ;GET NULL COUNT.
                        CONUL1: CALL CONOT1             ;PRINT CR.
                                MVI  C,0                ;GET NULL CHAR.
                                DCR  B                  ;DECREMENT COUNTER.
                                JNZ  CONUL1             ;DO NEXT NULL.
                                POP  B                  ;RESTORE B&C.
                                MOV  A,C                ;RESTORE A.
                                RET                     ;RETURN.
                                ENDIF

                                IF   SOLOS              ;IF PROC TECH SOLOS,
                                PUSH B                  ;SAVE B&C.
                                MOV  B,C                ;PUT CHAR IN B REG.
                                CALL SCRN               ;OUTPUT CHAR TO SOLOS.
                                POP  B                  ;RESTORE B&C.
                                MOV  A,C                ;PUT CHAR IN A.
                                RET                     ;RETURN FROM CONOT.
                                ENDIF
                        ;
```

```
                        ; SELECT DISK NUMBER ACCORDING TO REGISTER C.
                        ;
                        SELDSK:
4B68 210000                     LXI   H,0               ;SET UP FOR ERROR CODE
4B6B 79                         MOV   A,C               ;GET NEW DRIVE.
4B6C FE04                       CPI   NDISK             ;CALLING UNDEFINED DRIVE ?
4B6E D0                         RNC                     ;IF NO CY, H,L TELLS CPM YES.
4B6F 321D4D                     STA   PDISK             ;WE ARE OK, SAVE NEW DISK NUM
4B72 211C4D                     LXI   H,DISKNO          ;GET OLD DRIVE NUMBER.
                        SELMOR:
4B75 7E                         MOV   A,M         ;GET OLD DISK NUMBER.

                                IF    DUAL        ;IF DUAL DRIVE,
                                ANI   0FEH        ;CLEAR OUT BIT 0.
                                ENDIF

4B76 5F                         MOV   E,A         ;PUT OLD DISK NO. IN D&E.
4B77 1600                       MVI   D,0
4B79 21214D                     LXI   H,TRTAB     ;GET ADDRESS OF TRACK TABLE.
4B7C E5                         PUSH  H           ;SAVE ADDRESS OF TRTAB.
4B7D 19                         DAD   D           ;ADD DISK NO. TO ADDRESS.
4B7E DBF9                       IN    TRACK       ;READ 1771 TRACK REGISTER.
4B80 77                         MOV   M,A         ;PUT INTO TABLE.
4B81 79                         MOV   A,C         ;GET NEW DISK NUMBER.

                                IF    DUAL        ;IF A DUAL DRIVE,
                                ANI   0FEH        ;CLEAR BIT 0.
                                ENDIF

4B82 5F                         MOV   E,A         ;PUT NEW DISK NO. IN D&E.
4B83 E1                         POP   H           ;RESTORE ADDRESS OF TRTAB.
4B84 19                         DAD   D           ;ADD DISK NO. TO ADDRESS.
4B85 7E                         MOV   A,M         ;GET NEW TRACK NUMBER.
4B86 D3F9                       OUT   TRACK       ;PUT INTO 1771 TRACK REG.
4B88 79                         MOV   A,C         ;UPDATE OLD DISK NUMBER.
4B89 321C4D                     STA   DISKNO
4B8C 2F                         CMA               ;BITS INVERTED INTO LATCH.
4B8D 87                         ADD   A           ;PUT BITS 1&2 AT 4&5.
4B8E 87                         ADD   A
4B8F 87                         ADD   A
4B90 87                         ADD   A
4B91 F602                       ORI   2           ;MAKE LATCH COMMAND.
4B93 32284D                     STA   LATCH       ;SAVE NEW LATCH CODE.
                        ;
                        ;SELECT DRIVE AS A FUNCTION OF H,L
                        ;
4B96 2A1D4D                     LHLD  PDISK             ;LOAD DISK NUMBER AND ZERO BY
4B99 11334A                     LXI   D,DPBASE          ;POINT TO DISK PARM START.
4B9C 29                         DAD   H                 ;*2
4B9D 29                         DAD   H                 ; *4
4B9E 29                         DAD   H                 ;  *8
4B9F 29                         DAD   H                 ;   *16
4BA0 19                         DAD   D                 ;COMPUTE INDEX FOR THE DRIVE
4BA1 AF                         XRA   A           ;SET A = 0.
4BA2 C9                         RET               ;RETURN FROM SELDSK.
                        ;
                        ; MOVE DISK TO TRACK ZERO.
                        ;
4BA3 0E00               HOME:   MVI   C,0         ;SEEK TO TRACK ZERO.
                        ;
                        ; SET TRACK NUMBER TO WHATEVER IS IN REGISTER C.
                        ; ALSO PERFORM MOVE TO THE CORRECT TRACK (SEEK).
                        ;
                        SETTRK:
                                IF NOT DUBSID   ;IF NOT DOUBLE-SIDED,
4BA5 E5                         PUSH H          ;SAVE H&L.
```

```
4BA6 2A284D             LHLD LATCH      ;GET NEW & OLD LATCH.
4BA9 7D                 MOV  A,L        ;GET NEW LATCH.
4BAA D3FC               OUT  DCONT      ;SELECT DRIVE NOW.
4BAC 32294D             STA  CLATCH     ;REMEMBER CURRENT LATCH.
4BAF BC                 CMP  H          ;IS NEW SAME AS OLD?
4BB0 3EFF               MVI  A,0FFH     ;IF NOT, SET FLAG = FF.
4BB2 C2B64B             JNZ  SFLAG
4BB5 2F                 CMA             ;IF NEW = OLD, FLAG = 0.
4BB6 321F4D     SFLAG:  STA  HLSF       ;SET HEAD-LOAD/SELECT FLAG.
4BB9 E1                 POP  H          ;RESTORE H&L.
                        ENDIF

4BBA 79                 MOV  A,C        ;GET NEW TRACK NUMBER.

                        IF   DUBSID     ;IF DOUBLE-SIDED DISK,
                        RRC             ;SHIFT RIGHT ONCE.
                        PUSH PSW        ;SAVE REVISED TRACK NUMBER.
                        PUSH H          ;SAVE H&L.
                        LHLD LATCH      ;GET NEW & OLD LATCH.
                        MOV  A,L        ;PUT NEW LATCH IN A.
                        JC   SIDE2      ;IF TRACK # ODD, SIDE #2.
                        ORI  40H        ;CLEAR LATCH BIT FOR SIDE 1.
                        JMP  SETLAT     ;GO AHEAD AND SET LATCH.
                SIDE2:  ANI  0B2H       ;SET LATCH BIT FOR SIDE 2.
                SETLAT: OUT  DCONT      ;OUTPUT TO THE LATCH.
                        STA  CLATCH     ;SET CURRENT LATCH CODE.
                        XRA  H          ;COMPARE OLD WITH NEW,
                        ANI  0BFH       ;IGNORE SIDE BIT.
                        MVI  A,0FFH     ;IF OLD NOT = NEW,
                        JNZ  SETFL      ;   SET FLAG = FF.
                        CMA  A          ;IF OLD = NEW, FLAG = 0.
                SETFL:  STA  HLSF       ;SET FLAG ZERO IF SAME.
                        POP  H          ;RESTORE H&L.
                        POP  PSW        ;RESTORE THE TRACK NUMBER.
                        ANI  7FH        ;CLEAR MSB.
                        MOV  C,A        ;SET C=TRACK NUMBER.
                        ENDIF

4BBB 32184D             STA  TRK        ;UPDATE OLD WITH NEW.
                ;
                ; MOVE THE HEAD TO THE TRACK IN REGISTER A.
                ;
4BBE C5         SEEK:   PUSH B          ;SAVE B&C.
4BBF 47                 MOV  B,A        ;SAVE DESTINATION TRACK.
4BC0 3E0A               MVI  A,RTCNT    ;GET RETRY COUNT.
4BC2 32264D     SRETRY: STA  SERCNT     ;STORE IN ERROR COUNTER.
4BC5 DBF9               IN   TRACK      ;READ PRESENT TRACK NO.
4BC7 4F                 MOV  C,A        ;SOME
4BC8 79                 MOV  A,C        ; DELAY
4BC9 B8                 CMP  B          ;SAME AS NEW TRACK NO.?
4BCA C2CF4B             JNZ  NOTHR      ;JUMP IF NOT THERE.
4BCD C1         THERE:  POP  B          ;RESTORE B&C.
4BCE C9                 RET             ;RETURN FROM SEEK.
                NOTHR:
                ;
                ;THIS ROUTINE IS TO ALLOW TIME FOR THE DRIVE
                ;TUNNEL ERASE TO TERMINATE BEFORE MOVING THE
                ;HEAD.  THE DELAY IS APPROX. 700 MICRO-SEC. @
                ;4 MHZ CPU TIME, AND DOUBLE THIS FOR 2 MHZ CPU'S.
                ;
4BCF F5                 PUSH PSW        ;SAVE ACCUM
4BD0 3ED0               MVI   A,0D0H    ;DELAY COUNT = 208
4BD2 3D         BUSY1:  DCR   A         ;DECREASE COUNT
4BD3 C2D24B             JNZ   BUSY1     ;LOOP TILL DONE
4BD6 F1                 POP  PSW        ;RESTORE ACCUM
```

```
                        IF NOT PERSCI    ;IF NOT PERSCI DRIVE,
4BD7 78                 MOV  A,B         ;RESTORE A FROM B.
4BD8 D3FB               OUT  DDATA       ;TRACK TO DATA REGISTER.
4BDA DBF8       BUSY:   IN   DSTAT       ;READ DISK STATUS.
4BDC OF                 RRC              ;LOOK AT BIT O.
4BDD DADA4B             JC   BUSY        ;WAIT TILL NOT BUSY.
4BEO 3E1D               MVI  A,14H+STPRAT+HLAB  ;GET STEP RATE, DO
4BE2 D3F8               OUT  DCOM        ;SEEK WITH VERIFY.
4BE4 DBFC               IN   WAIT        ;WAIT FOR INTRQ.
4BE6 DBF8               IN   DSTAT       ;READ STATUS.
4BE8 E691               ANI  91H         ;LOOK AT BITS.
4BEA CACD4B             JZ  THERE        ;OK IF ZERO.
                        ENDIF

                        IF  PERSCI       ;IF PERSCI DRIVE,
                        MVI  A,40H+HLAB   ;IF CARRY = 1,
                        JC  SDIR         ;STEP IN.
                        MVI  A,60H+HLAB   ;OTHERWISE, OUT.
                SDIR:   OUT  DCOM        ;ISSUE STEP DIRECTION.
                        MVI  A,20        ;DELAY LOOP COUNT.
                DLOOP:  DCR  A           ;DECREMENT COUNTER.
                        JNZ  DLOOP
                        MOV  A,C         ;GET PRESENT TRACK.
                        SUB  B           ;FIGURE TRACKS TO STEP.
                        JP   STEP        ;IF NEGATIVE,
                        CMA              ;FIGURE THE
                        INR  A           ;TWO'S COMPLEMENT.
                STEP:   MOV  C,A         ;GET DIFFERENCE.
                        MVI  A,1         ;PERSCI STEP COMMAND.
                STEP1:  OUT  DCONT       ;STEP PERSCI (E-14).
                        DCR  C           ;COUNT THE STEP.
                        JNZ  STEP1       ;STEP UNTIL C = O.
                        IN   WAIT        ;CLEAR 1771.
                        IN   DSTAT
                        MOV  A,B         ;GET DEST. TRACK.
                        OUT  TRACK       ;UPDATE TRACK REG.
                        LDA  CLATCH      ;GET LATCH CODE.
                        ANI  72H         ;MAKE COMMAND TO
                        OUT  DCONT       ;SWITCH WAIT FOR
                        IN   WAIT        ;SEEK COMPLETE.
                        LDA  CLATCH      ;RESTORE LATCH TO
                        OUT  DCONT       ;OLD CODE.
                        XRA  A           ;MAKE GOOD RETURN.
                        POP  B           ;RESTORE  B&C.
                        RET
                        ENDIF

                        IF  NOT PERSCI   ;IF NOT PERSCI DRIVE,
4BED 3A264D             LDA  SERCNT      ;GET ERROR COUNT.
4BFO 3D                 DCR  A           ;DECREMENT COUNT.
4BF1 C2C24B             JNZ  SRETRY      ;RETRY SEEK.
4BF4 C1                 POP  B           ;RESTORE B&C.
4BF5 CD584C             CALL RECOV       ;IF SEEK RETRY = 10 CHECK
4BF8 C3BE4B             JMP  SEEK        ; FOR CNTL-C FOR ABORT.
                        ENDIF

                ;
                ; SET DISK SECTOR NUMBER.
                ;
4BFB 79         SETSEC:  MOV  A,C        ;GET SECTOR NUMBER.
4BFC 32194D              STA  SECT       ;PUT AT SECT # ADDRESS.
4BFF C9                  RET             ;RETURN FROM SETSEC.
                ;
                ;TRANSLATE THE SECTOR GIVEN B,C
                ;USING THE TRANSLATE TABLE
                ;GIVEN BY D,E
```

```
                        ;
                        SECTRAN:
4C00 EB                         XCHG                    ;H,L = TRANS
4C01 09                         DAD   B                 ;H,L = TRANS (SECTOR)
4C02 6E                         MOV   L,M               ;L   = TRANS (SECTOR)
4C03 2600                       MVI   H,0               ;CLEAR REG H
4C05 C9                         RET                     ;H,L = TRANSLATED SECTOR
                        ;
                        ; SET DISK DMA ADDRESS.
                        ;
4C06 60                 SETDMA:  MOV  H,B       ;MOVE B&C TO H&L.
4C07 69                          MOV  L,C
4C08 221A4D                      SHLD DMAADD    ;PUT AT DMA ADR ADDRESS.
4C0B C9                          RET            ;RETURN FROM SETDMA.
                        ;
                        ; HDLD - GET HEAD-LOAD BIT IF REQUIRED.
                        ;
4C0C 3A1F4D             HDLD:    LDA  HLSF      ;GET HEAD-LOAD FLAG.
4C0F B7                          ORA  A         ;IS A = ZERO?
4C10 CA244C                      JZ   HDLD1     ;HOP IF SO.
4C13 2F                          CMA            ;SET A = 0.
4C14 321F4D                      STA  HLSF      ;SET FLAG = 0 IF NOT.
                        ;
                        ;IF CHANGING TO A NEW DRIVE, PERFORM A SEEK TO
                        ;THE SAME TRACK TO ALLOW THE HEAD TO UNLOAD.
                        ;
4C17 DBF9                        IN   TRACK     ;GET PRESENT TRACK
4C19 D3FB                        OUT  DDATA     ;AND TELL 1771 ABOUT IT.
4C1B 3E1D                        MVI  A,14H+STPRAT+HLAB ;GET THE STEP RATE.
4C1D D3F8                        OUT  DCOM      ;SEND IT TO FLOPPY CONTROLLER.
4C1F DBFC                        IN   WAIT      ;WAIT FOR INTRQ.
4C21 3E04               HDLDY:   MVI  A,4       ;SET BIT TO LOAD HEAD.
4C23 C9                          RET            ;RETURN FROM HDLD.
4C24 DBF8               HDLD1:   IN   DSTAT     ;READ 1771 STATUS.
4C26 E620                        ANI  20H       ;LOOK AT HL BIT.
4C28 CA214C                      JZ   HDLDY     ;LOAD IF NOT LOADED.
4C2B AF                          XRA  A         ;OTHERWISE, A=0.
4C2C C9                          RET            ;RETURN FROM HDLD.
                        ;
                        ; READ THE SECTOR AT SECT, FROM THE PRESENT TRACK.
                        ; USE STARTING ADDRESS AT DMAADD.
                        ;
4C2D 3E0A               READ:    MVI  A,RTCNT   ;GET RETRY COUNT.
                        RRETRY:
4C2F CD9C4C                      CALL DSKSET    ;SET UP DISK.
4C32 C688                        ADI  88H       ;ADD CODE FOR READ SECT.
4C34 D3F8               READE:   OUT  DCOM      ;SEND COMMAND TO 1771.
4C36 DBFC               RLOOP:   IN   WAIT      ;WAIT FOR DRQ OR INTRQ.
4C38 B7                          ORA  A         ;SET FLAGS.
4C39 F2434C                      JP   RDDONE    ;DONE IF INTRQ.
4C3C DBFB                        IN   DDATA     ;READ A DATA BYTE FROM DISK.
4C3E 77                          MOV  M,A       ;PUT BYTE INTO MEMORY.
4C3F 23                          INX  H         ;INCREMENT MEMORY POINTER.
4C40 C3364C                      JMP  RLOOP     ;KEEP READING.
4C43 DBF8               RDDONE:  IN   DSTAT     ;READ DISK STATUS.

                                 IF   INTRP     ;IF INTERRUPTS ALLOWED,
                                 EI             ;ALLOW AGAIN HERE.
                                 ENDIF

4C45 E69D                        ANI  9DH       ;LOOK AT ERROR BITS.
4C47 C8                          RZ             ;RETURN IF NONE.
4C48 CD674C                      CALL ERCHK     ;CHECK FOR SEEK ERROR.
4C4B 3A254D                      LDA  ERCNT     ;GET ERROR COUNT.
4C4E 3D                          DCR  A         ;DECREMENT COUNT.
4C4F C22F4C                      JNZ  RRETRY    ;TRY TO READ AGAIN.
```

```
4C52 CD584C             CALL RECOV      ;CHECK FOR ABORT OR CONTINUE
4C55 C32D4C             JMP  READ       ;IF NOT CNTL-C, TRY TO READ AGAIN.
                        ;
                        ;RECOV
                        ;THIS ROUTINE IS CALLED BY ANY READ,WRITE,SEEK
                        ;ROUTINE IF THE RETRY COUNT GOES TO 10. IF IT
                        ;DOES,THIS ROUTINE CALLS CONIN FOR A KEY TO BE
                        ;PUSHED. IF THE KEY IS A CNTL-C, THEN A WARMBOOT
                        ;IS EXECUTED. IF ANY OTHER KEY IS PUSHED, THEN
                        ;A RETURN IS MADE BACK TO THE CALLER AND THAT
                        ;ROUTINE IS RETRIED FOR 10 MORE TIMES.
                        ;
4C58 21E74C     RECOV:  LXI  H,ERRMSG   ;POINT TO "ERROR" MESSAGE.
4C5B CDDC4C             CALL PMSG       ;PRINT IT ON CONSOL.
4C5E CD514B             CALL CONIN      ;CHECK FOR PUSHED KEY.
4C61 FE03               CPI  03H        ;IS IT A CNTL-C ?
4C63 C0                 RNZ             ;RETURN TO CALLER IF NOT.
4C64 C3EF4A             JMP  WBOOT      ;YES, DO WARMBOOT.
                        ;
                        ; ERCHK - CHECK FOR RECORD NOT FOUND ERROR.
                        ;
4C67 57         ERCHK:  MOV  D,A        ;SAVE ERROR BITS IN D.
4C68 E610               ANI  10H        ;IF RECORD NOT FOUND,
4C6A C2704C             JNZ  CHKSK      ;DO A CHECK ON SEEK.
4C6D 7A                 MOV  A,D        ;OTHERWISE RESTORE BITS
4C6E B7                 ORA  A          ;SET FLAGS,
4C6F C9                 RET             ;AND RETURN.
                        ;CHECK FOR SEEK TO CORRECT TRACK,
                        ;AND CHANGE IF NECESSARY.
4C70 3EC4       CHKSK:  MVI  A,0C4H     ;SEND COMMAND TO 1771
4C72 D3F8               OUT  DCOM       ;TO READ ADDRESS.
4C74 DBFC               IN   WAIT       ;WAIT FOR DRQ OR INTRQ.
4C76 DBFB               IN   DDATA      ;READ THE TRACK ADDRESS.
4C78 47                 MOV  B,A        ;SAVE IN REGISTER B.
4C79 DBFC       CHKS2:  IN   WAIT       ;WAIT FOR INTRQ.
4C7B B7                 ORA  A          ;SET FLAGS.
4C7C F2844C             JP   CHKS3      ;DONE WITH READ ADR OP.
4C7F DBFB               IN   DDATA      ;READ ANOTHER BYTE.
4C81 C3794C             JMP  CHKS2      ;DO IT AGAIN.
4C84 DBF8       CHKS3:  IN  DSTAT       ;READ DISK STATUS.
4C86 B7                 ORA  A          ;SET FLAGS.
4C87 CA904C             JZ   CHKS4      ;READ ADR OK IF 0.
4C8A CDA34B             CALL HOME       ;OTHERWISE, HOME FIRST.
4C8D C3934C             JMP  CHKS5
4C90 78         CHKS4:  MOV  A,B        ;UPDATE TRACK REGISTER.
4C91 D3F9               OUT  TRACK
4C93 3A184D     CHKS5:  LDA  TRK        ;GET REQUIRED TRACK NO.
4C96 CDBE4B             CALL SEEK       ;MOVE THE HEAD TO IT.
4C99 7A                 MOV  A,D        ;GET ERROR BITS.
4C9A B7                 ORA  A          ;SET FLAGS.
4C9B C9                 RET             ;RETURN FROM ERCHK.
                        ;
                        ;DISK SET UP ROUTINE.
                        ;THIS ROUTINE IS COMMON TO
                        ;BOTH THE READ AND WRITE
                        ;ROUTINES.
                        ;
                        DSKSET:
4C9C 32254D             STA  ERCNT      ;STORE IN ERROR CTR.
4C9F 3ED0               MVI  A,0D0H     ;CAUSE INTERRUPT.
4CA1 D3F8               OUT  DCOM
4CA3 E3                 XTHL            ;SOME
4CA4 E3                 XTHL            ; DELAY

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        DI              ;DISABLE THEM HERE.
```

```
                        ENDIF

4CA5 2A1A4D             LHLD DMAADD     ;GET STARTING ADDR.
4CA8 3A194D             LDA  SECT       ;GET SECTOR NUMBER.
4CAB D3FA               OUT  SECTP      ;SET SECTOR INTO 1771.
4CAD CD0C4C             CALL HDLD       ;GET HEAD-LOAD BIT?
4CB0 C9                 RET             ;RETURN TO CALLER
                ;
                ; WRITE THE SECTOR AT SECT, ON THE PRESENT TRACK.
                ; USE STARTING ADDRESS AT DMAADD.
                ;
4CB1 3E0A       WRITE:  MVI  A,RTCNT    ;GET RETRY COUNT.
                WRETRY:
4CB3 CD9C4C             CALL DSKSET     ;SET UP DISK.
4CB6 C6A8               ADI  0A8H       ;ADD CODE FOR WRITE.
4CB8 D3F8       WRITE2: OUT  DCOM
4CBA DBFC       WLOOP:  IN   WAIT       ;WAIT FOR READY.
4CBC B7                 ORA  A          ;SET FLAGS.
4CBD F2C74C             JP   WDONE      ;HOP OUT WHEN DONE.
4CC0 7E                 MOV  A,M        ;GET BYTE FROM MEM.
4CC1 D3FB               OUT  DDATA      ;WRITE ONTO DISK.
4CC3 23                 INX  H          ;INCREMENT MEM PTR.
4CC4 C3BA4C             JMP  WLOOP      ;KEEP WRITING.
4CC7 DBF8       WDONE:  IN   DSTAT      ;READ DISK STATUS.

                        IF   INTRP      ;IF INTERRUPTS ALLOWED,
                        EI              ;ENABLE AGAIN HERE.
                        ENDIF

4CC9 E6FD               ANI  0FDH       ;LOOK AT THESE BITS.
4CCB C8                 RZ              ;RETURN IF NO ERR.
4CCC CD674C             CALL ERCHK      ;CHECK/CORRECT SEEK ERR.
4CCF 3A254D             LDA  ERCNT      ;GET ERROR COUNT.
4CD2 3D                 DCR  A          ;DECREMENT COUNT.
4CD3 C2B34C             JNZ  WRETRY     ;TRY TO WRITE AGAIN.
4CD6 CD584C             CALL RECOV      ;CHECK FOR ABORT
4CD9 C3B14C             JMP  WRITE      ;RETRY WRITE AGAIN.

                ;
                ; PRINT THE MESSAGE AT H&L UNTIL A ZERO.
                ;
4CDC 7E         PMSG:    MOV  A,M       ;GET A CHARACTER.
4CDD B7                  ORA  A         ;IF IT'S ZERO,
4CDE C8                  RZ             ;RETURN.
4CDF 4F                  MOV  C,A       ;OTHERWISE,
4CE0 CD5D4B              CALL CONOT     ;PRINT IT.
4CE3 23                  INX  H         ;INCREMENT H&L,
4CE4 C3DC4C              JMP  PMSG      ;AND GET ANOTHER.
                ;
                ; CBIOS MESSAGES
                ;
4CE7 4552524F52ERRMSG:  DB   'ERROR.',0
4CEE 0D0A546172SMSG:    DB   0DH,0AH,'Tarbell '
4CF8 3230               DB   MSIZE/10+'0',MSIZE MOD 10 + '0'
4CFA 4B2043504D         DB   'K CPM 2.2',0
                ;
                ;LIST STATUS CHECK ROUTINE
                ;
                PRSTAT:
4D04 DB02               IN   LSTAT      ;CHECK LIST STATUS PORT
4D06 E602               ANI  LRBIT      ;MASK AND CHECK READY BIT
4D08 C34C4B             JMP  CONST1     ;FINISH IN THE CONST ROUTINE.
                ;
                ; WRITE A CHARACTER ON LISTING DEVICE.
                ;
                LIST:
```

```
                        IF LSTNUL           ;IF NULLS OR PAGING,
                        MVI  A,0DH          ;IF IT'S A CR,
                        CMP  C              ;THEN HOP OUT TO
                        JZ   LINUL          ;NULL ROUTINE.
                        ENDIF

                        IF LSTPAG           ;IF PAGING
                        MVI A,0AH           ;GET A LINEFEED
                        CMP C               ;DOES IT MATCH?
                        JZ LINUL3
                        ENDIF

4D0B DB02       LTBSY:  IN   LSTAT          ;READ LISTER STATUS.

                        IF NOT DELTA
4D0D E602               ANI  LRBIT          ;LOOK AT READY BIT.
                        ENDIF

                        IF  DELTA
                        ANI  81H            ;LOOK AT BITS 7 AND 0
                        XRI  81H            ;BOTH MUST BE ONES TO PRINT
                        JNZ  LTBSY          ;LOOP TILL ONES
                        ENDIF

                        IF NOT DELTA AND RDYLO
                        JNZ  LTBSY          ;LOOP TILL LOW.
                        ENDIF

                        IF NOT DELTA AND RDYHI
4D0F CA0B4D             JZ   LTBSY          ;LOOP TILL HIGH.
                        ENDIF

4D12 79                 MOV  A,C            ;GET DATA BYTE.
4D13 D303               OUT  LDATA          ;PRINT IT.
4D15 C9                 RET                 ;RETURN FROM LIST.

                        IF LSTNUL           ;IF LIST NULLS
                LINUL:  PUSH B              ;SAVE B&C.
                        MVI  B,(LNULL AND 0FFH)+1 ;GET NULL COUNT
                LINUL1: CALL LTBSY          ;PRINT (CR FIRST).
                        MVI  C,0            ;GET NULL CHAR.
                        DCR  B              ;DECREMENT COUNTER.
                        JNZ  LINUL1         ;DO NEXT NULL.
                        JMP  LINUL2         ;EXIT THE ROUTINE.
                        ENDIF

                        IF   LSTPAG         ;IF LIST DEV. PAGING,
                LINUL3: PUSH B              ;SAVE B,C PAIR
                        LDA  LFCNT          ;GET LINE-FEED COUNT.
                        INR  A              ;INCREMENT IT.
                        STA  LFCNT          ;SAVE IT BACK.
                        CPI  LINCNT-(LINCNT/11)  ;END OF PAGE?
                        MVI  B,1            ;SET UP FOR 1 LF.
                        JNZ  NOTEOP         ;HOP IF NOT END.
                        XRA  A              ;SET LF COUNT = 0.
                        STA  LFCNT
                        MVI  B,(LINCNT/11)+1  ;BETWEEN PAGES.
                NOTEOP: MVI  C,0AH          ;GET LINE-FEED CODE.
                LSTPA1: CALL LTBSY          ;PRINT LINE-FEED.
                        DCR  B              ;DECREMENT LF COUNTER.
                        JNZ  LSTPA1         ;DO NEXT LINE FEED?
                        ENDIF

                        IF LSTNUL OR LSTPAG  ;IF NULLS OR PAGING,
                LINUL2: POP  B              ;RESTORE B&C.
                        MOV  A,C            ;RESTORE A.
```

```
                        RET             ;RETURN FROM LIST.
                        ENDIF

                ;
                ; PUNCH PAPER TAPE.
                ;
                PUNCH:
4D16 C9                 RET             ;RETURN FROM PUNCH.
                ;
                ;  NORMALLY USED TO READ PAPER TAPE.
                ;
                READER:
4D17 C9                 RET             ;RETURN FROM READER.

                ;NOTE:  AS THERE ARE ONLY SIX (6) SECTORS
                ;AVAILABLE FOR CBIOS ON THE SECOND SYSTEM TRACK (1),
                ;THE LAST ADDRESS BEFORE THIS POINT SHOULD BE NO
                ;GREATER THAN THE CBIOS STARTING ADDRESS + 037F (HEX).
                ;THIS WILL NORMALLY BE XD7F (HEX).

                ;
                ; BIOS SCRATCH AREA.
                ;
4D18            TRK:    DS      1       ;CURRENT TRACK NUMBER.
4D19            SECT:   DS      1       ;CURRENT SECTOR NUMBER.
4D1A            DMAADD: DS      2       ;DISK TRANSFER ADDRESS.
                ;
                ; THE NEXT SEVERAL BYTES, BETWEEN STARTZ AND
                ; ENDZ, ARE SET TO ZERO AT COLD BOOT TIME.
                ;
                STARTZ:                 ;START OF ZEROED AREA.
4D1C            DISKNO: DS      1       ;DISK NUMBER
4D1D            PDISK:  DS      2       ;NEW DISK TO SELECT AND A ZERO BYTE

                ; SPECIAL FLAGS.
4D1F            HLSF:   DS      1       ;HEAD-LOAD SELECT FLAG.
4D20            LFCNT:  DS      1       ;PAGING LINE-FEED COUNT.
                ;
                ; TRTAB - DISK TRACK TABLE - PRESENT POSITION OF
                ;       HEADS FOR UP TO 4 DRIVES.
                ;
4D21            TRTAB:  DS      4

                ENDZ:                   ;END OF ZEROED AREA.

4D25            ERCNT:  DS      1       ;ERROR COUNT FOR RETRIES.
4D26            SERCNT: DS      1       ;SEEK RETRY COUNTER.
4D27            TEMP:   DS      1       ;TEMPORARY STORAGE.
4D28            LATCH:  DS      1       ;NEW CODE FOR LATCH.
4D29            CLATCH: DS      1       ;CURRENT CODE IN LATCH.
4D2A =          BEGDAT  EQU     $
4D2A            DIRBUF: DS      128     ;DIRECTORY BUFFER
4DAA            ALV0:   DS      31
4DC9            CSV0:   DS      16
4DD9            ALV1:   DS      31
4DF8            CSV1:   DS      16
4E08            ALV2:   DS      31
4E27            CSV2:   DS      16
4E37            ALV3:   DS      31
4E56            CSV3:   DS      16
4E66 =          ENDDAT  EQU     $
013C =          DATSIZ  EQU     $-BEGDAT        ;TOTAL SIZE OF DISK PARM STOR
4E66                    END

A>
```

```
                        ;
                        ; TARBELL ELECTRONICS
                        ; CP/M COLDSTART LOADER
                        ; VERSION OF 3-22-'79.
                        ;
                        ; THIS PROGRAM IS LOADED AT LOCATION ZERO
                        ; BY THE BOOTSTRAP PROGRAM, AND EXECUTED.
                        ; ITS PURPOSE IS TO LOAD AND EXECUTE THE
                        ; CP/M DISK OPERATING SYSTEM AT THE TOP
                        ; OF THE MEMORY IN USE.
                        ;
0000 =                  FALSE   EQU  0          ;DEFINE VALUE OF FALSE.
FFFF =                  TRUE    EQU  NOT FALSE  ;DEFINE VALUE OF TRUE.
                        ;
                        ;********* THIS IS THE AREA TO MAKE CHANGES IN ***********
                        ;********* FOR DIFFERENT SYSTEM CONFIGURATIONS ***********
                        ;                                                      **
0014 =                  MSIZE   EQU  20         ;MEMORY SIZE IN DECIMAL KB.    **
0000 =                  DELTA   EQU  FALSE      ;TRUE IF DELTA PRODUCTS CPU    **
0000 =                  PERSCI  EQU  FALSE      ;TRUE FOR PERSCI DRIVES.       **
0000 =                  DUBSID  EQU  FALSE      ;TRUE FOR DOUBLE SIDED SYSTEMS.**
001A =                  SPT     EQU  26         ;NUMBER OF SECTORS PER TRACK.  **
00F8 =                  DISK    EQU  0F8H       ;DISK PORT BASE ADDRESS.       **
                        ;                                                      **
                        ;**************************************************************
                        ;**************************************************************
                        ;
00F8 =                  DCOM    EQU  DISK       ;COMMAND PORT.
00F8 =                  DSTAT   EQU  DISK       ;STATUS PORT.
00F9 =                  TRACK   EQU  DISK+1     ;TRACK PORT.
00FA =                  SECT    EQU  DISK+2     ;SECTOR PORT.
00FB =                  DATA    EQU  DISK+3     ;DATA PORT.
00FC =                  WAIT    EQU  DISK+4     ;WAIT PORT.
00FC =                  DCONT   EQU  DISK+4     ;CONTROL PORT.
0000 =                  CBASE   EQU  (MSIZE-20)*1024
3400 =                  CPMB    EQU  CBASE+3400H;START OF CP/M.
4A00 =                  BOOTE   EQU  CPMB+1600H ;COLD BOOT ENTRY POINT.
0033 =                  NSECTS  EQU  51         ;SECTORS OF CP/M.
000A =                  RTCNT   EQU  10         ;NUMBER OF RETRYS.

0000                    ORG  0          ;START OF LOADER.

0000 1E0A       BOOT:   MVI  E,RTCNT    ;GET RETRY COUNT.

                        IF   DELTA      ;IF USING DELTA PRODUCTS CPU.
                        MVI  A,1        ;CODE TO KICK OUT ROM
                        OUT  9
                        ENDIF

0002 310001     BLOOP:  LXI  SP,100H    ;SET STACK POINTER.
0005 210034             LXI  H,CPMB     ;CP/M STARTS HERE.
0008 1633               MVI  D,NSECTS   ;NUMBER OF SECTORS TO READ.
000A 0E02               MVI  C,2        ;SECTOR NUMBER.
000C 0604       RNTRK:  MVI  B,4        ;FOR HEAD LOAD.
000E 79                 MOV  A,C        ;SECTOR IN A.
000F CD2A00     RNSEC:  CALL READ       ;READ FIRST SECTOR.
0012 15                 DCR  D          ;IF DONE,
0013 CA004A             JZ   BOOTE      ;GO TO CP/M.
0016 0600               MVI  B,0        ;FOR NO HEAD LOAD.
0018 0C                 INR  C          ;INCREMENT TRACK COUNT.
0019 79                 MOV  A,C        ;DONE WITH
001A FE1B               CPI  SPT+1      ;THIS TRACK?
001C DA0F00             JC   RNSEC      ;IF NOT, READ NEXT SECTOR.

                        IF   NOT PERSCI AND NOT DUBSID
```

```
001F 3E53               MVI   A,53H       ;STEP COMMAND.
0021 D3F8               OUT   DCOM        ;ISSUE IT.
0023 DBFC               IN    WAIT        ;WAIT UNTIL DONE.
                        ENDIF

                        IF    PERSCI AND NOT DUBSID
                        MVI   A,50H       ;INCREMENT TRACK
                        OUT   DCOM        ;REGISTER.
                        IN    WAIT        ;WAIT FOR INTRQ.
                        MVI   A,1         ;STEP
                        OUT   DCONT       ;PERSCI.
                        MVI   A,72H       ;SWITCH WAIT FOR
                        OUT   DCONT       ;SEEK COMPLETE.
                        IN    WAIT        ;WAIT.
                        MVI   A,0F2H      ;SWITCH WAIT
                        OUT   DCONT       ;BACK.
                        ENDIF

                        IF    DUBSID      ;IF DOUBLE SIDED SYSTEM.
                        MVI   A,0B2H      ;SIDE SELECT COMMAND.
                        OUT   DCONT       ;ISSUE IT.
                        ENDIF

0025 0E01               MVI   C,1         ;SECTOR NUMBER.
0027 C30C00             JMP   RNTRK       ;READ NEXT TRACK.

002A D3FA       READ:   OUT   SECT        ;SET SECTOR REGISTER.
002C CD4100             CALL  CHECK       ;CHECK FOR ERROR.
002F 3E88               MVI   A,88H       ;COMMAND FOR READ.
0031 B0                 ORA   B           ;GET HEAD LOAD BIT.
0032 D3F8               OUT   DCOM        ;ISSUE COMMAND.
0034 DBFC       RLOOP:  IN    WAIT        ;WAIT FOR DRQ.
0036 B7                 ORA   A           ;SET FLAGS.
0037 F24100             JP    CHECK       ;JUMP IF DONE.
003A DBFB               IN    DATA        ;READ DATA.
003C 77                 MOV   M,A         ;PUT IN MEMORY.
003D 23                 INX   H           ;INCREMENT POINTER.
003E C33400             JMP   RLOOP       ;LOOP UNTIL DONE.

0041 DBF8       CHECK:  IN    DSTAT       ;READ STATUS.
0043 E69D               ANI   9DH         ;LOOK AT ERROR BITS.
0045 C8                 RZ                ;OK IF ZERO.
0046 1D                 DCR   E           ;DECREMENT RETRY COUNT.
0047 C20200             JNZ   BLOOP       ;TRY AGAIN IF NOT ZERO.
004A 328000             STA   EC          ;SAVE ERROR CODE.
004D 2F                 CMA               ;INVERT AND SEND
004E D3FF               OUT   0FFH        ;TO FRONT PANEL.
0050 C35000     HERE:   JMP   HERE        ;LOOP.

007D                    ORG   7DH         ;PUT JUMP HERE.
007D C30000             JMP   BOOT        ;JUMP INTO BOOT.
0080            EC:     END               ;END OF BOOT.

A>!
```

```
;-----------------------------------------------
; cp/m basic input/output operating system (bios)
; tarbell electronics
; 2.x version of 01-29-82
; copyright (c) 1980 tarbell electronics
;-----------------------------------------------
; this bios module is the cpm v2.x auto select bios.
; this bios reads single or double density disk.
; the double density disk contains 51 sectors/track,
; 77 tracks. track 0 = single density, tracks 1 - 76
; are double density at 51 sectors per track.
; note:  if you leave dmacntl false, you must have a cpu
; which runs at 4 mhz to run double density.
; this bios now supports double sided single/double density.
; this section defines the i/o ports and status bits.
; by setting the proper values for the equ statements,
; the i/o may be automatically reconfigured to fit most
; situations. the true and false ones control conditional
; assemblies of different sections of i/o routines to fit
; different interface requirements.
;
TRUE    EQU  0FFFFH             ;define value of true.
FALSE   EQU  NOT TRUE           ;define value of false.
;
;************************************************
;*** this begins the area which requires changes ***
;***      for different console i/o systems      ***
;************************************************
;
MSIZE   EQU  64         ;memory size in kbytes.
INTRP   EQU  FALSE      ;true if interrupts allowed.
TARBELL EQU  FALSE      ;true if using the tarbell z-80 cpu.
IOBASE  EQU  0          ;base io addr for tarbell cpu (0 or 10 hex).
TIMER   EQU  FALSE      ;true if using cpu timer (tarbell cpu board).
STD     EQU  FALSE      ;true if standard i/o.
MSIO2   EQU  FALSE      ;true if mits 2sio.
VDB8024 EQU  FALSE      ;true if using vdb-8024 board.
DELTA   EQU  FALSE      ;true if using delta products cpu.
ISIO2   EQU  TRUE       ;true if imsai sio-2.
TUART   EQU  FALSE      ;true if cromemco tuart.
VIDEO   EQU  FALSE      ;true if using a memory mapped video board.
OTHER   EQU  FALSE      ;true if something else.
SOLOS   EQU  FALSE      ;true if proc tech solos.
DUBSID  EQU  FALSE      ;true for double sided drives (1 logical drive).
DMACNTL EQU  TRUE       ;true if using dma control.
NDISK   EQU  4          ;defines the number drives in system.
;
RDYLO   EQU  STD OR SOLOS OR OTHER  ;status ready when low.
RDYHI   EQU  NOT RDYLO
TARDEL  EQU  TARBELL OR DELTA   ;if using tarbell or delta cpu.
;
        IF  VIDEO       ;if using a video board
OUTADDR EQU  00000H     ;put output address here
        ENDIF
;
        IF NOT SOLOS AND NOT TARDEL     ;if not proc tech solos,
CSTAT   EQU  3          ;console status port.
CCOM    EQU  3          ;console command port.
CDATA   EQU  2          ;console data port.
LSTAT   EQU  5          ;list status port.
LCOM    EQU  5          ;list command port.
LDATA   EQU  4          ;list data port.
        ENDIF
;
```

```
CONUL   EQU  FALSE       ;console nulls?
CNULL   EQU  16          ;console null count.
LSTNUL  EQU  FALSE       ;list device nulls?
LNULL   EQU  0           ;list null count.
LSTPAG  EQU  FALSE       ;list device paging?
LINCNT  EQU  66          ;lines per page.
HLAB    EQU  8           ;8 for hd ld at beg of seek.
STPRAT  EQU  1           ;rate 0=3ms,1=6ms, 2=10ms, 3=20ms.
DUAL    EQU  FALSE       ;true if dual headed (2 heads moving together).
;
;*****************************************************
;*** this is the end of the area which normally need ***
;***     be changed for most console i/o systems     ***
;*****************************************************
;
        IF   TARDEL      ;if using tarbell or delta cpu
CCOM    EQU  IOBASE+1    ;console command port
CSTAT   EQU  IOBASE+1    ;console status port ( chan a.)
CDATA   EQU  IOBASE+0    ;console data port
LCOM    EQU  IOBASE+3    ;list command port
LSTAT   EQU  IOBASE+3    ;list status port (chan b.)
LDATA   EQU  IOBASE+2    ;list data port
        ENDIF
;
        IF      TIMER AND TARBELL       ;must be using tarbell cpu.
;
; timer equates
;
TCH0    EQU     IOBASE+4         ;timer chan 0 address
TCH1    EQU     IOBASE+5         ;timer chan 1 address
TCH2    EQU     IOBASE+6         ;timer chan 2 address
TCMND   EQU     IOBASE+7         ;timer command port
IMASK   EQU     IOBASE+8         ;interrupt masking port
CNTR0   EQU     00000000B        ;counter 0
CNTR1   EQU     01000000B        ;counter 1
CNTR2   EQU     10000000B        ;counter 2
RLWORD  EQU     00110000B        ;read/load lsb 1st, msb 2nd.
RLHBYTE EQU     00100000B        ;read/load msb only.
RLLBYTE EQU     00010000B        ;read/load lsb only.
CNTRLT  EQU     00000000B        ;counter latching operation.
BINARY  EQU     00000000B        ;select binary operation.
BCD     EQU     00000001B        ;select bcd operation.
MODE0   EQU     00000000B        ;interrupt on terminal count.
MODE1   EQU     00000010B        ;programmable one-shot.
MODE2   EQU     00000100B        ;rate generator.
MODE3   EQU     00000110B        ;square wave rate generator.
MODE4   EQU     00001000B        ;software triggered strobe.
MODE5   EQU     00001010B        ;hardware triggered strobe.
        ENDIF
;
        IF   SOLOS               ;if proc tech solos,
CSTAT   EQU  0FAH                ;console status port.
KBD     EQU  0C02EH              ;solos keyboard.
CLRSCR  EQU  0C0D5H              ;clear screen.
SCRN    EQU  0C054H              ;solos output.
        ENDIF
;
        IF NOT SOLOS             ;if not proc tech solos,
DMAP    EQU  0E0H                ;dma base address.
DISK    EQU  0F8H                ;disk base address.
        ENDIF
;
        IF SOLOS                 ;if proc tech solos,
DMAP    EQU  060H                ;dma base address.
DISK    EQU  078H                ;different disk ports.
        ENDIF
```

```
;
ADR0    EQU  DMAP+0             ;dma address reg port.
WCT0    EQU  DMAP+1             ;dma word count reg port.
CMND    EQU  DMAP+8             ;dma command port.
DCOM    EQU  DISK               ;disk command port.
DSTAT   EQU  DISK               ;disk status port.
TRACK   EQU  DISK+1             ;disk track port.
SECTP   EQU  DISK+2             ;disk sector port.
DDATA   EQU  DISK+3             ;disk data port.
WAIT    EQU  DISK+4             ;disk wait port.
DCONT   EQU  DISK+4             ;disk control port.
DMACHK  EQU  DISK+5             ;dma check port.
RTCNT   EQU  10                 ;retry count.
;
        IF   STD                ;if standard i/o,
CKBR    EQU  00000001B          ;keyboard ready bit.
CPTR    EQU  10000000B          ;cons output rdy bit.
        ENDIF
;
        IF   MSIO2              ;if mits 2sio,
CKBR    EQU  00000001B          ;keyboard ready bit.
CPTR    EQU  00000010B          ;print ready bit.
        ENDIF
;
        IF   VDB8024            ;if vdb-8024 board.
CKBR    EQU  00000010B          ;keyboard ready bit.
CPTR    EQU  00000100B          ;cons output rdy bit.
        ENDIF
;
        IF   ISIO2
CKBR    EQU  00000010B          ;keyboard ready bit.
CPTR    EQU  00000001B          ;print ready bit.
        ENDIF
;
        IF   TARDEL
CKBR    EQU  00000010B          ;keyboard ready bit.
CPTR    EQU  00000001B          ;print ready bit.
        ENDIF
;
        IF   TUART              ;if cromemco tuart,
CKBR    EQU  01000000B          ;keyboard ready bit.
CPTR    EQU  10000000B          ;print ready bit.
        ENDIF
;
        IF   SOLOS              ;if proc tech solos,
CKBR    EQU  00000001B          ;keyboard ready bit.
CPTR    EQU  10000000B          ;dummy equ.
        ENDIF
;
        IF   OTHER              ;if something else,
CKBR    EQU  00000010B          ;keyboard ready bit.
CPTR    EQU  10000000B          ;printer ready bit.
        ENDIF
;
LRBIT   EQU  CPTR               ;lister ready bit.
;
IOBYTE  EQU  3                  ;address of i/o byte.
CBASE   EQU  (MSIZE-20)*1024    ;bias for larger than 20k.
CPMB    EQU  CBASE+3400H        ;start of cpm 2.0
BDOS    EQU  CPMB+806H          ;start of bdos 2.0.
BIOS    EQU  CPMB+1600H         ;start of cbios io.
CDISK   EQU  4                  ;location 4 is current disk.
NSECTS  EQU  17                 ;number of sectors in it.
;
        ORG  CPMB+8
;
```

```
SMSG:   DB   0DH,0AH,'Tarbell '
        DB   MSIZE/10+'0',MSIZE MOD 10+'0'
        DB   'K CPM 2.2',0DH,0AH
        DB   'Auto-Select '
        IF   DUBSID
        DB   'Double Sided '
        ENDIF
        DB   'ver of 01-29-82',0
;
; boot
; this section is executed whenever reset and run
; is pushed, after the coldstart loader reads in
; the cpm system.
;
BOOT:   LXI  SP,80H             ;set stack pointer.
;
        IF INTRP AND NOT DMACNTL;if interrupts allowed,
        EI                      ;enable them here.
        ENDIF
;
        IF   MSIO2              ;if mits 2sio,
        MVI  A,3                ;initialize 2sio.
        OUT  CCOM
        OUT  LCOM
        MVI  A,11H
        OUT  CCOM
        OUT  LCOM
        ENDIF
;
        IF   TARDEL OR ISIO2
        LXI  H,IOINIT           ;point to 8251 init. bytes
        MVI  B,4                ;there are 4 of them
INITI0: MOV  A,M                ;get a byte
        OUT  CCOM               ;out to command port of console
        OUT  LCOM               ;out to command port of lister
        INX  H                  ;bump pointer
        DCR  B                  ;decrease count
        JNZ  INITI0             ;loop till done.
        ENDIF
;
        IF   TUART              ;if cromemco tuart,
        MVI  A,1                ;set a = 1.
        OUT  54H                ;select device a.
        OUT  52H                ;reset device b.
        LXI  H,BAUDRS           ;get adr of baud rate table.
        MVI  A,11H              ;octuple the clock.
IT1:    OUT  02H                ;& reset current dev.
        MOV  A,M                ;get baud rate from table.
        INX  H                  ;increment pointer.
        OUT  0                  ;set baud rate.
        CALL CONIN              ;read keyboard.
        CALL CONIN              ;read keyboard again.
        CPI  0DH                ;if not carriage-return,
        MVI  A,1                ;slow the clock.
        JNZ  IT1                ;until a carriage-return.
        ENDIF
;
        IF   SOLOS              ;if proc tech solos,
        CALL CLRSCR             ;clear screen.
        ENDIF
;
        IF DMACNTL
        LXI  H,RWDMA            ;point to dma routine
        SHLD DMAENT+1           ;modify boot jmp address.
        ENDIF
;
        IF   ENTER AND TARBELL  ;if going to tarbell cpm,
```

```
        IF   TIMER AND TARBELL  ;if using tarbell cpu
        MVI  A,CNTR0+RLWORD+MODE2+BINARY        ;init 8253
        OUT  TCMND              ;send it to command port
        LXI  B,33333            ;time constant for 60 hz
        MOV  A,C
        OUT  TCH0               ;ls byte of count
        MOV  A,B
        OUT  TCH0               ;ms byte of count
        ENDIF
;
        JMP  BOOTF              ;finish boot
;
        IF   TARDEL OR ISIO2
IOINIT: DB   0AAH,040H,0CEH,037H
        ENDIF
;
        IF   TUART              ;if cromemco tuart,
BAUDRS: DB   94H,0CEH,0A2H,92H,88H,84H,82H,1
        ENDIF
;
        ORG  BIOS-64            ;hide rest of boot here.
BOOTF:  XRA  A                  ;clear scratch area.
        STA  IOBYTE             ;clear i/o byte.
        STA  CDISK              ;select drive zero
        MVI  B,ENDZ-STARTZ      ;get length of zero area.
        LXI  H,STARTZ           ;get scratch address.
BOOTL:  MOV  M,A                ;put zero in memory.
        INX  H                  ;increment pointer.
        DCR  B                  ;decrement counter.
        JNZ  BOOTL              ;loop till done.
        IN   CDATA              ;clear console status.
        LXI  H,SMSG             ;point to sign on.
PMSG:   MOV  A,M                ;get a byte of the message
        INX  H                  ;bump memory pointer.
        ORA  A                  ;is it a zero?
        JZ   GOCPM              ;yes, we are done, jmp to cpm.
        MOV  C,A                ;nope, print more message.
        CALL CONOT              ;use the conot routine.
        JMP  PMSG               ;and loop till done.
;
        ORG  BIOS               ;start of cbios structure.
;
; i/o jump vector
; this is where cpm calls whenever it needs to do any input/output
; operation. user programs may use these entry points also, but note
; that the location of this vector changes with the memory size.
;
DMAENT: JMP  BOOT               ;from sboot loader,changed for dma.
WBOOTE: JMP  WBOOT              ;from warm boot.
        JMP  CONST              ;check console kb status.
        JMP  CONIN              ;read console character.
        JMP  CONOT              ;write console character.
        JMP  LIST               ;write listing char.
        JMP  PUNCH              ;write punch char.
        JMP  READER             ;read reader char.
        JMP  HOME               ;move disk to track zero.
        JMP  SELDSK             ;select disk drive.
        JMP  SETTRK             ;seek to track in reg a.
        JMP  SETSEC             ;set sector number.
        JMP  SETDMA             ;set disk starting adr.
        JMP  READ               ;read selected sector.
        JMP  WRITE              ;write selected sector.
        JMP  PRSTAT             ;list status check.
        JMP  SECTRAN            ;sector translate routine.
;
; this section defines the the disk parameters
```

```
;
DPBASE  EQU  $                  ;base of disk parameter block
DPE0:   DW   XLT0,0000H         ;translate table
        DW   0000H,0000H        ;scratch area
        DW   DIRBUF,SDTAB+3     ;dir buff, parm block
        DW   CSV0,ALV0          ;check, alloc vectors
;
DPE1:   DW   XLT1,0000H
        DW   0000H,0000H
        DW   DIRBUF,DPB1
        DW   CSV1,ALV1
;
DPE2:   DW   XLT2,0000H
        DW   0000H,0000H
        DW   DIRBUF,DPB2
        DW   CSV2,ALV2
;
DPE3:   DW   XLT3,0000H
        DW   0000H,0000H
        DW   DIRBUF,DPB3
        DW   CSV3,ALV3
;
;the following describes the disk physical nature, such as
;sectors/track,directory size, etc...
;the following table defines a single density drive.
;
SDTAB:  EQU  $                  ;one of 4 disk parm. blocks
        DB   00H                ;log byte single density
        DW   XLT0               ;use single density translate tab.
        DW   26                 ;sectors/track
        DB   3                  ;block shift
        DB   7                  ;block mask
        DB   0                  ;extnt mask
        DW   242                ;disk size - 1
        DW   63                 ;directory max.
        DB   192                ;alloc0
        DB   0                  ;alloc1
        DW   16                 ;check size
        DW   2                  ;number of system tracks
;
        IF   DUBSID             ;is using double sided drives.
;
; defines a single density/ double sided disk
;
        DB   02H                ;log byte doub sided
        DW   XLT0
        DW   26
        DB   4
        DB   15
        DB   0
        DW   242
        DW   95                 ;allow 95 entrys for dir.
        DB   192
        DB   0
        DW   24
        DW   2
        ENDIF
;
;the following table defines a double density drive.
;
DDTAB:  DB   01H                ;log byte doub den/sing sided
        DW   0                  ;no sector translate table.
        DW   51                 ;51 sectors.
        DB   4                  ;block shift.
        DB   15                 ;block mask.
        DB   0                  ;extent mask.
```

```
        DW   237                ;disk size -1
        DW   95                 ;directory max.
        DB   192                ;alloc0
        DB   0                  ;alloc1
        DW   24                 ;check size
        DW   2                  ;number of system tracks.
;
        IF   DUBSID             ;if using double sided drives.
;
; defines a double density/doub sided drive
;
        DB   03H                ;log byte and dub sided
        DW   0
        DW   51
        DB   5
        DB   31
        DB   0
        DW   237
        DW   95
        DB   192
        DB   0
        DW   24
        DW   2
        ENDIF
;
;sector translation table
;
XLT0    EQU  $                  ;start of trans. table
        DB   1,7,13,19,25
        DB   5,11,17,23,3
        DB   9,15,21,2,8
        DB   14,20,26,6,12
        DB   18,24,4,10,16,22
;
DPB1    EQU  SDTAB+3            ;equivalent parameters
XLT1    EQU  XLT0               ;same translate table
;
DPB2    EQU  SDTAB+3
XLT2    EQU  XLT0
;
DPB3    EQU  SDTAB+3
XLT3    EQU  XLT0
;
;disk set up routine. this routine is common to both the
;read and write routines for dma operation. this routine
;may be used stand alone by passing parameters to it and
;jumping to wboot-3 hex. this jump vector is changed when
;cp/m is booted up.
;
;entery point = rwdma:
;
;user must set up dmaadd for memory address and
;user must set up disk sector with 'setsec' entry
;the track to read or write must be set up using 'settrk'
;before using rwdma routine externally.
;
;entry parameters:
;b   = floppy disk (1793) read/write command byte
;c   = floppy disk (1793) force interrupt command byte
;d   = dma (8257) read/write command + high byte count
;e   = dma (8257) low byte count (80 hex = 128 bytes)
;
;exit values
;b,c = floppy commands
;d,e = dma command + byte count.
;h,l = (h,l + d,e)
```

```
;a   = floppy disk status byte
;
;stack usage is 1 level deep.
;
        IF DMACNTL              ;if using dma control
;
DMARW:  STA  ERCNT              ;save error count.
RWDMA:  LDA  SECT               ;get sector to read/write
        OUT  SECTP              ;and send it floppy chip.
        LHLD DMAADD             ;get cpm dma address.
DMARWE: XRA  A                  ;clear accum.
        OUT  CMND               ;reset dma chip.
        MOV  A,C                ;force interrupt command byte
        OUT  DCOM               ;send it to controller.
        MOV  A,E                ;byte count to transfer
        DCR  A                  ;count = count - 1.
        OUT  WCT0               ;send it to dma chip.
        MOV  A,D                ;get read/write code.
        OUT  WCT0               ;and tell dma chip what to do.
        MOV  A,L                ;get low address byte
        OUT  ADR0               ;and send it to dma chip.
        MOV  A,H                ;get high address byte
        OUT  ADR0               ;and send it to dma chip.
        MVI  A,41H              ;set up for request ch. 0
        OUT  CMND               ;send it to controller.
        CALL HDLD               ;check head load bit.
        ORA  B                  ;'or' in the read/write bits.
        OUT  DCOM               ;tell floppy chip what to do.
;
;adjust h,l for 128 byte increase.
;
        PUSH D                  ;save d,e
        MOV  A,D                ;get dma command byte
        ANI  3FH                ;strip off command bits 7 & 6
        MOV  D,A                ;now set for h,l adjust.
        DAD  D                  ;add it to h,l.
        POP  D                  ;restore d,e
;
;general purpose wait routine.
;
        MVI  A,20H              ;count value.
CNTLOOP:DCR  A
        JNZ  CNTLOOP            ;loop till = zero.
SLOOP:  IN   DMACHK             ;check for operation done.
        RLC                     ;by looking at bit 7.
        JC   SLOOP              ;loop till bit 7 = 0.
        IN   DSTAT              ;check and return disk status.
        RET                     ;return to caller.
        ENDIF
;
;warm-boot - read the ccp back into memory. bdos and bios
;assumed still in memory. if they are not, a cold start will
;have to be done to bring them back into memory.
;
WBOOT:  LXI  SP,80H             ;set stack pointer.
;
        IF INTRP AND NOT DMACNTL;if interrupts allowed,
        EI                      ;allow them here.
        ENDIF
;
        IF   LSTPAG             ;if list device paging,
        XRA  A                  ;reset line-feed count.
        STA  LFCNT
        ENDIF
;
        MVI  C,0                ;select disk 0.
```

```
        CALL SELDSK
        CALL HOME               ;move to track zero.
        LXI  H,0                ;clear h,l
        SHLD DRVFLG             ;clear drive flags
        SHLD DRVFLG+2
        MVI  B,NSECTS           ;get # sectors for cpm read.
        MVI  C,2                ;track (b)=0, sector (c)=2.
;
        IF INTRP AND NOT DMACNTL;if interrupts allowed,
        DI                      ;disable them here.
        ENDIF
;
        LXI  H,CPMB             ;get starting address.
RBLK1:  SHLD DMAADD             ;set starting address.
        CALL SETSEC             ;read starting at sector in c.
        PUSH B
        CALL READ               ;read a sector back.
        POP  B
        JNZ  RDERR              ;if error, print message.
        INR  C                  ;increment sector number.
        DCR  B                  ;decrement sector count.
        JNZ  RBLK1              ;not zero, keep reading
;
        IF INTRP AND NOT DMACNTL;if interrupts allowed,
        EI                      ;allow them again here.
        ENDIF
;
; set up jumps into cp/m in lower memory.
;
GOCPM:  MVI  A,0C3H             ;put jmp to wboot
        STA  0                  ;adr at zero.
        LXI  H,WBOOTE           ;warmboot entry point
        SHLD 1                  ;set it.
        STA  5                  ;set jump instruction.
        LXI  H,BDOS             ;put jump to bdos
        SHLD 6                  ;at adr 5,6,7.
        LXI  H,80H              ;set default dma adr.
        SHLD DMAADD             ;save it.
        LDA  CDISK              ;get disk number to
        MOV  C,A                ;pass to ccp in c.
        JMP  CPMB               ;jump to ccp.
;
RDERR:  CALL RECOV              ;we have an error in booting.
        JMP  WBOOT              ;do a warm boot.
;
; check console input status.
;
CONST:  CALL STATCON            ;check console status port.
CONST1: MVI  A,0                ;set a=0 for return.
;
        IF   RDYLO              ;if status ready low,
        RNZ                     ;not ready when not 0.
        ENDIF
;
        IF   RDYHI              ;if status ready high,
        RZ                      ;not ready when zero.
        ENDIF
;
        CMA                     ;if ready a=ff.
        RET                     ;return from const.
;
; statcon - check keyboard status
;
        IF   NOT SOLOS
STATCON:IN   CSTAT              ;in status port
        ANI  CKBR               ;mask ready bit.
```

```
        RET
        ENDIF
;
; read a character from console.
;
CONIN:
        IF   NOT SOLOS          ;if not proc tech solos,
        CALL STATCON            ;read console status.
        ENDIF
;
        IF   SOLOS              ;if proc tech solos,
        CALL KBD                ;read sol keyboard.
        JZ   CONIN              ;ready when not zero.
        ENDIF
;
        IF   RDYLO AND NOT SOLOS
        JNZ  CONIN              ;loop until low.
        ENDIF
;
        IF   RDYHI              ;if ready when high,
        JZ   CONIN              ;loop until high.
        ENDIF
;
        IF   NOT SOLOS          ;if not proc tech solos,
        IN   CDATA              ;read a character.
        ENDIF
;
        ANI  7FH                ;make most sig. bit = 0.
        RET                     ;return from conin.
;
; write a character to the console device.
;
CONOT:
        IF CONUL                ;if nulls required,
        MVI  A,0AH              ;if it's a lf,
        CMP  C                  ;then hop out
        JZ   CONULL             ;to null routine.
        ENDIF
CONOT1:
        IF NOT SOLOS AND NOT VIDEO
        IN   CSTAT              ;read console status.
        ANI  CPTR               ;if not ready,
        ENDIF
;
        IF RDYLO AND NOT SOLOS AND NOT VIDEO
        JNZ  CONOT1             ;loop until low.
        ENDIF
;
        IF RDYHI AND NOT VIDEO  ;if ready when high,
        JZ   CONOT1             ;loop until high.
        ENDIF
;
        IF NOT SOLOS AND NOT VIDEO
        MOV  A,C                ;get character.
        OUT  CDATA              ;print it.
        RET                     ;return.
        ENDIF
;
;this routine calles your video driver routine which must
;be in rom. all registers must be saved and restored by your
;video driver in order to be compatiable with cpm. cpm passes
;the char. to be output in the  c register. make any changes
;in this routine to pass the char from reg c to the register
;your video driver expects it to be in.
;
        IF  VIDEO               ;if using a video driver in rom.
```

```
        MOV  A,C                ;get the cpm char into reg a
        CALL OUTADDR            ;call your video driver.
        RET                     ;return to cpm.
        ENDIF
;
        IF CONUL
CONULL: PUSH B                  ;save b&c.
        MVI  B,CNULL+1          ;get null count.
CONUL1: CALL CONOT1             ;print cr.
        MVI  C,0                ;get null char.
        DCR  B                  ;decrement counter.
        JNZ  CONUL1             ;do next null.
        POP  B                  ;restore b&c.
        MOV  A,C                ;restore a.
        RET                     ;return.
        ENDIF
;
        IF   SOLOS              ;if proc tech solos,
        PUSH B                  ;save b&c.
        MOV  B,C                ;put char in b reg.
        CALL SCRN               ;output char to solos.
        POP  B                  ;restore b&c.
        MOV  A,C                ;put char in a.
        RET                     ;return from conot.
        ENDIF
;
; select disk number according to register c.
;
SELDSK: LXI  H,0                ;set up for error code
        MOV  A,C                ;get new drive.
        CPI  NDISK              ;calling undefined drive ?
        RNC                     ;if no cy, h,l tells cpm yes.
        LXI  H,DISKNO           ;get old drive number.
        MOV  A,M                ;get old disk number.
;
        IF   DUAL               ;if dual drive,
        ANI  0FEH               ;clear out bit 0.
        ENDIF
;
        MOV  E,A                ;put old disk no. in d&e.
        MVI  D,0
        LXI  H,TRTAB            ;get address of track table.
        PUSH H                  ;save address of trtab.
        DAD  D                  ;add disk no. to address.
        IN   TRACK              ;read 1771 track register.
        MOV  M,A                ;put into table.
        MOV  A,C                ;get new disk number.
;
        IF   DUAL               ;if a dual drive,
        ANI  0FEH               ;clear bit 0.
        ENDIF
;
        MOV  E,A                ;put new disk no. in d&e.
        POP  H                  ;restore address of trtab.
        DAD  D                  ;add disk no. to address.
        MOV  A,M                ;get new track number.
        OUT  TRACK              ;put into 1771 track reg.
        MOV  A,C                ;update old disk number.
        STA  DISKNO
        ADD  A                  ;put bits 1&2 at 4&5.
        ADD  A
        ADD  A
        ADD  A
        STA  LATCH              ;save new latch code.
DENSITY:LXI  H,DRVFLG           ;point to drive den. flag
        MVI  B,0                ;clear reg b.
```

```
        DAD   B                 ;index into drive flag loc.
        MOV   A,M               ;get the flag byte
        ORA   A                 ;logged in?
        JM    LOGED             ;yes, it's logged.
        PUSH  H                 ;no, save flag address.
        LDA   LATCH             ;get latch code
        OUT   DCONT             ;change latch or density
;
;read track 0 sector 1 for density byte at 7e hex.
;
        MVI   A,1               ;sector 1.
        STA   SECT              ;save the sector value.
        CALL  HOME              ;home the drive.
        LHLD  DMAADD            ;get cp/m dma address value
        PUSH  H                 ;save it on the stack.
        LXI   H,DBUFF           ;point to the dma buffer.
        SHLD  DMAADD            ;set up read dma address.
;
;read the data using read routine.
;
        CALL  READ              ;cbios read routine.
;
;get density byte value and determine drive status.
;
        POP   H                 ;restore dma address from the stack.
        SHLD  DMAADD            ; and restore the cp/m dma address.
        POP   H                 ;restore density flag address.
        LDA   DBUFF+7EH         ;index into dbuff to location dbuff+7e.
        ORI   80H               ;set logged in bit
        MOV   M,A               ;place it in flag table.
LOGED:  LXI   B,18              ;index value through drive table.
        ANI   12H               ;mask density and side bits out.
        ORA   A                 ;single density?
        LXI   H,SDTAB           ;point to start of tables
        JZ    DENSIT1           ;yes, overlay param. block
;
        IF    DUBSID
        DAD   B                 ;no, add offset to next table
        CPI   2                 ;single den doub sided?
        JZ    DENSIT1           ;yes
        DAD   B                 ;no
        CPI   10H               ;doub den single sided?
        JZ    DENSIT1           ;yes
        ENDIF
;
        DAD   B                 ;no, must be doub den , doub sided
DENSIT1:XCHG                    ;drive table pointer --½ d,e
        LDAX  D                 ;get log and drive type byte.
        INX   D                 ;bump pointer
        STA   DENS              ;set current drive density.
        PUSH  D                 ;save drive table pointer.
        CALL  PARINDX           ;compute parameter overlay area.
        POP   D                 ;restore drive table pointer.
        LXI   B,0208H           ;b = 2, c = 8 (count values).
MOVE:   LDAX  D                 ;get xlt0 byte.
        MOV   M,A               ;and put it into dw table for drive.
        INX   D                 ;bump
        INX   H                 ; pointers
        DCR   B                 ;decrease count.
        JNZ   MOVE              ; and loop till zero.
        DAD   B                 ;now add index into dpb0 area.
        MOV   M,E               ;get low pointer byte.
        INX   H                 ;bump pointer.
        MOV   M,D               ;get high pointer byte.
;
;select drive as a function of h,l
```

```
;
PARINDX:LHLD DISKNO             ;load disk number and zero byte
        LXI  D,DPBASE           ;point to disk parm start.
        DAD  H                  ;*2
        DAD  H                  ;*4
        DAD  H                  ;*8
        DAD  H                  ;*16
        DAD  D                  ;compute index for the drive
        XRA  A                  ;set a = 0.
        RET                     ;return from seldsk.
;
; move disk to track zero.
;
HOME:   MVI  C,0                ;seek to track zero.
        MVI  A,STPRAT           ;restore command
        OUT  DCOM               ;tell controller.
;
; set track number to whatever is in register c.
; also perform move to the correct track (seek).
;
SETTRK:
        LHLD LATCH              ;get new and old latch.
        MOV  A,H                ;get latch value.
        ANI  0B7H               ;strip density and side bits.
        MOV  H,A                ;restore it.
;
        IF   DUBSID             ;if using double sided drive.
        LDA  DENS               ;check if double sided.
        RRC
        RRC                     ;look at bit 1.
        JNC  NOTSID             ;if bit 1 = 0, it's single sided.
        MOV  A,C                ;it's doub sided, so get track number.
        RRC                     ;divide by 2.
        MOV  B,A                ;save it in reg b.
        MOV  A,L                ;get old latch value.
        JC   SIDE2              ;change side if odd track.
        ANI  0BFH               ;clear side bit from latch.
        JMP  SETLAT             ;go set the latch.
;
SIDE2:  ORI  40H                ;turn on side select bit.
SETLAT: STA  CLATCH             ;save it for later.
        ANI  0BFH               ;clear side bit.
        CALL OLDLAT             ;check for drive change.
        MOV  A,B                ;restore doub sided trk number.
        ANI  7FH                ;clear bit 7.
        MOV  C,A                ;trk number now in reg c.
        JMP  TRKSET             ;check for density of track going to.
        ENDIF
;
        IF   NOT DUBSID         ;if not using double sided drive
        JMP  NOTSID             ;jump around subroutine.
        ENDIF
;
OLDLAT: CMP  H                  ;new = old?
        MVI  A,0FFH             ;if not, set = ff
        JNZ  SFLAG
        CMA                     ;new = old, set = 0.
SFLAG:  STA  HLSF               ;save head load select flag.
        RET
;
NOTSID: MOV  A,L                ;get latch value.
        STA  CLATCH             ;save it
        CALL OLDLAT             ;check for drive change.
TRKSET: LDA  DENS               ;check drive density flag.
        RRC                     ;is bit 0 = 0?
        JNC  TRKSD              ;yes, we are single density.
```

```
        MOV  A,C                ;no, restore track number.
        CPI  1                  ;is it track 1?
        JC   TRKSD              ;if less than, set single density.
        LDA  CLATCH             ;get current latch value.
        ORI  8                  ;set for double density.
        JMP  TRKDD
;
TRKSD:  LDA  CLATCH             ;get current latch value.
        ANI  0F7H               ;turn off bit 4 (single density).
TRKDD:  STA  CLATCH             ;save new latch value.
        OUT  DCONT              ;select disk and make density change.
        MOV  A,C                ;restore track value
        STA  TRK                ;update old with new.
;
; move the head to the track in register a.
;
SEEK:   PUSH B                  ;save b&c.
        MOV  B,A                ;save destination track.
        MVI  A,RTCNT            ;get retry count.
SRETRY: STA  SERCNT             ;store in error counter.
        IN   TRACK              ;read present track no.
        CMP  B                  ;same as new track no.?
        JNZ  NOTHR              ;jump if not there.
THERE:  POP  B                  ;restore b&c.
        RET                     ;return from seek.
NOTHR:  MOV  A,B                ;restore a from b.
        OUT  DDATA              ;track to data register.
        MVI  A,14H+STPRAT+HLAB  ;get step rate, do
        OUT  DCOM               ;seek with verify.
;
        IF NOT DMACNTL
        IN   WAIT               ;wait for intrq.
        IN   DSTAT              ;read status.
        ENDIF
;
        IF DMACNTL
        CALL SLOOP              ;no wait status check.
        ENDIF
;
        ANI  91H                ;look at bits.
        JZ  THERE               ;ok if zero.
        LDA  SERCNT             ;get error count.
        DCR  A                  ;decrement count.
        JNZ  SRETRY             ;retry seek.
        POP  B                  ;restore b&c.
        PUSH B                  ;save
        CALL RECOV              ;if seek retry = 10 check
        POP  B
        MOV  A,C                ;recover track number.
        JMP  SEEK               ; for cntl-c for abort.
;
; set disk sector number.
;
SETSEC:  MOV  A,C               ;get sector number.
         STA  SECT              ;put at sect # address.
         RET                    ;return from setsec.
;
;translate the sector given b,c using
;the translate table;given by d,e
;
SECTRAN:
        MOV  L,C                ;get physical sector number
        INR  L                  ;bump it by one.
        MOV  A,D                ;are we using no xlat table?
        ORA  E                  ;it will be zero if not.
        RZ                      ;return if it is zero.
```

```
        XCHG                    ;h,l = trans
        DAD  B                  ;h,l = trans (sector)
        MOV  L,M                ;l   = trans (sector)
        MVI  H,0                ;clear reg h
        RET                     ;h,l = translated sector
;
; set disk dma address.
;
SETDMA: MOV  H,B                ;move b&c to h&l.
        MOV  L,C
        SHLD DMAADD             ;put at dma adr address.
        RET                     ;return from setdma.
;
; hdld - get head-load bit if required.
;
HDLD:   LDA  HLSF               ;get head-load flag.
        ORA  A                  ;is a = zero?
        JZ   HDLD1              ;hop if so.
        CMA                     ;set a = 0.
        STA  HLSF               ;set flag = 0 if not.
;
;if changing to a new drive, perform a seek to
;the same track to unload the head on new drive.
;
        IN   TRACK              ;get present track
        OUT  DDATA              ;tell controller.
        MVI  A,10H+STPRAT
        OUT  DCOM
;
        IF   NOT DMACNTL
        IN   WAIT               ;wait for intrq.
        ENDIF
;
        IF   DMACNTL
        CALL SLOOP              ;check dma status port.
        ENDIF
;
HDLD1:  IN   DSTAT              ;read 1771 status.
        ANI  20H                ;look at hl bit.
        MVI  A,4
        RZ                      ;return if head is not loaded.
        SUB  A                  ;head is already loaded.
        RET                     ;return from hdld.
;
; read the sector at sect, from the present track,
; use starting address at dmaadd.
;
READ:   MVI  A,RTCNT            ;get retry count.
RRETRY:
        IF DMACNTL
        LXI  B,80D0H            ;floppy read, force interrupt.
        LXI  D,4080H            ;dma read, dma count byte
        CALL DMARW              ;enter common read/write routine.
        ENDIF
;
        IF NOT DMACNTL
        MVI  B,80H              ;floppy read command byte.
        CALL DSKSET             ;set up disk controller.
        ORA  B                  ;'or' in the read command.
READE:  OUT  DCOM               ;send command to 1771.
RLOOP:  IN   WAIT               ;wait for drq or intrq.
        ORA  A                  ;set flags.
        JP   RDDONE             ;done if intrq.
        IN   DDATA              ;read a data byte from disk.
        MOV  M,A                ;put byte into memory.
        INX  H                  ;increment memory pointer.
        JMP  RLOOP              ;keep reading.
```

```
        JMP  RLOOP              ;keep reading.
RDDONE: IN   DSTAT              ;read disk status.
        ENDIF
;
        IF INTRP AND NOT DMACNTL;if interrupts allowed,
        EI                      ;allow again here.
        ENDIF
;
        ANI  9DH                ;look at error bits.
        RZ                      ;return if none.
        CALL ERCHK              ;check for seek error.
        JNZ  RRETRY             ;try to read again.
        CALL RECOV              ;check for abort or continue
        JMP  READ               ;if not cntl-c, try to read again.
;
;recov
;this routine is called by any read,write,seek routine if the retry
;count goes to 10. if it does,this routine calls conin for a key to
;be pushed. if the key is a cntl-c, then a warmboot is executed. if
;any other key is pushed, then a return is made back to the caller
;and that routine is retried for 10 more times.
;
RECOV:
        MVI  C,'e'              ;error code
        CALL CONOT              ;print it
        CALL CONIN              ;check for pushed key.
        CPI  03H                ;is it a cntl-c ?
        RNZ                     ;return to caller if not.
        JMP  WBOOT              ;yes, do warmboot.
;
; erchk - check for record not found error.
;
ERCHK:  ANI  10H                ;if record not found,
        JNZ  CHKSK              ;do a check on seek.
CHKOK:  LDA  ERCNT              ;get retrys allowed
        DCR  A                  ;decrease it,
        RET                     ;and return with number.
;
;check for seek to correct track,
;and change if necessary.
;
        IF NOT DMACNTL
CHKSK:  MVI  A,0C4H             ;send command to 1771
        OUT  DCOM               ;to read address.
        IN   WAIT               ;wait for drq or intrq.
        IN   DDATA              ;read the track address.
        PUSH PSW                ;save it on the stack.
CHKS2:  IN   DMACHK             ;wait for intrq.
        ORA  A                  ;set flags.
        JP   CHKS3              ;done with read adr op.
        IN   DDATA              ;read another byte.
        JMP  CHKS2              ;do it again.
CHKS3:  IN   DSTAT              ;read disk status.
        ENDIF
;
        IF   DMACNTL
CHKSK:  LXI  H,BIOS-7           ;point to unused space
        LXI  B,0C4D0H           ;read address, force interrupt cmnds.
        LXI  D,04006H           ;dma read, count byte
        CALL DMARWE             ;read the id using dma control.
        ORA  A                  ;set flags.
        JZ   CHKS4              ;read adr ok if 0.
        CALL HOME               ;otherwise, home first.
        JMP  CHKS5
        ENDIF
;
```

```
        IF   NOT DMACNTL
CHKS4:  POP  PSW                ;update track register.
        ENDIF
;
        IF   DMACNTL
CHKS4:  IN   SECTP              ;get the track byte
        ENDIF
;
        OUT  TRACK
CHKS5:  LDA  TRK                ;get required track no.
        CALL SEEK               ;move the head to it.
        JMP  CHKOK              ;exit from error check.
;
        IF   NOT DMACNTL
DSKSET: STA  ERCNT              ;store in error ctr.
        MVI  A,0D0H             ;cause interrupt.
        OUT  DCOM
        XTHL                    ;some
        XTHL                    ; delay
        ENDIF
;
        IF INTRP AND NOT DMACNTL;if interrupts allowed,
        DI                      ;disable them here.
        ENDIF
;
        IF   NOT DMACNTL
        LHLD DMAADD             ;get starting addr.
        LDA  SECT               ;get sector number.
        OUT  SECTP              ;set sector into 1771.
        CALL HDLD               ;get head-load bit?
        RET                     ;return to caller
        ENDIF
;
; write the sector at sect, on the present track,
; use starting address at dmaadd.
;
WRITE:  MVI  A,RTCNT            ;get retry count.
WRETRY:
        IF   DMACNTL
        LXI  B,0A0D0H           ;floppy write, force interrupt.
        LXI  D,08080H           ;dma write, dma count byte.
        CALL DMARW              ;enter common read/write routine.
        ENDIF
;
        IF   NOT DMACNTL
        MVI  B,0A0H             ;floppy write command byte.
        CALL DSKSET             ;set up floppy controller.
        ORA  B                  ;'or' in write command.
WRITE2: OUT  DCOM
WLOOP:  IN   WAIT               ;wait for ready.
        ORA  A                  ;set flags.
        JP   WDONE              ;hop out when done.
        MOV  A,M                ;get byte from mem.
        OUT  DDATA              ;write onto disk.
        INX  H                  ;increment mem ptr.
        JMP  WLOOP              ;keep writing.
WDONE:  IN   DSTAT              ;read disk status.
        ENDIF
;
        IF INTRP AND NOT DMACNTL;if interrupts allowed,
        EI                      ;enable again here.
        ENDIF
;
        ANI  0FDH               ;look at these bits.
        RZ                      ;return if no err.
        CALL ERCHK              ;check/correct seek err.
```

```
        JNZ  WRETRY             ;try to write again.
        CALL RECOV              ;check for abort
        JMP  WRITE              ;retry write again.
;
;list status check routine
;
PRSTAT: CALL PSTAT              ;check printer status port.
;
        MVI  A,0                ;return status activity.
;
        IF   TARDEL OR RDYLO
        RNZ
        ENDIF
;
        IF   NOT TARDEL AND RDYHI
        RZ
        ENDIF
;
        CMA                     ;invert it
;
; punch and reader are not supported.
;
PUNCH:
READER: RET
;
;pstat - printer status check routine.
;
PSTAT:  IN   LSTAT              ;read printer status port.
;
        IF   NOT TARDEL
        ANI  LRBIT
        ENDIF
;
        IF   TARDEL
        ANI  81H                ;mask ready bits
        XRI  81H
        ENDIF
;
        RET                     ;return to caller
;
; write a character on listing device.
;
LIST:
        IF   LSTNUL             ;if nulls or paging,
        MVI  A,0DH              ;if it's a cr,
        CMP  C                  ;then hop out to
        JZ   LINUL              ;null routine.
        ENDIF
;
        IF   LSTPAG             ;if paging
        MVI  A,0AH              ;get a linefeed
        CMP  C                  ;does it match?
        JZ   LINUL3
        MOV  A,C
        CPI  0CH
        RZ
        ENDIF
;
LTBSY:  CALL PSTAT              ;read lister status.
;
        IF   TARDEL OR RDYLO
        JNZ  LTBSY              ;loop till low.
        ENDIF
;
        IF   NOT TARDEL AND RDYHI
        JZ   LTBSY              ;loop till high.
```

```
        ENDIF
;
        MOV  A,C                ;get data byte.
        OUT  LDATA              ;print it.
        RET                     ;return from list.
;
        IF   LSTNUL             ;if list nulls
LINUL:  PUSH B                  ;save b&c.
        MVI  B,(LNULL AND 0FFH)+1 ;get null count
LINUL1: CALL LTBSY              ;print (cr first).
        MVI  C,0                ;get null char.
        DCR  B                  ;decrement counter.
        JNZ  LINUL1             ;do next null.
        JMP  LINUL2             ;exit the routine.
        ENDIF
;
        IF   LSTPAG             ;if list dev. paging,
LINUL3: PUSH B                  ;save b,c pair
        LDA  LFCNT              ;get line-feed count.
        INR  A                  ;increment it.
        STA  LFCNT              ;save it back.
        CPI  LINCNT-(LINCNT/11) ;end of page?
        MVI  B,1                ;set up for 1 lf.
        JNZ  NOTEOP             ;hop if not end.
        XRA  A                  ;set lf count = 0.
        STA  LFCNT
        MVI  B,(LINCNT/11)+1    ;between pages.
NOTEOP: MVI  C,0AH              ;get line-feed code.
LSTPA1: CALL LTBSY              ;print line-feed.
        DCR  B                  ;decrement lf counter.
        JNZ  LSTPA1             ;do next line feed?
        ENDIF
;
        IF   LSTNUL OR LSTPAG   ;if nulls or paging,
LINUL2: POP  B                  ;restore b&c.
        MOV  A,C                ;restore a.
        RET                     ;return from list.
        ENDIF
;
ENDPROG EQU     $-1             ;ending address.
;
;note:  as there are only six (6) sectors available for cbios on
;the second system track (1), the last address before this point
;should be no greater than the cbios starting address + 037f (hex).
;this will normally be xd7f (hex).
;
; bios scratch area.
;
TRK:    DS   1                  ;current track number.
SECT:   DS   1                  ;current sector number.
DMAADD: DS   2                  ;disk transfer address.
;
; the next several bytes, between startz and
; endz, are set to zero at cold boot time.
;
STARTZ:                         ;start of zeroed area.
;
DISKNO: DS   2                  ;disk number
;
; special flags.
;
HLSF:   DS   1                  ;head-load select flag.
LFCNT:  DS   1                  ;paging line-feed count.
;
; trtab - disk track table - present position of
;       heads for up to 4 drives.
```

```
;
TRTAB:  DS    4
DRVFLG: DS    4                 ;drive density flags.
DENS:   DS    1                 ;current drive density value.
ERCNT:  DS    1                 ;error count for retries.
SERCNT: DS    1                 ;seek retry counter.
LATCH:  DS    1                 ;new code for latch.
CLATCH: DS    1                 ;current code in latch.
;
ENDZ    EQU   $
;
BEGDAT  EQU   $
DIRBUF: DS    128               ;directory buffer
ALV0:   DS    31
CSV0:   DS    24
ALV1:   DS    31
CSV1:   DS    24
ALV2:   DS    31
CSV2:   DS    24
ALV3:   DS    31
CSV3:   DS    24
ENDDAT  EQU   $
DATSIZ  EQU   $-BEGDAT          ;total size of disk parm storage.
;
DBUFF:  DS    128               ;128 byte density select buffer.
;
        ORG   ENDPROG           ;show actual ending address of bios
        END
```

A½

```
A½
;
; tarbell electronics cp/m coldstart loader
; version of 10-27-81.
;-------------------------------------------------------------
; modified for dma control - 1-5-80.
; modified for reading larger bios from trk 1  - 6-5-80.
; modified for tarbell cpu card  -  7-3-80.
; modified to clear extended latch on disk board 6-29-81
; modified for doub den side 1 system track 9-29-81
; added larger sector check in seccmp + 1 lable  10-27-81
; g.w.mulchin
; tarbell electronics
;-------------------------------------------------------------
; copyright (c) 1980, 1981 tarbell electronics
;
;
;*************************************************************
;*                                                           *
;*                       ** note **                          *
;*                      =============                        *
;*                                                           *
;*      the equate for double density (doubden) must only be *
;* set true for a disk which is formatted in double density only *
;* and one which you wish to put an operating system on to.  *
;* otherwise, leave it false if you are building an operating *
;* system on to a single density formatted disk.             *
;*                                                           *
;*************************************************************
;
;
; this program is loaded at location zero by the bootstrap program,
; and executed. its purpose is to load and execute the cp/m disk
; operating system at the top of the memory in use.
;
FALSE   EQU  0          ;define value of false.
TRUE    EQU  NOT FALSE  ;define value of true.
;
;********* this is the area to make changes in ******************
;********* for different system configurations ******************
;                                                             **
MSIZE   EQU  64         ;memory size in decimal kb.         **
TARBELL EQU  FALSE      ;true if using tarbell cpu.         **
DUBSID  EQU  FALSE      ;true for double sided systems.     **
DELTA   EQU  FALSE      ;true if using delta cpu card       **
DOUBDEN EQU  FALSE      ;true if doub. den disk.            **
DMACNTL EQU  TRUE       ;true if using dma control          **
BASE    EQU  0          ;tarbell i/o ports (00 or 10 hex)   **
DDS     EQU  26         ;sectors in trk 1 , (range = 26 to 51)  **
DISK    EQU  0F8H       ;disk port base address.            **
;                                                             **
;*************************************************************
;
        IF   TARBELL
IO      EQU  BASE       ;i/o ports on tarbell cpu.
MMENB   EQU  IO+10      ;memory management enable port.
MEMMAG  EQU  BASE+32    ;memory management port.
        ENDIF

ADR0    EQU  0E0H       ;dma address port.
WCT0    EQU  0E1H       ;dma word count port.
CMND    EQU  0E8H       ;dma command port.
DCOM    EQU  DISK       ;command port.
DSTAT   EQU  DISK       ;status port.
```

```
TRACK   EQU  DISK+1      ;track port.
SECT    EQU  DISK+2      ;sector port.
DATA    EQU  DISK+3      ;data port.
WAIT    EQU  DISK+4      ;wait port.
DCONT   EQU  DISK+4      ;control port.
DMACHK  EQU  DISK+5      ;dma check port.
EXTMEM  EQU  DISK+5      ;extended disk latch
PANEL   EQU  0FFH        ;front panel machines.
CBASE   EQU  (MSIZE-20)*1024
CPMB    EQU  CBASE+3400H;start of cp/m.
BOOTE   EQU  CPMB+1600H ;cold boot entry point.
SPT     EQU  26          ;number of sectors per track. (sing den)
SDS     EQU  25          ;always 25 sectors to read in trk 1.
NSECTS  EQU  SDS + DDS   ;sectors of cp/m.
RTCNT   EQU  10          ;number of retrys.
;
        ORG  0           ;start of loader.
;
BOOT:
        IF   TARBELL     ;if using tarbell cpu
        LXI  D,1000H     ;count=16, data byte = 0
        MVI  C,MEMMAG AND 0FFH
MLOOP:  MOV  A,E         ;get address value
        ORA  C           ;make i/o port value
        STA  MOUT+1      ;modify port on the fly
        MOV  A,E         ;get init value.
        CMA              ;flip it for ram on cpu
MOUT:   OUT  BASE        ;put it to ram on cpu
        INR  E           ;bump data value
        DCR  D           ;decrease count
        JNZ  MLOOP       ;loop 16 times.
        OUT  MMENB       ;enable memory management.
        ENDIF
;
        IF   DELTA       ;if using delta cpu.
        MVI  A,1         ;get a 1 in reg a.
        OUT  9           ; and disable cpu rom slot.
        ENDIF
;
        XRA  A           ;clear accum
        OUT  EXTMEM      ;clear extended disk latch
        MVI  E,RTCNT     ;get retry count.
BLOOP:  LXI  SP,100H     ;set stack pointer.
        LXI  H,CPMB      ;cp/m starts here.
        MVI  D,NSECTS    ;number of sectors to read.
        MVI  C,2         ;sector number.
RNTRK:  MVI  B,4         ;for head load.
RNSEC:  CALL READ        ;read first sector.
        DCR  D           ;if done,
        JZ   BOOTE       ;go to cp/m.
        MVI  B,0         ;for no head load.
        INR  C           ;increment sector count.
        MOV  A,C         ;done with
SECCMP: CPI  SPT+1       ;this track?
        JC   RNSEC       ;if not, read next sector.
;
        IF   DOUBDEN AND NOT DUBSID
        MVI  A,DDS + 1   ;number of sectors to read on trk 1.
        STA  SECCMP+1    ;modify sector compare value.
        MVI  A,8         ;get set double density code
        OUT  WAIT        ;set latch for d.density
        ENDIF
;
        IF   NOT DUBSID
        MVI  A,5BH       ;step command.
        OUT  DCOM        ;issue it.
```

```
        IN   WAIT       ;wait until done.
        ENDIF
;
        IF   DUBSID     ;if double sided system.
        MVI  A,DDS + 1  ;number of sectors to read on trk 1.
        STA  SECCMP+1   ;modify sector compare value.
        MVI  A,48H      ;side select and density select
        OUT  DCONT      ;issue it.
        ENDIF
;
        MVI  C,1        ;sector number.
        JMP  RNTRK      ;read next track.
;
READ:
        IF   DMACNTL    ;if using dma control.
        MVI  A,41H      ;set up for chan 0 req.
        OUT  CMND
        MVI  A,7FH      ;count for 128 bytes
        OUT  WCT0
        MVI  A,40H      ;read command
        OUT  WCT0
        MOV  A,L        ;get low address byte
        OUT  ADR0
        MOV  A,H        ;high address byte
        OUT  ADR0
        ENDIF
;
        MOV  A,C        ;sector in a.
        OUT  SECT       ;set sector register.
        MVI  A,88H      ;command for read.
        ORA  B          ;get head load bit.
        OUT  DCOM       ;issue command.
;
        IF   NOT DMACNTL;if not using dma control.
RLOOP:  IN   WAIT       ;wait for drq.
        ORA  A          ;set flags.
        JP   CHECK      ;jump if done.
        IN   DATA       ;read data.
        MOV  M,A        ;put in memory.
        INX  H          ;increment pointer.
        JMP  RLOOP      ;loop until done.
        ENDIF
;
        IF  DMACNTL
SLOPP:  IN   DMACHK     ;check dma status
        RLC             ; bit 7
        JC   SLOPP      ;loop if carry
        PUSH B          ;save b,c pair
        LXI  B,128      ;count set for 128 bytes
        DAD  B          ;adjust h,l by 128 bytes
        POP  B          ;restore bc
        ENDIF
;
CHECK:  IN   DSTAT      ;read status.
        ANI  9DH        ;look at error bits.
        RZ              ;ok if zero.
        DCR  E          ;decrement retry count.
        JNZ  BLOOP      ;try again if not zero.
        CMA             ;flip for front panel
        OUT  PANEL      ;show error code from floppy.
HERE:   JMP  HERE       ;loop.
;
        ORG  7DH        ;put jump here.
;
        IF   DOUBDEN AND NOT DUBSID     ;if running double density
        RST  0          ;do restart 0
```

```
        DB    0DDH          ;this byte must be here if doub den.
        DB    0             ;this byte unused
        ENDIF
;
        IF    DOUBDEN AND DUBSID
        RST   0
        DB    0DFH
        DB    0
        ENDIF
;
        IF    NOT DOUBDEN AND NOT DUBSID
        RST   0             ;do warm boot with rst inst.
        DB    0E5H
        DB    0
        ENDIF
;
        IF    NOT DOUBDEN AND DUBSID
        RST   0
        DB    0E7H
        DB    0
        ENDIF
;
        END   BOOT          ;end of boot.
```

A½