C. Character Generator ROM Listing

```
                        .Z80
                        ;
                        ;       PIED PIPER CHARACTER SET PROM
                        ;
                        ;       OCTOBER 19, 1982
                        ;
                        ;       EACH CHARACCTER BEGINS WITH A LABEL HXY:
                        ;       WHERE XY IS THE HEX VALUE OF THE CHARACTER
                        ;
0000'                   H00:
0000'  10 10 10 10              DB      10H,10H,10H,10H,1FH,00H,00H,00H
0004'  1F 00 00 00
0008'  00 00 00 00              DB      00H,00H,00H,00H,00H,00H,00H,00H
000C'  00 00 00 00
0010'                   H01:
0010'  00 00 00 00              DB      00H,00H,00H,00H,1FH,10H,10H,10H
0014'  1F 10 10 10
0018'  10 00 00 00              DB      10H,00H,00H,00H,00H,00H,00H,00H
001C'  00 00 00 00
0020'                   H02:
0020'  10 10 10 10              DB      10H,10H,10H,10H,0F0H,00H,00H,00H
0024'  F0 00 00 00
0028'  00 00 00 00              DB      00H,00H,00H,00H,00H,00H,00H,00H
002C'  00 00 00 00
0030'                   H03:
0030'  00 00 00 00              DB      00H,00H,00H,00H,0F0H,10H,10H,10H
0034'  F0 10 10 10
0038'  10 00 00 00              DB      10H,00H,00H,00H,00H,00H,00H,00H
003C'  00 00 00 00
0040'                   H04:
0040'  10 10 10 10              DB      10H,10H,10H,10H,0FFH,00H,00H,00H
0044'  FF 00 00 00
0048'  00 00 00 00              DB      00H,00H,00H,00H,00H,00H,00H,00H
004C'  00 00 00 00
0050'                   H05:
0050'  00 00 00 00              DB      00H,00H,00H,00H,0FFH,10H,10H,10H
0054'  FF 10 10 10
0058'  10 00 00 00              DB      10H,00H,00H,00H,00H,00H,00H,00H
005C'  00 00 00 00
0060'                   H06:
0060'  10 10 10 10              DB      10H,10H,10H,10H,1FH,10H,10H,10H
0064'  1F 10 10 10
0068'  10 00 00 00              DB      10H,00H,00H,00H,00H,00H,00H,00H
006C'  00 00 00 00
0070'                   H07:
0070'  10 10 10 10              DB      10H,10H,10H,10H,0F0H,10H,10H,10H
0074'  F0 10 10 10
0078'  10 00 00 00              DB      10H,00H,00H,00H,00H,00H,00H,00H
007C'  00 00 00 00
0080'                   H08:
0080'  E0 80 80 E8              DB      0E0H,080H,080H,0E8H,030H,02EH,028H,08H
0084'  30 2E 28 08
0088'  0E 00 00 00              DB      0EH,00H,00H,00H,00H,00H,00H,00H
008C'  00 00 00 00
0090'                   H09:
0090'  FF AB D7 AB              DB      0FFH,0ABH,0D7H,0ABH,0D7H,0ABH,0D7H,0ABH
```

```
0094'  D7 AB D7 AB
0098'  FF 00 00 00                DB    0FFH,00H,00H,00H,00H,00H,00H,00H
009C'  00 00 00 00
00A0'                        H0A:
00A0'  10 10 10 10                DB    10H,10H,10H,10H,0FFH,10H,10H,10H
00A4'  FF 10 10 10
00A8'  10 00 00 00                DB    10H,00H,00H,00H,00H,00H,00H,00H
00AC'  00 00 00 00
00B0'                        H0B:
00B0'  10 10 10 10                DB    10H,10H,10H,10H,10H,10H,10H,10H
00B4'  10 10 10 10
00B8'  10 00 00 00                DB    10H,00H,00H,00H,00H,00H,00H,00H
00BC'  00 00 00 00
00C0'                        H0C:
00C0'  00 00 00 00                DB    00H,00H,00H,00H,0FFH,00H,00H,00H
00C4'  FF 00 00 00
00C8'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
00CC'  00 00 00 00
00D0'                        H0D:
00D0'  00 7C 00 00                DB    00H,7CH,00H,00H,00H,00H,00H,00H
00D4'  00 00 00 00
00D8'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
00DC'  00 00 00 00
00E0'                        H0E:
00E0'  00 00 08 7C                DB    00H,00H,08H,07CH,10H,07CH,20H,00H
00E4'  10 7C 20 00
00E8'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
00EC'  00 00 00 00
00F0'                        H0F:
00F0'  00 00 00 00                DB    00H,00H,00H,00H,10H,28H,10H,00H
00F4'  10 28 10 00
00F8'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
00FC'  00 00 00 00
0100'                        H10:
0100'  A0 C0 80 A8                DB    0A0H,0C0H,080H,0A8H,038H,02EH,08H,0CH
0104'  38 2E 08 0C
0108'  08 00 00 00                DB    08H,00H,00H,00H,00H,00H,00H,00H
010C'  00 00 00 00
0110'                        H11:
0110'  00 00 10 00                DB    00H,00H,10H,00H,7CH,00H,10H,00H
0114'  7C 00 10 00
0118'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
011C'  00 00 00 00
0120'                        H12:
0120'  00 60 18 04                DB    00H,60H,18H,04H,18H,60H,00H,7CH
0124'  18 60 00 7C
0128'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
012C'  00 00 00 00
0130'                        H13:
0130'  00 0C 30 40                DB    00H,0CH,30H,40H,30H,0CH,00H,7CH
0134'  30 0C 00 7C
0138'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
013C'  00 00 00 00
0140'                        H14:
0140'  00 38 20 20                DB    00H,38H,20H,20H,20H,20H,20H,20H
0144'  20 20 20 20
```

```
0148'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
014C'  00 00 00 00
0150'                        H15:
0150'  00 20 20 20                DB    00H,20H,20H,20H,20H,20H,20H,38H
0154'  20 20 20 38
0158'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
015C'  00 00 00 00
0160'                        H16:
0160'  00 7C 44 44                DB    00H,7CH,44H,44H,44H,44H,44H,7CH
0164'  44 44 44 7C
0168'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
016C'  00 00 00 00
0170'                        H17:
0170'  00 10 10 10                DB    00H,10H,10H,10H,10H,10H,10H,7CH
0174'  10 10 10 7C
0178'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
017C'  00 00 00 00
0180'                        H18:
0180'  00 10 38 54                DB    00H,10H,38H,54H,10H,10H,10H,10H
0184'  10 10 10 10
0188'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
018C'  00 00 00 00
0190'                        H19:
0190'  00 10 10 10                DB    00H,10H,10H,10H,10H,54H,38H,10H
0194'  10 54 38 10
0198'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
019C'  00 00 00 00
01A0'                        H1A:
01A0'  00 00 08 04                DB    00H,00H,08H,04H,7EH,04H,08H,00H
01A4'  7E 04 08 00
01A8'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
01AC'  00 00 00 00
01B0'                        H1B:
01B0'  00 00 10 20                DB    00H,00H,10H,20H,7EH,20H,10H,00H
01B4'  7E 20 10 00
01B8'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
01BC'  00 00 00 00
01C0'                        H1C:
01C0'  00 20 10 10                DB    00H,20H,10H,10H,10H,10H,10H,08H
01C4'  10 10 10 08
01C8'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
01CC'  00 00 00 00
01D0'                        H1D:
01D0'  00 00 00 78                DB    00H,00H,00H,78H,04H,04H,04H,78H
01D4'  04 04 04 78
01D8'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
01DC'  00 00 00 00
01E0'                        H1E:
01E0'  00 00 10 10                DB    00H,00H,10H,10H,28H,28H,44H,7CH
01E4'  28 28 44 7C
01E8'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
01EC'  00 00 00 00
01F0'                        H1F:
01F0'  00 00 7C 44                DB    00H,00H,7CH,44H,28H,28H,10H,10H
01F4'  28 28 10 10
01F8'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
```

```
01FC'  00 00 00 00
0200'                          H20:
0200'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
0204'  00 00 00 00
0208'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
020C'  00 00 00 00
0210'                          H21:
0210'  00 10 10 10                    DB      00H,10H,10H,10H,10H,10H,00H,10H
0214'  10 10 00 10
0218'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
021C'  00 00 00 00
0220'                          H22:
0220'  00 28 28 28                    DB      00H,28H,28H,28H,00H,00H,00H,00H
0224'  00 00 00 00
0228'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
022C'  00 00 00 00
0230'                          H23:
0230'  00 28 28 7C                    DB      00H,28H,28H,7CH,28H,7CH,28H,28H
0234'  28 7C 28 28
0238'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
023C'  00 00 00 00
0240'                          H24:
0240'  00 10 3C 50                    DB      00H,10H,3CH,50H,38H,14H,78H,10H
0244'  38 14 78 10
0248'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
024C'  00 00 00 00
0250'                          H25:
0250'  00 60 64 08                    DB      00H,60H,64H,08H,10H,20H,4CH,0CH
0254'  10 20 4C 0C
0258'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
025C'  00 00 00 00
0260'                          H26:
0260'  00 10 28 28                    DB      00H,10H,28H,28H,30H,54H,48H,34H
0264'  30 54 48 34
0268'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
026C'  00 00 00 00
0270'                          H27:
0270'  00 10 10 20                    DB      00H,10H,10H,20H,00H,00H,00H,00H
0274'  00 00 00 00
0278'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
027C'  00 00 00 00
0280'                          H28:
0280'  00 08 10 20                    DB      00H,08H,10H,20H,20H,20H,10H,08H
0284'  20 20 10 08
0288'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
028C'  00 00 00 00
0290'                          H29:
0290'  00 20 10 08                    DB      00H,20H,10H,08H,08H,08H,10H,20H
0294'  08 08 10 20
0298'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
029C'  00 00 00 00
02A0'                          H2A:
02A0'  00 00 10 54                    DB      00H,00H,10H,54H,38H,54H,10H,00H
02A4'  38 54 10 00
02A8'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
02AC'  00 00 00 00
```

```
02B0'                          H2B:
02B0'  00 00 10 10                    DB      00H,00H,10H,10H,7CH,10H,10H,00H
02B4'  7C 10 10 00
02B8'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
02BC'  00 00 00 00
02C0'                          H2C:
02C0'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,10H,10H
02C4'  00 00 10 10
02C8'  20 00 00 00                    DB      20H,00H,00H,00H,00H,00H,00H,00H
02CC'  00 00 00 00
02D0'                          H2D:
02D0'  00 00 00 00                    DB      00H,00H,00H,00H,7CH,00H,00H,00H
02D4'  7C 00 00 00
02D8'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
02DC'  00 00 00 00
02E0'                          H2E:
02E0'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,10H
02E4'  00 00 00 10
02E8'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
02EC'  00 00 00 00
02F0'                          H2F:
02F0'  00 00 04 08                    DB      00H,00H,04H,08H,10H,20H,40H,00H
02F4'  10 20 40 00
02F8'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
02FC'  00 00 00 00
0300'                          H30:
0300'  00 38 44 4C                    DB      00H,38H,44H,4CH,54H,64H,44H,38H
0304'  54 64 44 38
0308'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
030C'  00 00 00 00
0310'                          H31:
0310'  00 10 30 10                    DB      00H,10H,30H,10H,10H,10H,10H,38H
0314'  10 10 10 38
0318'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
031C'  00 00 00 00
0320'                          H32:
0320'  00 38 44 04                    DB      00H,38H,44H,04H,18H,20H,40H,7CH
0324'  18 20 40 7C
0328'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
032C'  00 00 00 00
0330'                          H33:
0330'  00 7C 04 08                    DB      00H,7CH,04H,08H,18H,04H,44H,38H
0334'  18 04 44 38
0338'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
033C'  00 00 00 00
0340'                          H34:
0340'  00 08 18 28                    DB      00H,08H,18H,28H,48H,7CH,08H,08H
0344'  48 7C 08 08
0348'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
034C'  00 00 00 00
0350'                          H35:
0350'  00 7C 40 78                    DB      00H,7CH,40H,78H,04H,04H,44H,38H
0354'  04 04 44 38
0358'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
035C'  00 00 00 00
0360'                          H36:
```

```
0360'  00 18 20 40              DB    00H,18H,20H,40H,78H,44H,44H,38H
0364'  78 44 44 38
0368'  00 00 00 00              DB    00H,00H,00H,00H,00H,00H,00H,00H
036C'  00 00 00 00
0370'                     H37:
0370'  00 7C 04 08              DB    00H,7CH,04H,08H,10H,20H,20H,20H
0374'  10 20 20 20
0378'  00 00 00 00              DB    00H,00H,00H,00H,00H,00H,00H,00H
037C'  00 00 00 00
0380'                     H38:
0380'  00 38 44 44              DB    00H,38H,44H,44H,38H,44H,44H,38H
0384'  38 44 44 38
0388'  00 00 00 00              DB    00H,00H,00H,00H,00H,00H,00H,00H
038C'  00 00 00 00
0390'                     H39:
0390'  00 38 44 44              DB    00H,38H,44H,44H,3CH,04H,08H,30H
0394'  3C 04 08 30
0398'  00 00 00 00              DB    00H,00H,00H,00H,00H,00H,00H,00H
039C'  00 00 00 00
03A0'                     H3A:
03A0'  00 00 00 00              DB    00H,00H,00H,00H,10H,00H,10H,00H
03A4'  10 00 10 00
03A8'  00 00 00 00              DB    00H,00H,00H,00H,00H,00H,00H,00H
03AC'  00 00 00 00
03B0'                     H3B:
03B0'  00 00 00 00              DB    00H,00H,00H,00H,10H,00H,10H,10H
03B4'  10 00 10 10
03B8'  20 00 00 00              DB    20H,00H,00H,00H,00H,00H,00H,00H
03BC'  00 00 00 00
03C0'                     H3C:
03C0'  00 08 10 20              DB    00H,08H,10H,20H,40H,20H,10H,08H
03C4'  40 20 10 08
03C8'  00 00 00 00              DB    00H,00H,00H,00H,00H,00H,00H,00H
03CC'  00 00 00 00
03D0'                     H3D:
03D0'  00 00 00 7C              DB    00H,00H,00H,7CH,00H,7CH,00H,00H
03D4'  00 7C 00 00
03D8'  00 00 00 00              DB    00H,00H,00H,00H,00H,00H,00H,00H
03DC'  00 00 00 00
03E0'                     H3E:
03E0'  00 20 10 08              DB    00H,20H,10H,08H,04H,08H,10H,20H
03E4'  04 08 10 20
03E8'  00 00 00 00              DB    00H,00H,00H,00H,00H,00H,00H,00H
03EC'  00 00 00 00
03F0'                     H3F:
03F0'  00 38 44 04              DB    00H,38H,44H,04H,08H,10H,00H,10H
03F4'  08 10 00 10
03F8'  00 00 00 00              DB    00H,00H,00H,00H,00H,00H,00H,00H
03FC'  00 00 00 00
0400'                     H40:
0400'  00 38 44 5C              DB    00H,38H,44H,5CH,54H,58H,44H,38H
0404'  54 58 44 38
0408'  00 00 00 00              DB    00H,00H,00H,00H,00H,00H,00H,00H
040C'  00 00 00 00
0410'                     H41:
0410'  00 10 28 44              DB    00H,10H,28H,44H,44H,7CH,44H,44H
```

```
0414'  44 7C 44 44
0418'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
041C'  00 00 00 00
0420'                          H42:
0420'  00 78 44 44                    DB      00H,78H,44H,44H,78H,44H,44H,78H
0424'  78 44 44 78
0428'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
042C'  00 00 00 00
0430'                          H43:
0430'  00 38 44 40                    DB      00H,38H,44H,40H,40H,40H,44H,38H
0434'  40 40 44 38
0438'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
043C'  00 00 00 00
0440'                          H44:
0440'  00 78 24 24                    DB      00H,78H,24H,24H,24H,24H,24H,78H
0444'  24 24 24 78
0448'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
044C'  00 00 00 00
0450'                          H45:
0450'  00 7C 40 40                    DB      00H,7CH,40H,40H,78H,40H,40H,7CH
0454'  78 40 40 7C
0458'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
045C'  00 00 00 00
0460'                          H46:
0460'  00 7C 40 40                    DB      00H,7CH,40H,40H,78H,40H,40H,40H
0464'  78 40 40 40
0468'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
046C'  00 00 00 00
0470'                          H47:
0470'  00 38 44 40                    DB      00H,38H,44H,40H,40H,4CH,44H,3CH
0474'  40 4C 44 3C
0478'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
047C'  00 00 00 00
0480'                          H48:
0480'  00 44 44 44                    DB      00H,44H,44H,44H,7CH,44H,44H,44H
0484'  7C 44 44 44
0488'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
048C'  00 00 00 00
0490'                          H49:
0490'  00 38 10 10                    DB      00H,38H,10H,10H,10H,10H,10H,38H
0494'  10 10 10 38
0498'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
049C'  00 00 00 00
04A0'                          H4A:
04A0'  00 04 04 04                    DB      00H,04H,04H,04H,04H,04H,44H,38H
04A4'  04 04 44 38
04A8'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
04AC'  00 00 00 00
04B0'                          H4B:
04B0'  00 44 48 50                    DB      00H,44H,48H,50H,60H,50H,48H,44H
04B4'  60 50 48 44
04B8'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
04BC'  00 00 00 00
04C0'                          H4C:
04C0'  00 40 40 40                    DB      00H,40H,40H,40H,40H,40H,40H,7CH
04C4'  40 40 40 7C
```

```
04C8'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
04CC'  00 00 00 00
04D0'                          H4D:
04D0'  00 44 6C 54                    DB      00H,44H,6CH,54H,44H,44H,44H,44H
04D4'  44 44 44 44
04D8'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
04DC'  00 00 00 00
04E0'                          H4E:
04E0'  00 44 44 64                    DB      00H,44H,44H,64H,54H,4CH,44H,44H
04E4'  54 4C 44 44
04E8'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
04EC'  00 00 00 00
04F0'                          H4F:
04F0'  00 38 44 44                    DB      00H,38H,44H,44H,44H,44H,44H,38H
04F4'  44 44 44 38
04F8'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
04FC'  00 00 00 00
0500'                          H50:
0500'  00 78 44 44                    DB      00H,78H,44H,44H,78H,40H,40H,40H
0504'  78 40 40 40
0508'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
050C'  00 00 00 00
0510'                          H51:
0510'  00 38 44 44                    DB      00H,38H,44H,44H,44H,54H,48H,34H
0514'  44 54 48 34
0518'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
051C'  00 00 00 00
0520'                          H52:
0520'  00 78 44 44                    DB      00H,78H,44H,44H,78H,50H,48H,44H
0524'  78 50 48 44
0528'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
052C'  00 00 00 00
0530'                          H53:
0530'  00 38 44 40                    DB      00H,38H,44H,40H,38H,04H,44H,38H
0534'  38 04 44 38
0538'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
053C'  00 00 00 00
0540'                          H54:
0540'  00 7C 10 10                    DB      00H,7CH,10H,10H,10H,10H,10H,10H
0544'  10 10 10 10
0548'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
054C'  00 00 00 00
0550'                          H55:
0550'  00 44 44 44                    DB      00H,44H,44H,44H,44H,44H,44H,38H
0554'  44 44 44 38
0558'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
055C'  00 00 00 00
0560'                          H56:
0560'  00 44 44 44                    DB      00H,44H,44H,44H,28H,28H,10H,10H
0564'  28 28 10 10
0568'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
056C'  00 00 00 00
0570'                          H57:
0570'  00 44 44 44                    DB      00H,44H,44H,44H,54H,54H,54H,28H
0574'  54 54 54 28
0578'  00 00 00 00                    DB      00H,00H,00H,00H,00H,00H,00H,00H
```

```
057C'  00 00 00 00
0580'                    H58:
0580'  00 44 44 28               DB    00H,44H,44H,28H,10H,28H,44H,44H
0584'  10 28 44 44
0588'  00 00 00 00               DB    00H,00H,00H,00H,00H,00H,00H,00H
058C'  00 00 00 00
0590'                    H59:
0590'  00 44 44 28               DB    00H,44H,44H,28H,10H,10H,10H,10H
0594'  10 10 10 10
0598'  00 00 00 00               DB    00H,00H,00H,00H,00H,00H,00H,00H
059C'  00 00 00 00
05A0'                    H5A:
05A0'  00 7C 04 08               DB    00H,7CH,04H,08H,10H,20H,40H,7CH
05A4'  10 20 40 7C
05A8'  00 00 00 00               DB    00H,00H,00H,00H,00H,00H,00H,00H
05AC'  00 00 00 00
05B0'                    H5B:
05B0'  00 38 20 20               DB    00H,38H,20H,20H,20H,20H,20H,38H
05B4'  20 20 20 38
05B8'  00 00 00 00               DB    00H,00H,00H,00H,00H,00H,00H,00H
05BC'  00 00 00 00
05C0'                    H5C:
05C0'  00 00 40 20               DB    00H,00H,40H,20H,10H,08H,04H,00H
05C4'  10 08 04 00
05C8'  00 00 00 00               DB    00H,00H,00H,00H,00H,00H,00H,00H
05CC'  00 00 00 00
05D0'                    H5D:
05D0'  00 38 08 08               DB    00H,38H,08H,08H,08H,08H,08H,38H
05D4'  08 08 08 38
05D8'  00 00 00 00               DB    00H,00H,00H,00H,00H,00H,00H,00H
05DC'  00 00 00 00
05E0'                    H5E:
05E0'  00 10 28 44               DB    00H,10H,28H,44H,00H,00H,00H,00H
05E4'  00 00 00 00
05E8'  00 00 00 00               DB    00H,00H,00H,00H,00H,00H,00H,00H
05EC'  00 00 00 00
05F0'                    H5F:
05F0'  00 00 00 00               DB    00H,00H,00H,00H,00H,00H,00H,7CH
05F4'  00 00 00 7C
05F8'  00 00 00 00               DB    00H,00H,00H,00H,00H,00H,00H,00H
05FC'  00 00 00 00
0600'                    H60:
0600'  00 0C 14 38               DB    00H,0CH,14H,38H,10H,10H,64H,58H
0604'  10 10 64 58
0608'  00 00 00 00               DB    00H,00H,00H,00H,00H,00H,00H,00H
060C'  00 00 00 00
0610'                    H61:
0610'  00 00 00 34               DB    00H,00H,00H,34H,4CH,44H,4CH,34H
0614'  4C 44 4C 34
0618'  00 00 00 00               DB    00H,00H,00H,00H,00H,00H,00H,00H
061C'  00 00 00 00
0620'                    H62:
0620'  00 40 40 58               DB    00H,40H,40H,58H,64H,44H,64H,58H
0624'  64 44 64 58
0628'  00 00 00 00               DB    00H,00H,00H,00H,00H,00H,00H,00H
062C'  00 00 00 00
```

```
0630'                           H63:
0630'  00 00 00 3C                      DB      00H,00H,00H,3CH,40H,40H,40H,3CH
0634'  40 40 40 3C
0638'  00 00 00 00                      DB      00H,00H,00H,00H,00H,00H,00H,00H
063C'  00 00 00 00
0640'                           H64:
0640'  00 04 04 34                      DB      00H,04H,04H,34H,4CH,44H,4CH,34H
0644'  4C 44 4C 34
0648'  00 00 00 00                      DB      00H,00H,00H,00H,00H,00H,00H,00H
064C'  00 00 00 00
0650'                           H65:
0650'  00 00 00 38                      DB      00H,00H,00H,38H,44H,7CH,40H,3CH
0654'  44 7C 40 3C
0658'  00 00 00 00                      DB      00H,00H,00H,00H,00H,00H,00H,00H
065C'  00 00 00 00
0660'                           H66:
0660'  00 0C 10 10                      DB      00H,0CH,10H,10H,38H,10H,10H,10H
0664'  38 10 10 10
0668'  00 00 00 00                      DB      00H,00H,00H,00H,00H,00H,00H,00H
066C'  00 00 00 00
0670'                           H67:
0670'  00 00 00 34                      DB      00H,00H,00H,34H,4CH,44H,3CH,04H
0674'  4C 44 3C 04
0678'  38 00 00 00                      DB      38H,00H,00H,00H,00H,00H,00H,00H
067C'  00 00 00 00
0680'                           H68:
0680'  00 40 40 78                      DB      00H,40H,40H,78H,44H,44H,44H,44H
0684'  44 44 44 44
0688'  00 00 00 00                      DB      00H,00H,00H,00H,00H,00H,00H,00H
068C'  00 00 00 00
0690'                           H69:
0690'  00 00 10 00                      DB      00H,00H,10H,00H,30H,10H,10H,38H
0694'  30 10 10 38
0698'  00 00 00 00                      DB      00H,00H,00H,00H,00H,00H,00H,00H
069C'  00 00 00 00
06A0'                           H6A:
06A0'  00 00 08 00                      DB      00H,00H,08H,00H,08H,08H,08H,08H
06A4'  08 08 08 08
06A8'  30 00 00 00                      DB      30H,00H,00H,00H,00H,00H,00H,00H
06AC'  00 00 00 00
06B0'                           H6B:
06B0'  00 40 40 48                      DB      00H,40H,40H,48H,50H,70H,48H,44H
06B4'  50 70 48 44
06B8'  00 00 00 00                      DB      00H,00H,00H,00H,00H,00H,00H,00H
06BC'  00 00 00 00
06C0'                           H6C:
06C0'  00 30 10 10                      DB      00H,30H,10H,10H,10H,10H,10H,38H
06C4'  10 10 10 38
06C8'  00 00 00 00                      DB      00H,00H,00H,00H,00H,00H,00H,00H
06CC'  00 00 00 00
06D0'                           H6D:
06D0'  00 00 00 68                      DB      00H,00H,00H,68H,54H,54H,54H,54H
06D4'  54 54 54 54
06D8'  00 00 00 00                      DB      00H,00H,00H,00H,00H,00H,00H,00H
06DC'  00 00 00 00
06E0'                           H6E:
```

```
06E0'  00 00 00 78                DB    00H,00H,00H,78H,44H,44H,44H,44H
06E4'  44 44 44 44
06E8'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
06EC'  00 00 00 00
06F0'                        H6F:
06F0'  00 00 00 38                DB    00H,00H,00H,38H,44H,44H,44H,38H
06F4'  44 44 44 38
06F8'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
06FC'  00 00 00 00
0700'                        H70:
0700'  00 00 00 58                DB    00H,00H,00H,58H,64H,44H,78H,40H
0704'  64 44 78 40
0708'  40 00 00 00                DB    40H,00H,00H,00H,00H,00H,00H,00H
070C'  00 00 00 00
0710'                        H71:
0710'  00 00 00 34                DB    00H,00H,00H,34H,4CH,44H,3CH,04H
0714'  4C 44 3C 04
0718'  04 00 00 00                DB    04H,00H,00H,00H,00H,00H,00H,00H
071C'  00 00 00 00
0720'                        H72:
0720'  00 00 00 5C                DB    00H,00H,00H,5CH,60H,40H,40H,40H
0724'  60 40 40 40
0728'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
072C'  00 00 00 00
0730'                        H73:
0730'  00 00 00 3C                DB    00H,00H,00H,3CH,40H,38H,04H,78H
0734'  40 38 04 78
0738'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
073C'  00 00 00 00
0740'                        H74:
0740'  00 10 10 38                DB    00H,10H,10H,38H,10H,10H,10H,08H
0744'  10 10 10 08
0748'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
074C'  00 00 00 00
0750'                        H75:
0750'  00 00 00 44                DB    00H,00H,00H,44H,44H,44H,44H,3CH
0754'  44 44 44 3C
0758'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
075C'  00 00 00 00
0760'                        H76:
0760'  00 00 00 44                DB    00H,00H,00H,44H,44H,28H,28H,10H
0764'  44 28 28 10
0768'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
076C'  00 00 00 00
0770'                        H77:
0770'  00 00 00 44                DB    00H,00H,00H,44H,44H,54H,54H,28H
0774'  44 54 54 28
0778'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
077C'  00 00 00 00
0780'                        H78:
0780'  00 00 00 44                DB    00H,00H,00H,44H,28H,10H,28H,44H
0784'  28 10 28 44
0788'  00 00 00 00                DB    00H,00H,00H,00H,00H,00H,00H,00H
078C'  00 00 00 00
0790'                        H79:
0790'  00 00 00 44                DB    00H,00H,00H,44H,44H,44H,3CH,04H
```

```
0794'  44 44 3C 04
0798'  38 00 00 00              DB      38H,00H,00H,00H,00H,00H,00H,00H
079C'  00 00 00 00
07A0'                   H7A:
07A0'  00 00 00 7C              DB      00H,00H,00H,7CH,08H,10H,20H,7CH
07A4'  08 10 20 7C
07A8'  00 00 00 00              DB      00H,00H,00H,00H,00H,00H,00H,00H
07AC'  00 00 00 00
07B0'                   H7B:
07B0'  00 18 20 20              DB      00H,18H,20H,20H,40H,20H,20H,18H
07B4'  40 20 20 18
07B8'  00 00 00 00              DB      00H,00H,00H,00H,00H,00H,00H,00H
07BC'  00 00 00 00
07C0'                   H7C:
07C0'  00 10 10 10              DB      00H,10H,10H,10H,00H,10H,10H,10H
07C4'  00 10 10 10
07C8'  00 00 00 00              DB      00H,00H,00H,00H,00H,00H,00H,00H
07CC'  00 00 00 00
07D0'                   H7D:
07D0'  00 30 08 08              DB      00H,30H,08H,08H,04H,08H,08H,30H
07D4'  04 08 08 30
07D8'  00 00 00 00              DB      00H,00H,00H,00H,00H,00H,00H,00H
07DC'  00 00 00 00
07E0'                   H7E:
07E0'  00 34 58 00              DB      00H,34H,58H,00H,00H,00H,00H,00H
07E4'  00 00 00 00
07E8'  00 00 00 00              DB      00H,00H,00H,00H,00H,00H,00H,00H
07EC'  00 00 00 00
07F0'                   H7F:
07F0'  FF FF FF FF              DB      0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
07F4'  FF FF FF FF
07F8'  FF FF FF FF              DB      0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
07FC'  FF FF FF FF
                                END
```

Macros:

Symbols:

0000'	H00	0010'	H01	0020'	H02
0030'	H03	0040'	H04	0050'	H05
0060'	H06	0070'	H07	0080'	H08
0090'	H09	00A0'	H0A	00B0'	H0B
00C0'	H0C	00D0'	H0D	00E0'	H0E
00F0'	H0F	0100'	H10	0110'	H11
0120'	H12	0130'	H13	0140'	H14
0150'	H15	0160'	H16	0170'	H17
0180'	H18	0190'	H19	01A0'	H1A
01B0'	H1B	01C0'	H1C	01D0'	H1D
01E0'	H1E	01F0'	H1F	0200'	H20
0210'	H21	0220'	H22	0230'	H23
0240'	H24	0250'	H25	0260'	H26
0270'	H27	0280'	H28	0290'	H29
02A0'	H2A	02B0'	H2B	02C0'	H2C
02D0'	H2D	02E0'	H2E	02F0'	H2F
0300'	H30	0310'	H31	0320'	H32
0330'	H33	0340'	H34	0350'	H35
0360'	H36	0370'	H37	0380'	H38
0390'	H39	03A0'	H3A	03B0'	H3B
03C0'	H3C	03D0'	H3D	03E0'	H3E
03F0'	H3F	0400'	H40	0410'	H41
0420'	H42	0430'	H43	0440'	H44
0450'	H45	0460'	H46	0470'	H47
0480'	H48	0490'	H49	04A0'	H4A
04B0'	H4B	04C0'	H4C	04D0'	H4D
04E0'	H4E	04F0'	H4F	0500'	H50
0510'	H51	0520'	H52	0530'	H53
0540'	H54	0550'	H55	0560'	H56
0570'	H57	0580'	H58	0590'	H59
05A0'	H5A	05B0'	H5B	05C0'	H5C
05D0'	H5D	05E0'	H5E	05F0'	H5F
0600'	H60	0610'	H61	0620'	H62
0630'	H63	0640'	H64	0650'	H65
0660'	H66	0670'	H67	0680'	H68
0690'	H69	06A0'	H6A	06B0'	H6B
06C0'	H6C	06D0'	H6D	06E0'	H6E
06F0'	H6F	0700'	H70	0710'	H71
0720'	H72	0730'	H73	0740'	H74
0750'	H75	0760'	H76	0770'	H77
0780'	H78	0790'	H79	07A0'	H7A
07B0'	H7B	07C0'	H7C	07D0'	H7D
07E0'	H7E	07F0'	H7F		

No Fatal error(s)

D. Bootstrap ROM Listing

```
                        .Z80
                        ;
                        ;       BOOTSTRAP PROM FOR PIED PIPER COMMUNICATOR I
                        ;
                        ;       COPYRIGHT  SEMI-TECH MICROELECTRONICS CORPORATION
                        ;
                        ;-------------------------------------------------------------
                        ;
                        ;       THE BOOT PROM PERFORMS THE FOLLOWING:-
                        ;
                        ;       1.  HARDWARE INITIALIZATION
                        ;       2.  START UP DIAGNOSTICS
                        ;       3.  SOFTWARE INITIALIZATION
                        ;       4.  READ IN BOOT SECTOR FROM DISK
                        ;       5.  PASS CONTROL TO BOOT PROGRAM
                        ;
                        ;-------------------------------------------------------------
                        ;
                        ;       THE FOLLOWING EQUATES MUST BE VERIFIED FOR PRODUCTION
                        ;
                        ;       CMDREG - COMMAND REGISTER
                        ;       STAREG - STATUS REGISTER
                        ;       VTCBAS - PVTC BASE ADDRESS
                        ;       VTIDAT - PVTC INPUT BUFFER
                        ;       VTODAT - PVTC OUTPUT BUFFER
                        ;       VTCMOD - PVTC MODE REGISTER
                        ;       VTMODE - PVTC MODE
                        ;       KEYBD  - KEYBOARD BASE ADDRESS
                        ;       CAPREG - CAPITAL LOCK STATUS REGISTER
                        ;       CAPLCK - CAPITAL LOCK BIT
                        ;       PROMST - PROM PROGRAM ORIGIN
                        ;       ROMSAD - UPPER HALF OF ROM START ADDRESS
                        ;       ITRVEC - INTERRUPT VECTOR HIGH BYTE
                        ;       PROMAD - ADDRESS OF LOWER HALF OF ROM TO MOVE TO
                        ;       M1BEG  - START ADDRESS OF UPPER MEMORY TEST
                        ;       M1SIZE - SIZE OF UPPER MEMORY TEST
                        ;       M2BEG  - START ADDRESS OF LOWER MEMORY TEST
                        ;       M2SIZE - SIZE OF LOWER MEMORY TEST
                        ;       MEMEND - END OF MEMORY
                        ;       IOREG1 - STATUS REGISTER
                        ;       FDBASE - FLOPPY DISK CONTROLLER BASE ADDRESS
                        ;       WHDBAS - HARD DISK CONTROLLER BASE ADDRESS
                        ;
                        ;       SPECIAL CHANGES MADE FOR PP TESTING CAN BE FOUND BY SEARCHING
                        ;       FOR THE PATTERN ";'{"
                        ;
                        ;       GENERAL EQUATES
                        ;
000D                    CR      EQU     0DH             ; CARRIAGE RETURN
000A                    LF      EQU     0AH             ; LINE FEED
0008                    BS      EQU     08H
0020                    BLANK   EQU     20H             ; BLANK
0002                    CNTLB   EQU     02H             ; RETRY BOOT
0029                    DISPON  EQU     29H             ; DISPLAY ALL MESSAGE - FUNC-S
0034                    MEMTST  EQU     34H             ; MEMORY TEST - FUNC-M
002A                    DSKTST  EQU     2AH             ; DISK TEST - FUNC-D
```

```
0000                    PVTCR0  EQU     00H             ; VIDEO RAM READ BANK 0 - FUNC-1
0001                    PVTCW0  EQU     01H             ; VIDEO RAM WRITE BANK 0 - FUNC-2
0009                    PVTCR1  EQU     09H             ; VIDEO RAM READ BANK 1 - FUNC-3
000A                    PVTCW1  EQU     0AH             ; VIDEO RAM WRITE BANK 1 - FUNC-4
0002                    RAMRD   EQU     02H             ; MEMORY READ - FUNC-5
0003                    RAMWR   EQU     03H             ; MEMORY WRITE - FUNC-6
00A5                    TSTPAT  EQU     0A5H            ; TEST PATTERN FOR MEMORY TEST
                        ;
                        ;       GENERAL REGISTERS
                        ;
0018                    CMDREG  EQU     18H             ; COMMAND REGISTER (WRITE ONLY)
                                                        ; BIT 7 = 1 - DISABLE KEYBOARD SCAN
                                                        ; BIT 6 = 1 - ENABLE HARD DISK INTERRUPT
                                                        ; BIT 5 = 1 - ENABLE TIMER INTERRUPT
                                                        ; BIT 4 = 1 - ENABLE KEYBOARD INTERRUPT
                                                        ; BIT 3 = PRINTER DATA STROBE
                                                        ; BIT 2 = FLOPPY DISK SIDE SELECT
                                                        ; BIT 1 = SELECT FLOPPY DISK DRIVE B
                                                        ; BIT 0 = SELECT FLOPPY DISK DRIVE A
0005                    TIMER   EQU     5       ; TIMER ENABLE BIT
0018                    STAREG  EQU     18H             ; STATUS REGISTER (READ ONLY)
                                                        ; BIT 7 = KEYBOARD SCANNER STATUS
                                                        ; BIT 6 = FLOPPY DISK INDEX
                                                        ; BIT 5 = 1 - PRINTER BUSY
                                                        ;       = 0 - PRINTER READY
                                                        ; BIT 4 = 0 CAPITAL LOCK
                                                        ; BIT 3-1 = INTERRUPT
                                                        ;       = 0 - DISK DATA REQUEST
                                                        ;       = 1 - DISK INTERRUPT REQUEST
                                                        ;       = 2 - PVTC INTERRUPT REQUEST
                                                        ;       = 3 - KEYBOARD INTERRUPT REQUEST
                                                        ;       = 4 - TIMER INTERRUPT REQUEST
                                                        ;       = 5 - EXPANSION J8
                                                        ;       = 6 - EXPANSION J7
                                                        ;       = 7 - J1
                                                        ; BIT 0 = 1 - INTERRUPT PENDING
                        ;
                        ;       PVTC REGISTERS
                        ;
0000                    VTCBAS  EQU     00H             ; VTC BASE ADDRESS
0000                    VTCINI  EQU     VTCBAS+0        ; INITIALIZATION REGISTER
0001                    VTCSTA  EQU     VTCBAS+1        ; STATUS REGISTER
0001                    VTCCMD  EQU     VTCBAS+1        ; COMMAND REGISTER
0002                    VTCSC1  EQU     VTCBAS+2        ; SCREEN START REGISTER 1
0003                    VTCSC2  EQU     VTCBAS+3        ; SCREEN START REGISTER 2
0004                    VTCCU1  EQU     VTCBAS+4        ; CURSOR REGISTER 1
0005                    VTCCU2  EQU     VTCBAS+5        ; CURSOR REGISTER 2
0006                    VTCPT1  EQU     VTCBAS+6        ; POINTER REGISTER 1
0007                    VTCPT2  EQU     VTCBAS+7        ; POINTER REGISTER 2
000D                    VTIDAT  EQU     VTCBAS+0DH      ; INPUT DATA REGISTER
001C                    VTODAT  EQU     VTCBAS+1CH      ; OUTPUT DATA REGISTER
001D                    VTCMOD  EQU     VTCBAS+1DH      ; MODE REGISTER
                        ;
                        ;       PVTC COMMANDS
                        ;
0000                    VTCRST  EQU     00000000B       ; VTC RESET
```

```
00AB                    WRINC   EQU     10101011B       ; WRITE AT CURSOR AND INCREMENT
00AA                    WRCUR   EQU     10101010B       ; WRITE AT CURSOR
00BB                    WRCP    EQU     10111011B       ; WRITE FROM CURSOR TO PRINTER
00AC                    RDCUR   EQU     10101100B       ; READ AT CURSOR
00AD                    RDINC   EQU     10101101B       ; READ AT CURSOR AND INCREMENT
003D                    ONDICU  EQU     00111101B       ; DISPLAY ON
0028                    OFFDIS  EQU     00101000B       ; DISPLAY OFF
0031                    CURON   EQU     00110001B       ; CURSOR ON
0030                    CUROFF  EQU     00110000B       ; CURSOR OFF
0020                    VTCRDY  EQU     00100000B       ; VTC READY MASK
0010                    VBLANK  EQU     00010000B       ; VTC VBLANK MASK
0008                    LINZER  EQU     00001000B       ; VTC LINE ZERO MASK
0004                    SPLITS  EQU     00000100B       ; VTC SPLIT SCREEN MASK
0080                    VTCBEG  EQU     0080H           ; VTC START ADDRESS
07FF                    VTCEND  EQU     07FFH           ; VTC END ADDRESS
0050                    RSTVBL  EQU     01010000B       ; RESET VBLANK
0048                    RSTLN0  EQU     01001000B       ; RESET LINE ZERO
0004                    VBIT    EQU     4       ; VBLANK BIT
0003                    LINE0   EQU     3       ; LINE ZERO BIT
0018                    INTMSK  EQU     18H             ; INTERRUPT STATUS MASK
0040                    RSTMSK  EQU     40H             ; RESET COMMAND
0001                    COL40   EQU     00000001B       ; 40 COLUMN
0002                    REVVID  EQU     00000010B       ; REVERSE VIDEO
0004                    INTENS  EQU     00000100B       ; HIGH LIGHT
0008                    CHAR2K  EQU     00001000B       ; REDUCE CHARACTER SET
0010                    INTERL  EQU     00010000B       ; INTERLACE
0020                    BONW    EQU     00100000B       ; BLACK ON WHITE
0040                    DOT7    EQU     01000000B       ; 7 DOTS HORIZONTAL
0080                    BEEP    EQU     10000000B       ; ACTIVATE BEEP
0054                    VTMODE  EQU     INTERL+INTENS+DOT7
                        ;
                        ;       PVTC INITIALIZATION REGISTERS
                        ;
0044                    IR0     EQU     01000100B
0032                    IR1     EQU     00110010B
000E                    IR2     EQU     00001110B
0066                    IR3     EQU     01100110B
0017                    IR4     EQU     00010111B
004F                    IR5     EQU     01001111B
0008                    IR6     EQU     00001000B
0009                    IR7     EQU     00001001B
0080                    IR8     EQU     10000000B
0010                    IR9     EQU     00010000B
0080                    IR10    EQU     10000000B
                        ;
                        ;       PVTC CONSTANTS
                        ;
0050                    MAXCOL  EQU     80
0017                    MAXLIN  EQU     23
0000                    SCRLOF  EQU     0       ; SCROLL OFF
0001                    SCRLON  EQU     1       ; SCROLL ON
                        ;
                        ;       KEYBOARD
                        ;
0008                    KEYBD   EQU     08H             ; BASE ADDRESS
0008                    KEYBDA  EQU     KEYBD+0         ; DATA REGISTER
```

```
0009                    KEYBST  EQU     KEYBD+1         ; STATUS REGISTER
0009                    KEYBCM  EQU     KEYBD+1         ; COMMAND REGISTER
000F                    KEYBMK  EQU     00001111B       ; KEYBOARD STATUS MASK
0000                    KBMOD1  EQU     00000000B       ; ENCODED SCAN, 2-KEY LOCKOUT
002A                    KBMOD2  EQU     00101010B       ; PROGRAM CLOCK FOR 20 MSEC INTERVAL
003F                    KEYMSK  EQU     00111111B       ; CONTROL AND SHIFT BITS MASK
0006                    SHIFT   EQU     6       ; SHIFT KEY DEPRESSED BIT
0007                    CNTL    EQU     7       ; CONTROL KEY DEPRESSED BIT
0007                    KBDENA  EQU     7       ; KEYBOARD SCAN ENABLE/DISABLE BIT
0018                    CAPREG  EQU     STAREG          ; CAPITAL LOCK REGISTER
0004                    CAPLCK  EQU     4       ; CAPITAL LOCK BIT
0005                    CAPBIT  EQU     5       ; UPPER CASE BIT
0005                    PT05    EQU     5       ; 50 MSEC DELAY FOR AUTO REPEAT
0050                    ONESEC  EQU     80      ; 800 MSEC DELAY FOR REPEAT
                        ;
                        ;       PROGRAM ADDRESSES
                        ;
0000                    PROMST  EQU     00H
00FB                    ITRVEC  EQU     0FBH            ; INTERRUPT VECTOR HIGH BYTE
F4A0                    ROMSAD  EQU     0F4A0H          ; ADDRESS OF UPPER AREA OF PROM
F000                    PROMAD  EQU     0F000H          ; ADDRESS TO MOVE PROM TO
1000                    ROMSIZ  EQU     1000H           ; SIZE OF PROM
8000                    M1BEG   EQU     8000H           ; START ADDRESS FOR UPPER MEMORY TEST
FFFF                    M3BEG   EQU     0FFFFH          ; START ADDRESS FOR MEMORY LOOP TEST
8000                    M1SIZE  EQU     8000H           ; UPPER MEMORY TEST SIZE
0000                    M2BEG   EQU     PROMST          ; START ADDRESS FOR LOWER MEMORY TEST
EFFF                    M4BEG   EQU     0EFFFH          ; START ADDRESS FOR MEMORY LOOP TEST
F000                    M2SIZE  EQU     0F000H          ; LOWER MEMORY TEST SIZE
FFFF                    MEMEND  EQU     0FFFFH          ; END OF MEMORY
                        ;
                        ;       FLOPPY DISK CONSTATNS
                        ;
0018                    SEL     EQU     CMDREG          ; DISK SELECT PORT
0010                    FDBASE  EQU     10H             ; FLOPPY DISK CONTROLLER BASE ADDRESS
0010                    FDCMD   EQU     FDBASE+0        ; FLOPPY DISK CONTROLLER COMMAND REGISTER
0010                    FDSTAT  EQU     FDBASE+0        ; FLOPPY DISK CONTROLLER STATUS REGISTER
0011                    FDTRK   EQU     FDBASE+1        ; FLOPPY DISK CONTROLLER TRACK REGISTER
0012                    FDSEC   EQU     FDBASE+2        ; FLOPPY DISK CONTROLLER SECTOR REGISTER
0013                    FDDATA  EQU     FDBASE+3        ; FLOPPY DISK CONTROLLER DTA REGISTER
                        ;
0000                    RSCMD   EQU     00000000B       ; FLOPPY DISK RESTORE COMMAND
0008                    HDLOAD  EQU     00001000B       ; FLOPPY DISK HEAD LOAD BIT
0082                    RDCMD   EQU     10000010B       ; FLOPPY DISK READ SECTOR COMMAND
00C0                    RACMD   EQU     11000000B       ; FLOPPY DISK READ ADDRESS COMMAND
0004                    VERIFY  EQU     00000100B       ; FLOPPY DISK COMMAND VERIFY BIT
0010                    SKCMD   EQU     00010000B       ; floppy disk seek command
0050                    SICMD   EQU     01010000B       ; floppy disk step in command
00A2                    WRCMD   EQU     10100010B       ; floppy disk write sector command
00D0                    FITRPT  EQU     11010000B       ; FLOPPY DISK FORCE INTERRUPT COMMAND
0003                    SLOSTP  EQU     00000011B       ; SLOW STEPPING RATE
5511                    SPDLB   EQU     21777           ; FLOPPY DISK DRIVE SPEED LOWER BOUND
588A                    SPDUB   EQU     22666           ; FLOPPY DISK DRIVE SPEED UPPER BOUND
                        ;
8000                    VFYBUF  EQU     8000H           ; DISK TEST BUFFER
0080                    BUF     EQU     80H             ; DEFAULT BUFFER ADDRESS
0001                    DSKA    EQU     1       ; DISK DRIVE INDICATOR
```

```
0007                    SELMSK  EQU     00000111B       ; DISK SELECT MASK
                                                ; BIT 0-1 DRIVE NUMBER
                                                ; BIT 2 SIDE
00F8                    IOMSK   EQU     11111000B       ; I/O REGISTER MASK FOR DISK SELECT
0003                    DSK     EQU     00000011B       ; DISK DRIVE MASK
                                                ; BIT 0 -- DRIVE A
                                                ; BIT 1 -- DRIVE B
0010                    SKERR   EQU     00010000B       ; seek error status mask
0010                    RNF     EQU     00010000B       ; record not found status mask
0020                    HLD     EQU     00100000B
0098                    RSERR   EQU     10011000B       ; RESTORE ERROR MASK
009E                    RDERR   EQU     10011110B       ; FLOPPY DISK READ ERROR MASK
0080                    TIMEXP  EQU     10000000B       ; TIMED OUT ERROR
0007                    EXPTIM  EQU     7       ; TIMED OUT BIT
0002                    SIDE1   EQU     2       ; FLOPPY DISK SIDE SELECT BIT
0005                    DSIDE   EQU     5       ; FLOPPY DISK SELECT BYTE SIDE INFO BIT
                        ;
0000                    DSSTEP  EQU     0       ; STEPPING RATE FOR DS FLOPPY
000A                    RETRY   EQU     10      ; MAXIMUM NUMBER OF RETRIES
0005                    WRCMND  EQU     5       ; FLOPPY COMMAND WRITE BIT
A2ED                    INICOD  EQU     0A2EDH          ; INI INSTRUCTION
A3ED                    OUTICD  EQU     0A3EDH          ; OUTI INSTRUCTION
00C9                    RETCOD  EQU     0C9H            ; RET INSTRUCTION
                        ;
                        ;       HARD DISK CONSTANTS
                        ;
0030                    WHDBAS  EQU     30H             ; CONTROLLER BASE ADDRESS
0037                    HDCMD   EQU     WHDBAS+7        ; COMMAND REGISTER
0037                    HDSTAT  EQU     WHDBAS+7        ; STATUS REGISTER
0036                    HDSDH   EQU     WHDBAS+6        ; SECTOR SIZE, DRIVE, HEAD SELECT
0035                    HDCYLH  EQU     WHDBAS+5        ; CYLINDER HIGH BYTE
0034                    HDCYLL  EQU     WHDBAS+4        ; CYLINDER LOW BYTE
0033                    HDSEC   EQU     WHDBAS+3        ; SECTOR NUMBER
0032                    HDCNT   EQU     WHDBAS+2        ; SECTOR COUNT
0031                    HDWPC   EQU     WHDBAS+1        ; WRITE PRE-COMPENSATION
0031                    HDERR   EQU     WHDBAS+1        ; ERROR REGISTER
0030                    HDDATA  EQU     WHDBAS+0        ; DATA REGISTER
                        ;
0020                    HDREAD  EQU     00100000B       ; READ SECTOR COMMAND
0000                    HDERRS  EQU     0       ; ERROR BIT
                        ;
                                ORG     PROMST
                        ;
                        ;       HARDWARE INITIALIZATION
                        ;
0000'                   ROMBEG:
0000'  F3                       DI              ; DISABLE INTERRUPT
0001'  18 25                    JR      ITABEN          ; HOP AROUND INITIALIZATION TABLE
                        ;
                        ;       INITIALIZATION TABLE
                        ;
0003'                   INITAB:
0003'  25                       DB      ITABEN - INITAB
0004'  18 00                    DB      CMDREG,0
0006'  10 D0                    DB      FDCMD,FITRPT
0008'  1D 54                    DB      VTCMOD,VTMODE
```

```
000A'  01 00                    DB      VTCCMD,VTCRST
000C'  01 00                    DB      VTCCMD,VTCRST
000E'  00 44                    DB      VTCINI,IR0
0010'  00 32                    DB      VTCINI,IR1
0012'  00 0E                    DB      VTCINI,IR2
0014'  00 66                    DB      VTCINI,IR3
0016'  00 17                    DB      VTCINI,IR4
0018'  00 4F                    DB      VTCINI,IR5
001A'  00 08                    DB      VTCINI,IR6
001C'  00 09                    DB      VTCINI,IR7
001E'  00 80                    DB      VTCINI,IR8
0020'  00 10                    DB      VTCINI,IR9
0022'  00 80                    DB      VTCINI,IR10
0024'  09 00                    DB      KEYBCM,KBMOD1
0026'  09 2A                    DB      KEYBCM,KBMOD2
0028'                   ITABEN:
0028'  21 0003'                 LD      HL,INITAB       ; POINT TO TABLE
002B'  46                       LD      B,(HL)          ; OBTAIN LOOP COUNTER
002C'  CB 38                    SRL     B       ; DIVIDE BY TWO
002E'                   INILP1:
002E'  23                       INC     HL
002F'  4E                       LD      C,(HL)          ; GET PORT ADDRESS
0030'  23                       INC     HL
0031'  7E                       LD      A,(HL)          ; GET INITIALIZATION BYTE
0032'  ED 79                    OUT     (C),A
0034'  10 F8                    DJNZ    INILP1
                        ;
                        ;       HARDWARE INITIALIZATION DONE
                        ;
                        ;-----------------------------------------------------------
                        ;
                        ;       START UP DIAGNOSTICS
                        ;
                        ;       1.  CPU TEST
                        ;
0036'  AF                       XOR     A       ; GENERATE NC, Z, PE, P
0037'  ED 47                    LD      I,A             ; INITIALIZE ERROR INDICATOR
0039'  DA 0071'                 JP      C,CPUERR        ; TEST IF FLAGS SET PROPERLY
003C'  C2 0071'                 JP      NZ,CPUERR
003F'  E2 0071'                 JP      PO,CPUERR
0042'  FA 0071'                 JP      M,CPUERR
0045'  D6 02                    SUB     2       ; GENERATE C, NZ, PO, M
0047'  D2 0071'                 JP      NC,CPUERR       ; TEST IF FLAGS SET PROPERLY
004A'  CA 0071'                 JP      Z,CPUERR
004D'  EA 0071'                 JP      PE,CPUERR
0050'  F2 0071'                 JP      P,CPUERR
0053'  3E FF                    LD      A,0FFH          ; LOAD REGISTERS WITH PATTERN
0055'                   NXTTST:
0055'  47                       LD      B,A
0056'  48                       LD      C,B
0057'  51                       LD      D,C
0058'  5A                       LD      E,D
0059'  63                       LD      H,E
005A'  6C                       LD      L,H
005B'  08                       EX      AF,AF'
005C'  7D                       LD      A,L
```

```
005D'  D9                      EXX
005E'  47                      LD      B,A
005F'  48                      LD      C,B
0060'  51                      LD      D,C
0061'  5A                      LD      E,D
0062'  63                      LD      H,E
0063'  6C                      LD      L,H
0064'  08                      EX      AF,AF'
0065'  30 07                   JR      NC,CPUTST
0067'  FE FF                   CP      0FFH
0069'  20 06                   JR      NZ,CPUERR
006B'  AF                      XOR     A       ; SET FOR SECOND PATTERN
006C'  18 E7                   JR      NXTTST
006E'                  CPUTST:
006E'  B7                      OR      A
006F'  28 01                   JR      Z,CPUEND
0071'                  CPUERR:
0071'  76                      HALT            ; CPU ERROR
0072'                  CPUEND:
                       ;
                       ;       2.  ROM CHECKSUM
                       ;
0072'  21 04A0'                LD      HL,LAST         ; COMPUTE SIZE OF LOWER HALF OF CODE
0075'  01 0000                 LD      BC,PROMST
0078'  ED 42                   SBC     HL,BC
007A'  EB                      EX      DE,HL           ; SAVE FOR LATER
007B'  21 FF93'                LD      HL,CHKSUM       ; POINT TO CHECKSUM ADDRESS
007E'  01 F4A0                 LD      BC,ROMSAD
0081'  AF                      XOR     A
0082'  ED 42                   SBC     HL,BC           ; COMPUTE UPPER HALF ROM SIZE
0084'  19                      ADD     HL,DE
0085'  44                      LD      B,H
0086'  4D                      LD      C,L
0087'  21 0000'                LD      HL,ROMBEG       ; POINT TO START OF CODE
008A'                  TSTLP1:
008A'  AE                      XOR     (HL)            ; COMPUTE CHECKSUM
008B'  5F                      LD      E,A             ; SAVE FOR LATER
008C'  23                      INC     HL      ; POINT TO NEXT BYTE
008D'  0B                      DEC     BC      ; CHECK IF NO MORE
008E'  78                      LD      A,B
008F'  B1                      OR      C
0090'  7B                      LD      A,E             ; RETRIEVE CHECKSUM
0091'  20 F7                   JR      NZ,TSTLP1
0093'  BE                      CP      (HL)            ; COMPARE WITH PRE-DETERMINED VALUE
0094'  20 DB                   JR      NZ,CPUERR       ; HALT IF PROM ERROR
                       ;
                       ;       3.  PVTC
                       ;
0096'                  PVTC:
0096'  3E 58                   LD      A,INTMSK+RSTMSK
0098'  D3 01                   OUT     (VTCCMD),A      ; CHECK PVTC OPERATION
009A'  01 0000                 LD      BC,0            ; DELAY LOOP
009D'                  PVTC1:
009D'  10 FE                   DJNZ    PVTC1
009F'  0D                      DEC     C
00A0'  20 FB                   JR      NZ,PVTC1
```

```
00A2'  DB 01                    IN      A,(VTCSTA)      ; GET STATUS
00A4'  E6 38                    AND     VTCRDY+VBLANK+LINZER
00A6'  EE 38                    XOR     VTCRDY+VBLANK+LINZER
00A8'  20 24                    JR      NZ,PVTCER       ; ERROR IF NOT ALL SET
00AA'  AF                       XOR     A       ; SET STARTING ADDRESS
00AB'  D3 04                    OUT     (VTCCU1),A
00AD'  D3 05                    OUT     (VTCCU2),A
00AF'  21 07FF                  LD      HL,VTCEND       ; SET ENDING ADDRESS
00B2'  7D                       LD      A,L             ; SET ENDING ADDRESS
00B3'  D3 06                    OUT     (VTCPT1),A
00B5'  7C                       LD      A,H
00B6'  D3 07                    OUT     (VTCPT2),A
00B8'  3E A5                    LD      A,TSTPAT        ; SET UP TEST PATTERNS
00BA'  5F                       LD      E,A
00BB'  2F                       CPL
00BC'  57                       LD      D,A
00BD'  D3 1C                    OUT     (VTODAT),A      ; LOAD UP TEST PATTERN
00BF'  3E BB                    LD      A,WRCP
00C1'  D3 01                    OUT     (VTCCMD),A
00C3'                   PVTC2:
00C3'  10 FE                    DJNZ    PVTC2           ; SOFTWARE TIMER LOOP
00C5'  0D                       DEC     C
00C6'  20 FB                    JR      NZ,PVTC2
00C8'  DB 01                    IN      A,(VTCSTA)      ; GET STATUS
00CA'  E6 20                    AND     VTCRDY          ; CHECK IF READY
00CC'  20 09                    JR      NZ,WAITL0       ; OK IF SET
00CE'                   PVTCER:
00CE'  ED 57                    LD      A,I             ; SET ERROR
00D0'  CB C7                    SET     0,A
00D2'  ED 47                    LD      I,A
00D4'  C3 09B7'                 JP      INDBAD-ROMSAD+LAST-PROMST
                        ;
                        ;       3.1  DISPLAY RAM
                        ;
00D7'                   WAITL0:
00D7'  AF                       XOR     A       ; SET STARTING ADDRESS
00D8'  D3 04                    OUT     (VTCCU1),A
00DA'  D3 05                    OUT     (VTCCU2),A
00DC'  01 07FF                  LD      BC,VTCEND       ; SET LOOP COUNTER
00DF'  03                       INC     BC
00E0'                   VTSTLP:
00E0'  3E AC                    LD      A,RDCUR         ; READ BACK FOR COMPARISON
00E2'  D3 01                    OUT     (VTCCMD),A
00E4'  21 00E9'                 LD      HL,WAITL1
00E7'  18 42                    JR      VTCWAIT
00E9'                   WAITL1:
00E9'  DB 0D                    IN      A,(VTIDAT)
00EB'  BA                       CP      D
00EC'  20 22                    JR      NZ,VTCERR       ; BRANCH IF ERROR
00EE'  7B                       LD      A,E             ; GET NEXT PATTERN
00EF'  D3 1C                    OUT     (VTODAT),A
00F1'  3E AA                    LD      A,WRCUR         ; WRITE OUT
00F3'  D3 01                    OUT     (VTCCMD),A
00F5'  21 00FA'                 LD      HL,WAITL2
00F8'  18 31                    JR      VTCWAIT
00FA'                   WAITL2:
```

```
00FA'  3E AD                    LD      A,RDINC         ; READ BACK AND COMPARE
00FC'  D3 01                    OUT     (VTCCMD),A
00FE'  21 0103'                 LD      HL,WAITL3
0101'  18 28                    JR      VTCWAIT
0103'                   WAITL3:
0103'  DB 0D                    IN      A,(VTIDAT)
0105'  BB                       CP      E
0106'  20 08                    JR      NZ,VTCERR
0108'  0B                       DEC     BC      ; CHECK IF MORE TO CHECK
0109'  78                       LD      A,B
010A'  B1                       OR      C
010B'  20 D3                    JR      NZ,VTSTLP
010D'  B7                       OR      A       ; CHECK IF ERROR
010E'  28 22                    JR      Z,VTCOK
0110'                   VTCERR:
0110'  21 0080                  LD      HL,VTCBEG
0113'  7D                       LD      A,L             ; SET UP CURSOR
0114'  D3 04                    OUT     (VTCCU1),A
0116'  7C                       LD      A,H
0117'  D3 05                    OUT     (VTCCU2),A
0119'  3E 3D                    LD      A,ONDICU        ; TURN ON DISPLAY
011B'  D3 01                    OUT     (VTCCMD),A
011D'  21 0122'                 LD      HL,VTCERW
0120'  18 09                    JR      VTCWAIT
0122'                   VTCERW:
0122'  ED 57                    LD      A,I             ; GET ERROR INDICATOR
0124'  CB CF                    SET     1,A             ; SET PVTC ERROR BIT
0126'  ED 47                    LD      I,A             ; SAVE INDICATOR
0128'  C3 09B7'                 JP      INDBAD-ROMSAD+LAST-PROMST
012B'                   VTCWAIT:
012B'  DB 01                    IN      A,(VTCSTA)      ; GET STATUS REGISTER
012D'  E6 20                    AND     VTCRDY          ; CHECK IF READY
012F'  28 FA                    JR      Z,VTCWAIT       ; REPEAT IF NOT
0131'  E9                       JP      (HL)            ; SIMULATED RETURN
                        ;
                        ;       SET UP PVTC FOR DISPLAY
                        ;
0132'                   VTCOK:
0132'  21 0080                  LD      HL,VTCBEG       ; GET PVTC START ADDRESS
0135'  7D                       LD      A,L
0136'  D3 04                    OUT     (VTCCU1),A
0138'  D3 02                    OUT     (VTCSC1),A
013A'  7C                       LD      A,H
013B'  D3 05                    OUT     (VTCCU2),A
013D'  D3 03                    OUT     (VTCSC2),A
013F'  11 07FF                  LD      DE,VTCEND       ; GET PVTC END ADDRESS
0142'  7B                       LD      A,E
0143'  D3 06                    OUT     (VTCPT1),A
0145'  7A                       LD      A,D
0146'  D3 07                    OUT     (VTCPT2),A
0148'  3E 20                    LD      A,BLANK         ; BLANK OUT DISPLAY
014A'  D3 1C                    OUT     (VTODAT),A
014C'  3E BB                    LD      A,WRCP
014E'  D3 01                    OUT     (VTCCMD),A
0150'  21 0155'                 LD      HL,WAITL4
0153'  18 D6                    JR      VTCWAIT
```

```
0155'                       WAITL4:
0155'  7D                           LD      A,L             ; SET UP CURSOR
0156'  D3 04                        OUT     (VTCCU1),A
0158'  7C                           LD      A,H
0159'  D3 05                        OUT     (VTCCU2),A
015B'  3E 3D                        LD      A,ONDICU        ; TURN ON DISPLAY
015D'  D3 01                        OUT     (VTCCMD),A
015F'  21 0164'                     LD      HL,WAITL5
0162'  18 C7                        JR      VTCWAIT
0164'                       WAITL5:
0164'  DD 21 0308'                  LD      IX,SIGNON       ; POINT TO SIGNON MESSAGE
0168'  FD 21 016F'                  LD      IY,WAITL6
016C'  C3 02E2'                     JP      PMSG
016F'                       WAITL6:
                            ;
                            ;       CHECK IF DISPLAY REQUIRED
                            ;       OR ANY TEST SHOULD BE ACTIVATED
                            ;
016F'  01 0000                      LD      BC,0            ; SET UP SOFTWARE TIMER
0172'                       WAITL8:
0172'  DB 09                        IN      A,(KEYBST)      ; GET KEYBOARD STATUS
0174'  E6 0F                        AND     KEYBMK          ; CHECK IF DATA AVAILABLE
0176'  20 08                        JR      NZ,KEYRDY
0178'  0B                           DEC     BC      ; CHECK IF TIMED OUT
0179'  78                           LD      A,B
017A'  B1                           OR      C
017B'  20 F5                        JR      NZ,WAITL8
017D'  C3 0384'                     JP      NODISP
0180'                       KEYRDY:
0180'  DB 08                        IN      A,(KEYBDA)      ; GET KEY ENTERED
0182'  FE 29                        CP      DISPON          ; CHECK IF DISPLAY REQUIRE
0184'  CA 037E'                     JP      Z,SHOW          ;
0187'  FE 34                        CP      MEMTST          ; CHECK IF ACTIVATE MEMORY TEST
0189'  28 0B                        JR      Z,TESTM1
018B'  FE 2A                        CP      DSKTST          ; CHECK IF ACTIVATE DISK TEST
018D'  C2 0384'                     JP      NZ,NODISP
0190'  ED 57                        LD      A,I
0192'  CB DF                        SET     3,A             ; SET INDICATOR FOR LATER
0194'  ED 47                        LD      I,A
                            ;
                            ;       TEST UPPER MEMORY FIRST
                            ;
0196'                       TESTM1:
0196'  DD 21 0319'                  LD      IX,TMMSG        ; DISPLAY MEMORY TEST MESSAGE
019A'  FD 21 01A1'                  LD      IY,TSTMEM
019E'  C3 02E2'                     JP      PMSG
01A1'                       TSTMEM:
01A1'  08                           EX      AF,AF'          ; SET FOR FORWARD DIRECTION
01A2'  AF                           XOR     A
01A3'  08                           EX      AF,AF'
01A4'  1E 01                        LD      E,01H           ; INITIALIZE TEST PATTERN
01A6'                       NXPAT3:
01A6'  21 8000                      LD      HL,M1BEG        ; GET STARTING ADDRESS
01A9'  01 8000                      LD      BC,M1SIZE       ; SET UP LOOP COUNTER
01AC'                       LDLP3:
01AC'  73                           LD      (HL),E          ; LOAD PATTERN IN MEMORY
```

```
01AD'  23                        INC     HL      ; INCREMENT MEMORY INDEX
01AE'  0B                        DEC     BC      ; REPEAT UNTIL ALL LOADED
01AF'  78                        LD      A,B
01B0'  B1                        OR      C
01B1'  20 F9                     JR      NZ,LDLP3
01B3'  3E 01                     LD      A,01H           ; SET TO FLASH DRIVE A LED
01B5'  FD 21 01BC'               LD      IY,CHECK3       ; SET RETURN ADDRESS
01B9'  C3 029D'                  JP      FLHLP           ; FLASH LED
01BC'                    CHECK3:
01BC'  21 FFFF                   LD      HL,M3BEG        ; COMPUTE END ADDRESS
01BF'  01 8000                   LD      BC,M1SIZE       ; SET UP LOOP COUNTER
01C2'                    CHKLP3:
01C2'  7B                        LD      A,E             ; GET TEST PATTERN
01C3'  56                        LD      D,(HL)          ; GET MEMORY CONTENT
01C4'  BA                        CP      D       ; COMPARE WITH PATTERN
01C5'  20 65                     JR      NZ,M3ERR
01C7'  2F                        CPL             ; COMPLEMENT PATTERN
01C8'  77                        LD      (HL),A          ; LOAD IN MEMORY
01C9'  56                        LD      D,(HL)
01CA'  BA                        CP      D       ; AND CHECK IMMEDIATELY
01CB'  20 5F                     JR      NZ,M3ERR
01CD'                    NXTAD3:
01CD'  2B                        DEC     HL      ; DECREMENT MEMORY INDEX
01CE'  0B                        DEC     BC      ; REPEAT UNTIL ALL CHECKED
01CF'  79                        LD      A,C
01D0'  B0                        OR      B
01D1'  20 EF                     JR      NZ,CHKLP3
01D3'  3E 01                     LD      A,01H           ; SET TO FLASH DRIVE A LED
01D5'  FD 21 01DC'               LD      IY,NXADR3
01D9'  C3 029D'                  JP      FLHLP
01DC'                    NXADR3:
01DC'  AF                        XOR     A       ; CLEAR CARRY FLAG
01DD'  CB 23                     SLA     E       ; UPDATE PATTERN
01DF'  30 C5                     JR      NC,NXPAT3
01E1'  08                        EX      AF,AF'
01E2'  3E 01                     LD      A,1             ; SET FOR BACKWARD DIRECTION
01E4'  08                        EX      AF,AF'
01E5'  1E 01                     LD      E,01H           ; INITIALIZE TEST PATTERN
01E7'                    NXPAT5:
01E7'  21 FFFF                   LD      HL,M3BEG        ; GET STARTING ADDRESS
01EA'  01 8000                   LD      BC,M1SIZE       ; SET UP LOOP COUNTER
01ED'                    LDLP5:
01ED'  73                        LD      (HL),E          ; LOAD PATTERN IN MEMORY
01EE'  2B                        DEC     HL      ; DECREMENT MEMORY INDEX
01EF'  0B                        DEC     BC      ; REPEAT UNTIL ALL LOADED
01F0'  78                        LD      A,B
01F1'  B1                        OR      C
01F2'  20 F9                     JR      NZ,LDLP5
01F4'  3E 01                     LD      A,01H           ; SET TO FLASH DRIVE A LED
01F6'  FD 21 01FD'               LD      IY,CHECK5       ; SET RETURN ADDRESS
01FA'  C3 029D'                  JP      FLHLP           ; FLASH LED
01FD'                    CHECK5:
01FD'  21 8000                   LD      HL,M1BEG        ; COMPUTE END ADDRESS
0200'  01 8000                   LD      BC,M1SIZE       ; SET UP LOOP COUNTER
0203'                    CHKLP5:
0203'  7B                        LD      A,E             ; GET TEST PATTERN
```

```
0204'  56                        LD      D,(HL)          ; GET MEMORY CONTENT
0205'  BA                        CP      D       ; COMPARE WITH PATTERN
0206'  20 24                     JR      NZ,M3ERR
0208'  2F                        CPL             ; COMPLEMENT PATTERN
0209'  77                        LD      (HL),A          ; LOAD IN MEMORY
020A'  56                        LD      D,(HL)
020B'  BA                        CP      D       ; AND CHECK IMMEDIATELY
020C'  20 1E                     JR      NZ,M3ERR
020E'                    NXTAD5:
020E'  23                        INC     HL      ; INCREMENT MEMORY INDEX
020F'  0B                        DEC     BC      ; REPEAT UNTIL ALL CHECKED
0210'  79                        LD      A,C
0211'  B0                        OR      B
0212'  20 EF                     JR      NZ,CHKLP5
0214'  3E 01                     LD      A,01H           ; SET TO FLASH DRIVE A LED
0216'  FD 21 021D'               LD      IY,NXADR5
021A'  C3 029D'                  JP      FLHLP
021D'                    NXADR5:
021D'  AF                        XOR     A       ; CLEAR CARRY FLAG
021E'  CB 23                     SLA     E       ; UPDATE PATTERN
0220'  30 C5                     JR      NC,NXPAT5
0222'  ED 57                     LD      A,I
0224'  CB 57                     BIT     2,A             ; CHECK IF ERROR IN UPPER MEMORY
0226'  CA 02B8'                  JP      Z,M3OK
0229'  C3 01A1'                  JP      TSTMEM          ; REPEAT FOREVER
022C'                    M3ERR:
022C'  ED 57                     LD      A,I             ; SET ERROR INDICATOR
022E'  CB D7                     SET     2,A
0230'  ED 47                     LD      I,A
0232'  DD 21 032A'               LD      IX,M1HDG        ; DISPLAY ERROR MESSAGE
0236'  FD 21 023D'               LD      IY,M3ERR1
023A'  C3 02E2'                  JP      PMSG
023D'                    M3ERR1:
023D'  7C                        LD      A,H             ; DISPLAY MEMORY ADDRESS
023E'  FD 21 0245'               LD      IY,M3ERR3
0242'  C3 0446'                  JP      B2H
0245'                    M3ERR3:
0245'  7D                        LD      A,L
0246'  FD 21 024D'               LD      IY,M3ERR4
024A'  C3 0446'                  JP      B2H
024D'                    M3ERR4:
024D'  DD 21 0346'               LD      IX,CONTENT      ; DISPLAY MEMORY CONTENT
0251'  FD 21 0258'               LD      IY,M3ERR5
0255'  C3 02E2'                  JP      PMSG
0258'                    M3ERR5:
0258'  7A                        LD      A,D
0259'  FD 21 0260'               LD      IY,M3ERR6
025D'  C3 0446'                  JP      B2H
0260'                    M3ERR6:
0260'  DD 21 0354'               LD      IX,PATTERN      ; DISPLAY TEST PATTERN
0264'  FD 21 026B'               LD      IY,M3ERR7
0268'  C3 02F0'                  JP      PMSGL1
026B'                    M3ERR7:
026B'  7B                        LD      A,E
026C'  FD 21 0273'               LD      IY,M3ERR8
0270'  C3 0446'                  JP      B2H
```

```
0273'                           M3ERR8:
0273'  DD 21 0360'                     LD      IX,PROMPT
0277'  FD 21 027E'                     LD      IY,M3ERR9
027B'  C3 02E2'                        JP      PMSG
027E'                           M3ERR9:
027E'  3E 03                           LD      A,03H           ; SET TO FLASH BOTH DRIVES LED
0280'  FD 21 0286'                     LD      IY,M3ER10
0284'  18 17                           JR      FLHLP
0286'                           M3ER10:
0286'  DB 09                           IN      A,(KEYBST)      ; CHECK IF ANY KEY PRESS
0288'  E6 0F                           AND     KEYBMK
028A'  28 F2                           JR      Z,M3ERR9        ; REPEAT FLASHING LED
028C'  DB 08                           IN      A,(KEYBDA)      ; CLEAR KEY
028E'  08                              EX      AF,AF'
028F'  FD 21 01CD'                     LD      IY,NXTAD3       ; SET RETURN ADDRESS FOR FORWARD TEST
0293'  B7                              OR      A       ; CHECK IF GOING FORWARD OR BACKWARD
0294'  28 04                           JR      Z,NXTFWD
0296'  FD 21 020E'                     LD      IY,NXTAD5       ; SET RETURN ADDRESS FOR BACKWARD TEST
029A'                           NXTFWD:
029A'  08                              EX      AF,AF'
029B'  FD E9                           JP      (IY)            ; GO TO NEXT LOCATION
                                ;
                                ;       FLASH DRIVE A LED
                                ;
029D'                           FLHLP:
029D'  D9                              EXX             ; USE ALTERNATE REGISTER SET
029E'  67                              LD      H,A             ; SAVE LED INDICATION
029F'  2E 02                           LD      L,2
02A1'                           FLHL10:
02A1'  7C                              LD      A,H             ; FLASH LED
02A2'  D3 18                           OUT     (CMDREG),A
02A4'                           FLHLP1:
02A4'  10 FE                           DJNZ    FLHLP1
02A6'  0D                              DEC     C
02A7'  20 FB                           JR      NZ,FLHLP1
02A9'  AF                              XOR     A
02AA'  D3 18                           OUT     (CMDREG),A
02AC'                           FLHLP2:
02AC'  10 FE                           DJNZ    FLHLP2
02AE'  0D                              DEC     C
02AF'  20 FB                           JR      NZ,FLHLP2
02B1'  2D                              DEC     L
02B2'  20 ED                           JR      NZ,FLHL10
02B4'  D9                              EXX
02B5'  AF                              XOR     A
02B6'  FD E9                           JP      (IY)            ; RETURN
                                ;
                                ;       MOVE CODE TO UPPER MEMORY
                                ;
02B8'                           M30K:
02B8'  21 04A0'                        LD      HL,LAST         ; COMPUTE SIZE OF LOWER HALF OF CODE
02BB'  01 0000                         LD      BC,PROMST
02BE'  AF                              XOR     A
02BF'  ED 42                           SBC     HL,BC
02C1'  44                              LD      B,H
02C2'  4D                              LD      C,L
```

```
02C3'  21 0000              LD    HL,PROMST     ; MOVE LOWER HALF FIRST
02C6'  11 F000              LD    DE,PROMAD
02C9'  ED B0                LDIR
02CB'  21 FFFF              LD    HL,MEMEND     ; COMPUTE SIZE OF UPPER HALF OF CODE
02CE'  23                   INC   HL
02CF'  01 F4A0              LD    BC,ROMSAD
02D2'  AF                   XOR   A
02D3'  ED 42                SBC   HL,BC
02D5'  44                   LD    B,H
02D6'  4D                   LD    C,L
02D7'  21 04A0'             LD    HL,LAST       ; MOVE UPPER HALF
02DA'  11 F4A0              LD    DE,ROMSAD
02DD'  ED B0                LDIR          ; MOVE BLOCK
                    ;
                    ;       JUMP TO RAM AREA TO START TESTING
                    ;
02DF'  C3 F4A0'             JP    ACTTST
                    ;
                    ;       PRINT OUT A MESSAGE
                    ;
02E2'               PMSG:
02E2'  DD 7E 00             LD    A,(IX)        ; GET CURSOR POSITION
02E5'  D3 04                OUT   (VTCCU1),A    ; AND SET CURSOR
02E7'  DD 23                INC   IX
02E9'  DD 7E 00             LD    A,(IX)
02EC'  D3 05                OUT   (VTCCU2),A
02EE'  DD 23                INC   IX
02F0'               PMSGL1:
02F0'  DD 7E 00             LD    A,(IX)        ; GET CHARACTER FROM MESSAGE
02F3'  B7                   OR    A       ; CHECK IF NO MORE
02F4'  28 10                JR    Z,PMSGRT
02F6'  D3 1C                OUT   (VTODAT),A    ; DISPLAY ONTO PVTC
02F8'  3E AB                LD    A,WRINC
02FA'  D3 01                OUT   (VTCCMD),A
02FC'               PMSGL2:
02FC'  DB 01                IN    A,(VTCSTA)    ; GET STATUS REGISTER
02FE'  E6 20                AND   VTCRDY        ; CHECK IF READY
0300'  28 FA                JR    Z,PMSGL2      ; REPEAT IF NOT
0302'  DD 23                INC   IX      ; POINT TO NEXT CHARACTER
0304'  18 EA                JR    PMSGL1
0306'               PMSGRT:
0306'  FD E9                JP    (IY)          ; SIMULATED RETURN
                    ;
                    ;       MESSAGES
                    ;
0308'  0224         SIGNON: DW    VTCBEG+420
030A'  50 49 45 44          DB    'PIED PIPER',0
030E'  20 50 49 50
0312'  45 52 00
0315'  03FA 0000    BADSYS: DW    VTCBEG+480+410,0
0319'  02BA         TMMSG:  DW    VTCBEG+160+410
031B'  4D 65 6D 6F          DB    'Memory test...',0
031F'  72 79 20 74
0323'  65 73 74 2E
0327'  2E 2E 00
032A'  030A         M1HDG:  DW    VTCBEG+240+410
```

```
032C'  4D 65 6D 6F              DB      'Memory error at address: ',0
0330'  72 79 20 65
0334'  72 72 6F 72
0338'  20 61 74 20
033C'  61 64 64 72
0340'  65 73 73 3A
0344'  20 00
0346'  035A             CONTENT:DW      VTCBEG+320+410
0348'  20 20 43 6F              DB      '  Content: ',0
034C'  6E 74 65 6E
0350'  74 3A 20 00
0354'  20 20 50 61      PATTERN:DB      '  Pattern: ',0
0358'  74 74 65 72
035C'  6E 3A 20 00
0360'  03AA             PROMPT: DW      VTCBEG+400+410
0362'  50 72 65 73              DB      'Press any key to continue. ',0
0366'  73 20 61 6E
036A'  79 20 6B 65
036E'  79 20 74 6F
0372'  20 63 6F 6E
0376'  74 69 6E 75
037A'  65 2E 20 00
                        ;
                        ;       4.  CHECK IF WARM BOOT
                        ;
037E'                   SHOW:
037E'  ED 57                    LD      A,I             ; SET DISPLAY REQUIRED BIT
0380'  CB FF                    SET     7,A
0382'  ED 47                    LD      I,A
0384'                   NODISP:
0384'  21 FF93'                 LD      HL,CHKSUM       ; COMPUTE LENGTH OF BLOCK TO COMPARE
0387'  0E 10                    LD      C,10H
0389'  06 FB                    LD      B,ITRVEC        ; STARTING AFTER INTERRUPT VECTORS
038B'  AF                       XOR     A
038C'  ED 42                    SBC     HL,BC
038E'  44                       LD      B,H
038F'  4D                       LD      C,L
0390'  26 FB                    LD      H,ITRVEC        ; COMPUTE START OFFSET OF ROM
0392'  2E 10                    LD      L,10H
0394'  11 F4A0                  LD      DE,ROMSAD
0397'  AF                       XOR     A
0398'  ED 52                    SBC     HL,DE
039A'  11 04A0'                 LD      DE,LAST-PROMST
039D'  19                       ADD     HL,DE
039E'  16 FB                    LD      D,ITRVEC        ; SET START OFFSET IN RAM
03A0'  1E 10                    LD      E,10H
03A2'                   WAITL7:
03A2'  1A                       LD      A,(DE)
03A3'  BE                       CP      (HL)
03A4'  20 10                    JR      NZ,WAITL9       ; PROCEED NORMAL DIAGNOTIC IF NOT SAME
03A6'  23                       INC     HL
03A7'  13                       INC     DE
03A8'  0B                       DEC     BC
03A9'  78                       LD      A,B
03AA'  B1                       OR      C
03AB'  20 F5                    JR      NZ,WAITL7       ; REPEAT UNTIL WHOLE BLOCK CHECKED
```

```
03AD'  ED 57                    LD      A,I             ; SET INDICATOR FOR LATER
03AF'  CB F7                    SET     6,A
03B1'  ED 47                    LD      I,A
03B3'  C3 0476'                 JP      MOVE            ; THEN PROCEED DIRECTLY TO BOOT
                        ;
                        ;       5.  MEMORY TEST (LOCATION 8000H TO FFFFH)
                        ;
03B6'                   WAITL9:
03B6'  1E A5                    LD      E,TSTPAT        ; INITIALIZE TEST PATTERN
03B8'                   NXPAT1:
03B8'  21 8000                  LD      HL,M1BEG        ; GET STARTING ADDRESS
03BB'  01 8000                  LD      BC,M1SIZE       ; SET UP LOOP COUNTER
03BE'                   LDLP1:
03BE'  73                       LD      (HL),E          ; LOAD PATTERN IN MEMORY
03BF'  23                       INC     HL      ; INCREMENT MEMORY INDEX
03C0'  0B                       DEC     BC      ; REPEAT UNTIL ALL LOADED
03C1'  78                       LD      A,B
03C2'  B1                       OR      C
03C3'  20 F9                    JR      NZ,LDLP1
03C5'  21 8000                  LD      HL,M1BEG        ; COMPUTE END ADDRESS
03C8'  01 8000                  LD      BC,M1SIZE
03CB'  09                       ADD     HL,BC
03CC'  2B                       DEC     HL      ; GET STARTING ADDRESS FOR CHECKINF
03CD'  01 8000                  LD      BC,M1SIZE       ; SET UP LOOP COUNTER
03D0'                   CHKLP1:
03D0'  7B                       LD      A,E             ; GET TEST PATTERN
03D1'  56                       LD      D,(HL)          ; GET MEMORY CONTENT
03D2'  BA                       CP      D       ; COMPARE WITH PATTERN
03D3'  20 14                    JR      NZ,M1ERR
03D5'  2F                       CPL             ; COMPLEMENT PATTERN
03D6'  77                       LD      (HL),A          ; LOAD IN MEMORY
03D7'  56                       LD      D,(HL)
03D8'  BA                       CP      D       ; AND CHECK IMMEDIATELY
03D9'  20 0E                    JR      NZ,M1ERR
03DB'  2B                       DEC     HL      ; DECREMENT MEMORY INDEX
03DC'  0B                       DEC     BC      ; REPEAT UNTIL ALL CHECKED
03DD'  78                       LD      A,B
03DE'  B1                       OR      C
03DF'  20 EF                    JR      NZ,CHKLP1
03E1'  AF                       XOR     A       ; CLEAR CARRY FLAG
03E2'  CB 23                    SLA     E       ; UPDATE PATTERN
03E4'  30 D2                    JR      NC,NXPAT1
03E6'  C3 0476'                 JP      M1OK
03E9'                   M1ERR:
03E9'  ED 57                    LD      A,I             ; GET ERROR INDICATOR
03EB'  CB D7                    SET     2,A             ; SET UPPER MEMORY ERROR BIT
03ED'  ED 47                    LD      I,A             ; SAVE INDICATOR
03EF'  CB 7F                    BIT     7,A             ; CHECK IF DISPLAY REQUIRED
03F1'  28 3D                    JR      Z,M1ERR8
03F3'  DD 21 032A'              LD      IX,M1HDG
03F7'  FD 21 03FE'              LD      IY,M1ERR1
03FB'  C3 02E2'                 JP      PMSG
03FE'                   M1ERR1:
03FE'  7C                       LD      A,H             ; DISPLAY MEMORY ADDRESS
03FF'  FD 21 0405'              LD      IY,M1ERR3
0403'  18 41                    JR      B2H
```

```
0405'                           M1ERR3:
0405'  7D                               LD      A,L
0406'  FD 21 040C'                     LD      IY,M1ERR4
040A'  18 3A                            JR      B2H
040C'                           M1ERR4:
040C'  DD 21 0346'                     LD      IX,CONTENT      ; DISPLAY MEMORY CONTENT
0410'  FD 21 0417'                     LD      IY,M1ERR5
0414'  C3 02E2'                        JP      PMSG
0417'                           M1ERR5:
0417'  7A                               LD      A,D
0418'  FD 21 041E'                     LD      IY,M1ERR6
041C'  18 28                            JR      B2H
041E'                           M1ERR6:
041E'  DD 21 0354'                     LD      IX,PATTERN      ; DISPLAY TEST PATTERN
0422'  FD 21 0429'                     LD      IY,M1ERR7
0426'  C3 02F0'                        JP      PMSGL1
0429'                           M1ERR7:
0429'  7B                               LD      A,E
042A'  FD 21 0430'                     LD      IY,M1ERR8
042E'  18 16                            JR      B2H
0430'                           M1ERR8:
0430'  DD 21 0315'                     LD      IX,BADSYS
0434'  FD 21 043B'                     LD      IY,M1ERR9
0438'  C3 02E2'                        JP      PMSG
043B'                           M1ERR9:
043B'  DD 21 07AF'                     LD      IX,FAILMSG-ROMSAD+LAST-PROMST
043F'  FD 21 09B7'                     LD      IY,INDBAD-ROMSAD+LAST-PROMST
0443'  C3 02F0'                        JP      PMSGL1
                                ;
                                ;       CONVERT A BINARY NUMBER TO HEX
                                ;
0446'                           B2H:
0446'  47                               LD      B,A             ; SAVE NUMBER FOR LATER
0447'  CB 3F                           SRL     A       ; OBTAIN UPPER NIBBLE
0449'  CB 3F                           SRL     A
044B'  CB 3F                           SRL     A
044D'  CB 3F                           SRL     A
044F'  DD 21 0455'                     LD      IX,B2HLSN
0453'  18 0B                           JR      B2HEXE          ; CONVERT UPPER NIBBLE
0455'                           B2HLSN:
0455'  78                               LD      A,B             ; RETRIEVE NUMBER
0456'  E6 0F                           AND     0FH             ; OBTAIN LOWER NIBBLE
0458'  DD 21 045E'                     LD      IX,B2HRET
045C'  18 02                           JR      B2HEXE
045E'                           B2HRET:
045E'  FD E9                           JP      (IY)            ; RETURN
0460'                           B2HEXE:
0460'  C6 30                           ADD     A,30H           ; CONVERT TO ASCII NUMBER
0462'  FE 3A                           CP      3AH             ; CHECK IF OVER 9
0464'  38 02                           JR      C,DIGIT
0466'  C6 07                           ADD     A,7             ; CONVERT TO ALPHA
0468'                           DIGIT:
0468'  D3 1C                           OUT     (VTODAT),A      ; DISPLAY DIGIT
046A'  3E AB                           LD      A,WRINC
046C'  D3 01                           OUT     (VTCCMD),A
046E'                           B2HLP:
```

```
046E'  DB 01                    IN      A,(VTCSTA)      ; GET STATUS
0470'  E6 20                    AND     VTCRDY          ; WAIT UNTIL NOT BUSY
0472'  28 FA                    JR      Z,B2HLP
0474'  DD E9                    JP      (IX)            ; RETURN
                        ;
0476'                   M1OK:
                        ;
                        ;       6.  TO MOVE PROGRAM FROM ROM TO UPPER RAM AREA
                        ;
0476'                   MOVE:
0476'  21 04A0'                 LD      HL,LAST         ; COMPUTE SIZE OF LOWER HALF OF CODE
0479'  01 0000                  LD      BC,PROMST
047C'  AF                       XOR     A
047D'  ED 42                    SBC     HL,BC
047F'  44                       LD      B,H
0480'  4D                       LD      C,L
0481'  21 0000                  LD      HL,PROMST       ; MOVE LOWER HALF FIRST
0484'  11 F000                  LD      DE,PROMAD
0487'  ED B0                    LDIR
0489'  21 FFFF                  LD      HL,MEMEND       ; COMPUTE SIZE OF UPPER HALF OF CODE
048C'  23                       INC     HL
048D'  01 F4A0                  LD      BC,ROMSAD
0490'  AF                       XOR     A
0491'  ED 42                    SBC     HL,BC
0493'  44                       LD      B,H
0494'  4D                       LD      C,L
0495'  21 04A0'                 LD      HL,LAST         ; MOVE UPPER HALF
0498'  11 F4A0                  LD      DE,ROMSAD
049B'  ED B0                    LDIR            ; MOVE BLOCK
                        ;
                        ;       JUMP TO RAM AREA TO START EXECUTING
                        ;
049D'  C3 F697'                 JP      AFMOVE
04A0'                   LAST    EQU     $
                        ;
                                ORG     ROMSAD+PROMST
                        ;
                        ;       ACTIVATE TEST
                        ;
F4A0'                   ACTTST:
F4A0'  DB 18                    IN      A,(STAREG)      ; DISABLE PROM
F4A2'  3E 6D                    LD      A,06DH          ; CHECK IF PROM DISABLED
F4A4'  32 0000                  LD      (PROMST),A      ; WRITE PATTERN TO MEMORY
F4A7'  47                       LD      B,A             ; SAVE PATTERN
F4A8'  3A 0000                  LD      A,(PROMST)      ; READ BACK
F4AB'  B8                       CP      B
F4AC'  C2 0071'                 JP      NZ,CPUERR       ; HALT IF PROM STILL ENABLED
F4AF'  ED 57                    LD      A,I
F4B1'  CB 5F                    BIT     3,A             ; CHECK IF DISK TEST
F4B3'  CA F5A1'                 JP      Z,TESTM         ; BRANCH IF MEMORY TEST IF NOT
                        ;
                        ;       FLOPPY DISK TEST
                        ;
F4B6'  31 F000                  LD      SP,PROMAD       ; SET UP STACK POINTER
F4B9'  21 021A                  LD      HL,VTCBEG+410   ; SYNCHRONIZE PVTC REGISTERS
F4BC'  CD F892'                 CALL    SETCUR
```

```
F4BF'  DD 21 F72E'              LD      IX,DTMSG        ; DISPLAY DISK TEST MESSAGE
F4C3'  CD F84E'                 CALL    OUTMSG
F4C6'                   DTLP1:
F4C6'  DB 09                    IN      A,(KEYBST)      ; GET KEYBOARD STATUS
F4C8'  E6 0F                    AND     KEYBMK          ; CHECK IF KEY ENTERED
F4CA'  28 FA                    JR      Z,DTLP1         ; WAIT UNTIL KEY ENTERED
F4CC'  DB 08                    IN      A,(KEYBDA)      ; CLEAR BUFFER
F4CE'  3E FB                    LD      A,ITRVEC        ; INITIALIZE INTERRUPT VECTOR
F4D0'  ED 47                    LD      I,A
                        ;       IM2             ; SET FOR MODE TWO
F4D2'  ED 5E                    DB      0EDH,5EH
F4D4'  3E 01                    LD      A,DSKA          ; SELECT DRIVE A
F4D6'  CD FAA5'                 CALL    CHECK           ; CHECK IF DRIVE FUNCTIONING
F4D9'  B7                       OR      A       ; SKIP TEST IF ERROR
F4DA'  CA F9A7'                 JP      Z,LOCAL
F4DD'  CD FA60'                 CALL    SPEED           ; CHECK DRIVE SPEED
F4E0'  B7                       OR      A       ; SKIP TEST IF ERROR
F4E1'  C2 F9A7'                 JP      NZ,LOCAL
F4E4'  3E 01                    LD      A,DSKA          ; RESET TO SIDE 0 AND TRACK 0
F4E6'  CD FC62'                 CALL    DSKSEL
F4E9'  1E 0C                    LD      E,RSCMD+VERIFY+HDLOAD
F4EB'  3A FFD1'                 LD      A,(STEPS)
F4EE'  B3                       OR      E
F4EF'  CD FD06'                 CALL    TYPE1
F4F2'  3E 28                    LD      A,40            ; SET TO SEEK CYLINDER 40
F4F4'  D3 13                    OUT     (FDDATA),A
F4F6'  1E 1C                    LD      E,SKCMD+VERIFY+HDLOAD
F4F8'  3A FFD1'                 LD      A,(STEPS)
F4FB'  B3                       OR      E
F4FC'  CD FD06'                 CALL    TYPE1           ; SEEK CYLINDER 40
F4FF'  11 0028                  LD      DE,40           ; DEFAULT MAXIMUM CYLINDER TO 40
F502'  E6 10                    AND     RNF             ; CHECK IF RECORD NOT FOUND
F504'  20 03                    JR      NZ,DVFY3
F506'  11 0050                  LD      DE,80           ; SET MAXIMUM CYLINDER TO 80
F509'                   DVFY3:
F509'  3A FFD3'                 LD      A,(SELBYT)      ; CHECK IF DOUBLE SIDED
F50C'  CB 6F                    BIT     DSIDE,A
F50E'  28 04                    JR      Z,DVFY9         ; SAVE LIMIT IF NOT
F510'  CB 23                    SLA     E       ; MULTIPLY LIMIT BY 2
F512'  CB 12                    RL      D
F514'                   DVFY9:
F514'  ED 53 FF9F'              LD      (MAXTRK),DE     ; SAVE LIMIT FOR LATER
F518'                   DTLP2:
F518'  3E 01                    LD      A,DSKA          ; SELECT SIDE 0 OF DRIVE A
F51A'  CD FC62'                 CALL    DSKSEL
F51D'  1E 0C                    LD      E,RSCMD+VERIFY+HDLOAD   ; RESTORE DRIVE
F51F'  3A FFD1'                 LD      A,(STEPS)
F522'  B3                       OR      E
F523'  CD FD06'                 CALL    TYPE1
F526'  11 0000                  LD      DE,0            ; START FROM TRACK 0
F529'  ED 53 FFA1'              LD      (TRKNBR),DE     ; AND SAVE FOR LATER
F52D'  3E 01                    LD      A,1             ; START FROM SECTOR 1
F52F'  0E 82                    LD      C,RDCMD         ; SET FOR READING SIDE 0
F531'                   DVFY1:
F531'  32 FFA3'                 LD      (SECNBR),A
F534'  D3 12                    OUT     (FDSEC),A       ; SET UP SECTOR REGISTER
```

```
F536'                       DVFY2:
F536'  79                           LD      A,C
F537'  21 8000                      LD      HL,VFYBUF
F53A'  CD FC78'                     CALL    FDRWS           ; READ IN SECTOR
F53D'  B7                           OR      A       ; CHECK IF ERROR
F53E'  28 03                        JR      Z,DVFY4
F540'  CD F597'                     CALL    VFYERR
F543'                       DVFY4:
F543'  3A FFA3'                     LD      A,(SECNBR)      ; ADVANCE TO NEXT SECTOR
F546'  3C                           INC     A
F547'  FE 09                        CP      9       ; CHECK IF READ IN EIGHT SECTORS ALREADY
F549'  20 E6                        JR      NZ,DVFY1
F54B'  ED 5B FFA1'                  LD      DE,(TRKNBR)     ; ADVANCE TO NEXT TRACK
F54F'  13                           INC     DE
F550'  2A FF9F'                     LD      HL,(MAXTRK)     ; GET TRACK LIMIT
F553'  AF                           XOR     A
F554'  ED 52                        SBC     HL,DE           ; CHECK IF LIMIT REACH
F556'  28 C0                        JR      Z,DTLP2         ; REPEAT TEST IF SO
F558'  ED 53 FFA1'                  LD      (TRKNBR),DE
F55C'  3A FFD3'                     LD      A,(SELBYT)      ; GET NUMBER OF SIDES
F55F'  CB 6F                        BIT     DSIDE,A         ; CHECK IF ONE SIDE
F561'  28 1C                        JR      Z,DVFY5
F563'  3A FFE2'                     LD      A,(IOREG)       ; GET CURRENT SELECT BYTE
F566'  CB 57                        BIT     SIDE1,A         ; CHECK CURRENT SIDE
F568'  20 08                        JR      NZ,DVFY6
F56A'  06 01                        LD      B,1             ; SET SIDE 1
F56C'  0E 8A                        LD      C,RDCMD+8
F56E'  CB D7                        SET     SIDE1,A
F570'  18 06                        JR      DVFY7
F572'                       DVFY6:
F572'  06 00                        LD      B,0             ; SET SIDE 0
F574'  0E 82                        LD      C,RDCMD
F576'  CB 97                        RES     SIDE1,A
F578'                       DVFY7:
F578'  CD FC62'                     CALL    DSKSEL          ; SELECT PROPER SIDE
F57B'  78                           LD      A,B             ; CHECK IF STEP REQUIRED
F57C'  B7                           OR      A
F57D'  20 B2                        JR      NZ,DVFY1
F57F'                       DVFY5:
F57F'  1E 58                        LD      E,SICMD+HDLOAD  ; SET UP STEP COMMAND
F581'  3A FFD1'                     LD      A,(STEPS)
F584'  B3                           OR      E
F585'  CD FD06'                     CALL    TYPE1
F588'  3A FFA1'                     LD      A,(TRKNBR)      ; GET TRACK NUMBER
F58B'  C6 32                        ADD     A,50            ; MINIMUM DELAY IS 50 MSED
F58D'                       DVFY8:
F58D'  CD FD60'                     CALL    DELAY
F590'  3D                           DEC     A
F591'  20 FA                        JR      NZ,DVFY8
F593'  3E 01                        LD      A,1             ; RESET TO SECTOR ONE
F595'  18 9A                        JR      DVFY1           ; CONTINUE TEST
                            ;
                            ;       HANDLE DISK TEST ERROR
                            ;
F597'                       VFYERR:
F597'  D3 1C                        OUT     (VTODAT),A      ; DISPLAY ERROR CODE
```

```
F599'  3E AB                    LD      A,WRINC
F59B'  D3 01                    OUT     (VTCCMD),A
F59D'  CD F8C2'                 CALL    WAIT
F5A0'  C9                       RET
                        ;
                        ;       MEMORY TEST
                        ;
F5A1'                   TESTM:
F5A1'  08                       EX      AF,AF'          ; SET FOR FORWARD DIRECTION
F5A2'  AF                       XOR     A
F5A3'  08                       EX      AF,AF'
F5A4'  1E 01                    LD      E,01H           ; INITIALIZE TEST PATTERN
F5A6'                   NXPAT4:
F5A6'  21 0000                  LD      HL,M2BEG        ; GET STARTING ADDRESS
F5A9'  01 F000                  LD      BC,M2SIZE       ; SET UP LOOP COUNTER
F5AC'                   LDLP4:
F5AC'  73                       LD      (HL),E          ; LOAD PATTERN IN MEMORY
F5AD'  23                       INC     HL      ; INCREMENT MEMORY INDEX
F5AE'  0B                       DEC     BC      ; REPEAT UNTIL ALL LOADED
F5AF'  78                       LD      A,B
F5B0'  B1                       OR      C
F5B1'  20 F9                    JR      NZ,LDLP4
F5B3'  3E 02                    LD      A,02H           ; SET TO FLASH DRIVE A LED
F5B5'  FD 21 F5BC'              LD      IY,CHECK4       ; SET RETURN ADDRESS
F5B9'  C3 F29D'                 JP      FLHLP+PROMAD-PROMST
F5BC'                   CHECK4:
F5BC'  21 EFFF                  LD      HL,M4BEG        ; COMPUTE END ADDRESS
F5BF'  01 F000                  LD      BC,M2SIZE       ; SET UP LOOP COUNTER
F5C2'                   CHKLP4:
F5C2'  7B                       LD      A,E             ; GET TEST PATTERN
F5C3'  56                       LD      D,(HL)          ; GET MEMORY CONTENT
F5C4'  BA                       CP      D       ; COMPARE WITH PATTERN
F5C5'  20 5E                    JR      NZ,M4ERR
F5C7'  2F                       CPL             ; COMPLEMENT PATTERN
F5C8'  77                       LD      (HL),A          ; LOAD IN MEMORY
F5C9'  56                       LD      D,(HL)
F5CA'  BA                       CP      D       ; AND CHECK IMMEDIATELY
F5CB'  20 58                    JR      NZ,M4ERR
F5CD'                   NXTAD4:
F5CD'  2B                       DEC     HL      ; DECREMENT MEMORY INDEX
F5CE'  0B                       DEC     BC      ; REPEAT UNTIL ALL CHECKED
F5CF'  79                       LD      A,C
F5D0'  B0                       OR      B
F5D1'  20 EF                    JR      NZ,CHKLP4
F5D3'  3E 02                    LD      A,02H           ; SET TO FLASH DRIVE A LED
F5D5'  FD 21 F5DC'              LD      IY,NXADR4
F5D9'  C3 F29D'                 JP      FLHLP+PROMAD-PROMST
F5DC'                   NXADR4:
F5DC'  AF                       XOR     A       ; CLEAR CARRY FLAG
F5DD'  CB 23                    SLA     E       ; UPDATE PATTERN
F5DF'  30 C5                    JR      NC,NXPAT4
F5E1'  08                       EX      AF,AF'
F5E2'  3E 01                    LD      A,1             ; SET FOR BACKWARD DIRECTION
F5E4'  08                       EX      AF,AF'
F5E5'  1E 01                    LD      E,01H           ; INITIALIZE TEST PATTERN
F5E7'                   NXPAT6:
```

```
F5E7'  21 EFFF                    LD      HL,M4BEG        ; GET STARTING ADDRESS
F5EA'  01 F000                    LD      BC,M2SIZE       ; SET UP LOOP COUNTER
F5ED'                       LDLP6:
F5ED'  73                         LD      (HL),E          ; LOAD PATTERN IN MEMORY
F5EE'  2B                         DEC     HL      ; DECREMENT MEMORY INDEX
F5EF'  0B                         DEC     BC      ; REPEAT UNTIL ALL LOADED
F5F0'  78                         LD      A,B
F5F1'  B1                         OR      C
F5F2'  20 F9                      JR      NZ,LDLP6
F5F4'  3E 02                      LD      A,02H           ; SET TO FLASH DRIVE B LED
F5F6'  FD 21 F5FD'                LD      IY,CHECK6       ; SET RETURN ADDRESS
F5FA'  C3 F29D'                   JP      FLHLP+PROMAD-PROMST
F5FD'                       CHECK6:
F5FD'  21 0000                    LD      HL,M2BEG        ; COMPUTE END ADDRESS
F600'  01 F000                    LD      BC,M2SIZE       ; SET UP LOOP COUNTER
F603'                       CHKLP6:
F603'  7B                         LD      A,E             ; GET TEST PATTERN
F604'  56                         LD      D,(HL)          ; GET MEMORY CONTENT
F605'  BA                         CP      D       ; COMPARE WITH PATTERN
F606'  20 1D                      JR      NZ,M4ERR
F608'  2F                         CPL             ; COMPLEMENT PATTERN
F609'  77                         LD      (HL),A          ; LOAD IN MEMORY
F60A'  56                         LD      D,(HL)
F60B'  BA                         CP      D       ; AND CHECK IMMEDIATELY
F60C'  20 17                      JR      NZ,M4ERR
F60E'                       NXTAD6:
F60E'  23                         INC     HL      ; INCREMENT MEMORY INDEX
F60F'  0B                         DEC     BC      ; REPEAT UNTIL ALL CHECKED
F610'  79                         LD      A,C
F611'  B0                         OR      B
F612'  20 EF                      JR      NZ,CHKLP6
F614'  3E 02                      LD      A,02H           ; SET TO FLASH DRIVE B LED
F616'  FD 21 F61D'                LD      IY,NXADR6
F61A'  C3 F29D'                   JP      FLHLP+PROMAD-PROMST
F61D'                       NXADR6:
F61D'  AF                         XOR     A       ; CLEAR CARRY FLAG
F61E'  CB 23                      SLA     E       ; UPDATE PATTERN
F620'  30 C5                      JR      NC,NXPAT6
F622'  C3 F5A1'                   JP      TESTM           ; REPEAT UNTIL RESET
F625'                       M4ERR:
F625'  ED 57                      LD      A,I             ; SET ERROR INDICATOR
F627'  CB D7                      SET     2,A
F629'  ED 47                      LD      I,A
F62B'  DD 21 F32A'                LD      IX,M1HDG+PROMAD-PROMST
F62F'  FD 21 F636'                LD      IY,M4ERR1
F633'  C3 F2E2'                   JP      PMSG+PROMAD-PROMST
F636'                       M4ERR1:
F636'  7C                         LD      A,H             ; DISPLAY MEMORY ADDRESS
F637'  FD 21 F63E'                LD      IY,M4ERR3
F63B'  C3 F446'                   JP      B2H+PROMAD-PROMST
F63E'                       M4ERR3:
F63E'  7D                         LD      A,L
F63F'  FD 21 F646'                LD      IY,M4ERR4
F643'  C3 F446'                   JP      B2H+PROMAD-PROMST
F646'                       M4ERR4:
F646'  DD 21 0346'                LD      IX,CONTENT      ; DISPLAY MEMORY CONTENT
```

```
F64A'  FD 21 F651'          LD    IY,M4ERR5
F64E'  C3 F2E2'             JP    PMSG+PROMAD-PROMST
F651'                  M4ERR5:
F651'  7A                   LD    A,D
F652'  FD 21 F659'          LD    IY,M4ERR6
F656'  C3 F446'             JP    B2H+PROMAD-PROMST
F659'                  M4ERR6:
F659'  DD 21 0354'          LD    IX,PATTERN      ; DISPLAY TEST PATTERN
F65D'  FD 21 F664'          LD    IY,M4ERR7
F661'  C3 F2F0'             JP    PMSGL1+PROMAD-PROMST
F664'                  M4ERR7:
F664'  7B                   LD    A,E
F665'  FD 21 F66C'          LD    IY,M4ERR8
F669'  C3 F446'             JP    B2H+PROMAD-PROMST
F66C'                  M4ERR8:
F66C'  DD 21 0360'          LD    IX,PROMPT
F670'  FD 21 F677'          LD    IY,M4ERR9
F674'  C3 F2E2'             JP    PMSG+PROMAD-PROMST
F677'                  M4ERR9:
F677'  3E 03                LD    A,03H           ; SET TO FLASH BOTH DRIVES LED
F679'  FD 21 F680'          LD    IY,M4ER10
F67D'  C3 F29D'             JP    FLHLP+PROMAD-PROMST
F680'                  M4ER10:
F680'  DB 09                IN    A,(KEYBST)      ; GET KEYBOARD STATUS
F682'  E6 0F                AND   KEYBMK
F684'  28 F1                JR    Z,M4ERR9        ; REPEAT FLASHING LED
F686'  DB 08                IN    A,(KEYBDA)      ; CLEAR KEY
F688'  08                   EX    AF,AF'
F689'  FD 21 F60E'          LD    IY,NXTAD6       ; SET RETURN ADDRESS FOR BACKWARD TEST
F68D'  B7                   OR    A       ; CHECK IF GOING FORWARD OR BACKWARD
F68E'  20 04                JR    NZ,NXTBWD
F690'  FD 21 F5CD'          LD    IY,NXTAD4       ; SET RETURN ADDRESS FOR FORWARD TEST
F694'                  NXTBWD:
F694'  08                   EX    AF,AF'
F695'  FD E9                JP    (IY)            ; GO TO NEXT LOCATION
                       ;
                       ;       7.  MEMORY TEST (LOCATION 0000H TO F800H)
                       ;
F697'                  AFMOVE:
F697'  DB 18                IN    A,(STAREG)      ; DISABLE ROM
F699'  3E 6D                LD    A,06DH          ; CHECK IF PROM DISABLED
F69B'  32 0000              LD    (PROMST),A      ; WRITE PATTERN TO MEMORY
F69E'  47                   LD    B,A             ; SAVE PATTERN
F69F'  3A 0000              LD    A,(PROMST)      ; READ BACK
F6A2'  B8                   CP    B
F6A3'  C2 0071'             JP    NZ,CPUERR       ; HALT IF PROM STILL ENABLED
F6A6'  ED 57                LD    A,I             ; CHECK IF WARM BOOT
F6A8'  CB 77                BIT   6,A
F6AA'  C2 F8C9'             JP    NZ,BOOT         ; SKIP TO BOOT
F6AD'                  MTEST2:
F6AD'  1E A5                LD    E,TSTPAT        ; INITIALIZE TEST PATTERN
F6AF'                  NXPAT2:
F6AF'  21 0000              LD    HL,M2BEG        ; GET STARTING ADDRESS
F6B2'  01 F000              LD    BC,M2SIZE       ; SET UP LOOP COUNTER
F6B5'                  LDLP2:
F6B5'  73                   LD    (HL),E          ; LOAD PATTERN IN MEMORY
```

```
F6B6'  23                      INC     HL      ; INCREMENT MEMORY INDEX
F6B7'  0B                      DEC     BC      ; REPEAT UNTIL ALL LOADED
F6B8'  78                      LD      A,B
F6B9'  B1                      OR      C
F6BA'  20 F9                   JR      NZ,LDLP2
F6BC'  21 0000                 LD      HL,M2BEG        ; COMPUTE END ADDRESS
F6BF'  01 F000                 LD      BC,M2SIZE
F6C2'  09                      ADD     HL,BC
F6C3'  2B                      DEC     HL      ; GET STARTING ADDRESS FOR CHECKINF
F6C4'  01 F000                 LD      BC,M2SIZE       ; SET UP LOOP COUNTER
F6C7'                  CHKLP2:
F6C7'  7B                      LD      A,E
F6C8'  56                      LD      D,(HL)          ; GET MEMORY CONTENT
F6C9'  BA                      CP      D       ; COMPARE WITH PATTERN
F6CA'  20 13                   JR      NZ,M2ERR
F6CC'  2F                      CPL             ; COMPLEMENT PATTERN
F6CD'  77                      LD      (HL),A          ; LOAD IN MEMORY
F6CE'  56                      LD      D,(HL)
F6CF'  BA                      CP      D       ; AND CHECK IMMEDIATELY
F6D0'  20 0D                   JR      NZ,M2ERR
F6D2'  2B                      DEC     HL      ; DECREMENT MEMORY INDEX
F6D3'  0B                      DEC     BC      ; REPEAT UNTIL ALL CHECKED
F6D4'  78                      LD      A,B
F6D5'  B1                      OR      C
F6D6'  20 EF                   JR      NZ,CHKLP2
F6D8'  AF                      XOR     A       ; CLEAR CARRY FLAG
F6D9'  CB 23                   SLA     E       ; UPDATE PATTERN
F6DB'  30 D2                   JR      NC,NXPAT2
F6DD'  18 4C                   JR      M2OK
F6DF'                  M2ERR:
F6DF'  ED 57                   LD      A,I             ; GET ERROR INDICATOR
F6E1'  CB D7                   SET     2,A             ; SET UPPER MEMORY ERROR BIT
F6E3'  ED 47                   LD      I,A             ; SAVE INDICATOR
F6E5'  CB 7F                   BIT     7,A             ; CHECK IF DISPLAY REQUIRED
F6E7'  CA F72B'                JP      Z,HDTEST        ; BRANCH IF NOT
F6EA'  DD 21 F32A'             LD      IX,M1HDG+PROMAD-PROMST  ; POINT TO MEMORY BAD MESSAGE
F6EE'  FD 21 F6F5'             LD      IY,M2ERR1
F6F2'  C3 F2F0'                JP      PMSGL1+PROMAD-PROMST
F6F5'                  M2ERR1:
F6F5'  7C                      LD      A,H             ; DISPLAY MEMORY ADDRESS
F6F6'  FD 21 F6FD'             LD      IY,M2ERR3
F6FA'  C3 F446'                JP      B2H+PROMAD-PROMST
F6FD'                  M2ERR3:
F6FD'  7D                      LD      A,L
F6FE'  FD 21 F705'             LD      IY,M2ERR4
F702'  C3 F446'                JP      B2H+PROMAD-PROMST
F705'                  M2ERR4:
F705'  DD 21 0346'             LD      IX,CONTENT      ; DISPLAY MEMORY CONTENT
F709'  FD 21 F710'             LD      IY,M2ERR5
F70D'  C3 F2E2'                JP      PMSG+PROMAD-PROMST
F710'                  M2ERR5:
F710'  7A                      LD      A,D
F711'  FD 21 F718'             LD      IY,M2ERR6
F715'  C3 F446'                JP      B2H+PROMAD-PROMST
F718'                  M2ERR6:
F718'  DD 21 0354'             LD      IX,PATTERN      ; DISPLAY TEST PATTERN
```

```
F71C'  FD 21 F723'            LD      IY,M2ERR7
F720'  C3 F2F0'               JP      PMSGL1+PROMAD-PROMST
F723'                  M2ERR7:
F723'  7B                     LD      A,E
F724'  FD 21 F72B'            LD      IY,HDTEST
F728'  C3 F446'               JP      B2H+PROMAD-PROMST
F72B'                  M2OK:
                       ;
                       ;      HARD DISK CONTROLLER TEST
                       ;
F72B'                  HDTEST:
F72B'  C3 F8C9'               JP      BOOT
                       ;
                       ;      MESSAGES
                       ;
F72E'  0D 0A 0A 44     DTMSG:  DB      CR,LF,LF,'Disk test.    '
F732'  69 73 6B 20
F736'  74 65 73 74
F73A'  2E 20 20 20
F73E'  20
F73F'  0D 0A 49 6E            DB      CR,LF,'Insert formatted diskette in drive.'
F743'  73 65 72 74
F747'  20 66 6F 72
F74B'  6D 61 74 74
F74F'  65 64 20 64
F753'  69 73 6B 65
F757'  74 74 65 20
F75B'  69 6E 20 64
F75F'  72 69 76 65
F763'  2E
F764'  0D 0A 50 72            DB      CR,LF,'Press any key to start.',0
F768'  65 73 73 20
F76C'  61 6E 79 20
F770'  6B 65 79 20
F774'  74 6F 20 73
F778'  74 61 72 74
F77C'  2E 00
F77E'  0D 0A 0A 0A     SYSOK:  DB      CR,LF,LF,LF,LF,LF,LF,LF
F782'  0A 0A 0A 0A
F786'  49 6E 73 65            DB      'Insert system diskette in Drive.',0
F78A'  72 74 20 73
F78E'  79 73 74 65
F792'  6D 20 64 69
F796'  73 6B 65 74
F79A'  74 65 20 69
F79E'  6E 20 44 72
F7A2'  69 76 65 2E
F7A6'  00
F7A7'  0D 0A 0A 0A     SYSBAD: DB      CR,LF,LF,LF,LF,LF,LF,LF
F7AB'  0A 0A 0A 0A
F7AF'  45 71 75 69     FAILMSG:DB      'Equipment Failure - Return for Servicing',0
F7B3'  70 6D 65 6E
F7B7'  74 20 46 61
F7BB'  69 6C 75 72
F7BF'  65 20 2D 20
F7C3'  52 65 74 75
```

```
F7C7'  72 6E 20 66
F7CB'  6F 72 20 53
F7CF'  65 72 76 69
F7D3'  63 69 6E 67
F7D7'  00
F7D8'  0D 0A 53 79          BOOTERR:DB      CR,LF,'System error - refer to user manual.',0
F7DC'  73 74 65 6D
F7E0'  20 65 72 72
F7E4'  6F 72 20 2D
F7E8'  20 72 65 66
F7EC'  65 72 20 74
F7F0'  6F 20 75 73
F7F4'  65 72 20 6D
F7F8'  61 6E 75 61
F7FC'  6C 2E 00
F7FF'  0D 0A 0A 49          ERRITR: DB      CR,LF,LF,'Interrupt pending - ',0
F803'  6E 74 65 72
F807'  72 75 70 74
F80B'  20 70 65 6E
F80F'  64 69 6E 67
F813'  20 2D 20 00
F817'  F827'                ITRTAB: DW      ITR0
F819'  F82B'                        DW      ITR1
F81B'  F831'                        DW      ITR2
F81D'  F836'                        DW      ITR3
F81F'  F83F'                        DW      ITR4
F821'  F845'                        DW      ITR5
F823'  F848'                        DW      ITR6
F825'  F84B'                        DW      ITR7
F827'  44 52 51 00          ITR0:   DB      'DRQ',0
F82B'  49 4E 54 52          ITR1:   DB      'INTRQ',0
F82F'  51 00
F831'  50 56 54 43          ITR2:   DB      'PVTC',0
F835'  00
F836'  4B 65 79 62          ITR3:   DB      'Keyboard',0
F83A'  6F 61 72 64
F83E'  00
F83F'  54 69 6D 65          ITR4:   DB      'Timer',0
F843'  72 00
F845'  4A 38 00             ITR5:   DB      'J8',0
F848'  4A 37 00             ITR6:   DB      'J7',0
F84B'  4A 31 00             ITR7:   DB      'J1',0
                            ;
                            ;       OUTMSG    PRINT OUT A MESSAGE
                            ;
                            ;       IN:     IX - ADDRESS OF STRING
                            ;
F84E'                       OUTMSG:
F84E'  DD 7E 00                     LD      A,(IX)          ; GET CHARACTER
F851'  B7                           OR      A       ; CHECK IF NO MORE
F852'  28 07                        JR      Z,OMSGRT
F854'  CD F85C'                     CALL    DATAOUT
F857'  DD 23                        INC     IX      ; POINT TO NEXT CHARACTER
F859'  18 F3                        JR      OUTMSG
F85B'                       OMSGRT:
F85B'  C9                           RET
```

```
                        ;
                        ;       DATA OUTPUT (80 X 24)
                        ;
                        ;       IN:     A - CHARACTER
                        ;
F85C'                   DATAOUT:
F85C'   FE 20                   CP      BLANK           ; CHECK IF CONTROL CHARACTER
F85E'   38 05                   JR      C,CNTLCH        ; YES
F860'   CD F874'                CALL    CHAROUT         ; SEND CHARACTER
F863'   18 0E                   JR      DATORT          ; RETURN
F865'                   CNTLCH:
F865'   FE 0A                   CP      LF      ; CHECK IF LINEFEED
F867'   20 05                   JR      NZ,CHKCR        ; NO, GO TO CHECK FOR CARRIAGE RETURN
F869'   CD F89B'                CALL    LFEED           ; PROCESS LINEFEED
F86C'   18 05                   JR      DATORT          ; RETURN
F86E'                   CHKCR:
F86E'   FE 0D                   CP      CR      ; CHECK IF CARRIAGE RETURN
F870'   CC F8AA'                CALL    Z,CARRTN        ; PROCESS CARRIGAE RETURN
F873'                   DATORT:
F873'   C9                     RET             ; RETURN
                                ;
                                ; CHARACTER OUTPUT TO SCREEN
                                ;
                                ;       A = CHARACTER
                                ;
F874'                   CHAROUT:
F874'   F5                     PUSH    AF
F875'   E5                     PUSH    HL
F876'   D3 1C                  OUT     (VTODAT),A      ; SEND CHARACTER
F878'   3E AB                  LD      A,WRINC         ; WRITE CHARACTER & INCREMENT CURSOR
F87A'   D3 01                  OUT     (VTCCMD),A
F87C'   CD F8C2'                CALL    WAIT
F87F'   3A FFD4'                LD      A,(COLNO)       ; UPDATE COLUMN NUMBER
F882'   3C                     INC     A
F883'   32 FFD4'                LD      (COLNO),A
F886'                   CHORET:
F886'   E1                     POP     HL
F887'   F1                     POP     AF
F888'   C9                     RET             ; RETURN
                                ;
                                ;       GET CURRENT CURSOR POSITION
                                ;
                                ;       OUT:    HL - CURSOR POSITION
                                ;
F889'                   GETCUR:
F889'   F5                     PUSH    AF      ; SAVE REGISTERS
F88A'   DB 04                  IN      A,(VTCCU1)      ; GET CURSOR ADDRESS
F88C'   6F                     LD      L,A
F88D'   DB 05                  IN      A,(VTCCU2)
F88F'   67                     LD      H,A
F890'   F1                     POP     AF      ; RETRIEVE REGISTERS
F891'   C9                     RET
                                ;
                                ;       SET CURSOR POSITION
                                ;
                                ;       IN:     HL - CURSOR POSITION
```

```
                                ;
F892'                           SETCUR:
F892'  F5                       PUSH    AF      ; SAVE REGISTERS
F893'  7D                       LD      A,L
F894'  D3 04                    OUT     (VTCCU1),A
F896'  7C                       LD      A,H
F897'  D3 05                    OUT     (VTCCU2),A
F899'  F1                       POP     AF      ; RETRIEVE REGISTERS
F89A'  C9                       RET
                                ;
                                ; LINE FEED
                                ;
F89B'                           LFEED:
F89B'  D5                       PUSH    DE
F89C'  E5                       PUSH    HL
F89D'                           LFEED1:
F89D'  CD F889'                 CALL    GETCUR          ; GET CURRENT CURSOR POSITION
F8A0'  11 0050                  LD      DE,MAXCOL
F8A3'  19                       ADD     HL,DE
F8A4'  CD F892'                 CALL    SETCUR          ; UPDATE CURSOR POSITION
F8A7'                           LFRET:
F8A7'  E1                       POP     HL
F8A8'  D1                       POP     DE
F8A9'  C9                       RET             ; RETURN
                                ;
                                ; CARRIAGE RETURN
                                ;
F8AA'                           CARRTN:
F8AA'  F5                       PUSH    AF
F8AB'  D5                       PUSH    DE
F8AC'  E5                       PUSH    HL
F8AD'  CD F889'                 CALL    GETCUR          ; GET CURRENT CURSOR POSITION
F8B0'  3A FFD4'                 LD      A,(COLNO)
F8B3'  5F                       LD      E,A
F8B4'  97                       SUB     A
F8B5'  32 FFD4'                 LD      (COLNO),A
F8B8'  57                       LD      D,A
F8B9'  ED 52                    SBC     HL,DE
F8BB'  CD F892'                 CALL    SETCUR          ; SET NEW CURSOR POSITION
F8BE'  E1                       POP     HL
F8BF'  D1                       POP     DE
F8C0'  F1                       POP     AF
F8C1'  C9                       RET             ; RETURN
                                ;
                                ;       WAIT    WAIT FOR PVTC TO BE READY
                                ;
F8C2'                           WAIT:
F8C2'  DB 01                    IN      A,(VTCSTA)      ; GET STATUS REGISTER
F8C4'  E6 20                    AND     VTCRDY          ; CHECK IF READY
F8C6'  28 FA                    JR      Z,WAIT          ; REPEAT IF NOT
F8C8'  C9                       RET
                                ;
                                ;       CP/M BOOT
                                ;
F8C9'                           BOOT:
F8C9'  31 007D                  LD      SP,BUF-3        ; SET UP STACK POINTER
```

```
F8CC'  21 021A                    LD      HL,VTCBEG+410   ; SYNCHRONIZE PVTC
F8CF'  CD F892'                   CALL    SETCUR
F8D2'  DB 18                      IN      A,(STAREG)      ; GET STATUS
F8D4'  CB 47                      BIT     0,A             ; CHECK IF ANY INTERRUPT PENDING
F8D6'  20 2A                      JR      NZ,BOOT4
                                  ;
                                  ;       INTERRUPT LINE ERROR
                                  ;
F8D8'  ED 57                      LD      A,I             ; CHECK IF DISPLAY REQUIRED
F8DA'  CB 7F                      BIT     7,A
F8DC'  28 20                      JR      Z,ITRERR        ; SKIP IF NOT
F8DE'  DD 21 F7FF'                LD      IX,ERRITR       ; DISPLAY INTERRUPT LINE ERROR MESSAGE
F8E2'  CD F84E'                   CALL    OUTMSG
F8E5'  DB 18                      IN      A,(STAREG)      ; GET STATUS
F8E7'  E6 0E                      AND     0EH             ; GET INTERRUPT STATUS
F8E9'  4F                         LD      C,A             ; SET UP BIAS
F8EA'  06 00                      LD      B,0
F8EC'  DD 21 F817'                LD      IX,ITRTAB       ; POINT TO ERROR MESSAGE TABLE
F8F0'  DD 09                      ADD     IX,BC           ; POINT TO PROPER ERROR MESSAGE
F8F2'  DD 4E 00                   LD      C,(IX)          ; GET MESSAGE ADDRESS
F8F5'  DD 46 01                   LD      B,(IX+1)
F8F8'  C5                         PUSH    BC
F8F9'  DD E1                      POP     IX
F8FB'  CD F84E'                   CALL    OUTMSG
F8FE'                          ITRERR:
F8FE'  3E 20                      LD      A,20H
F900'  ED 47                      LD      I,A
                                  ;
                                  ;       TEST DISK CONTROLLER
                                  ;
F902'                          BOOT4:
F902'  3E A5                      LD      A,TSTPAT        ; GET TEST PATTERN
F904'  D3 11                      OUT     (FDTRK),A       ; SEND TO TRACK REGISTER
F906'  D3 12                      OUT     (FDSEC),A
F908'  D3 13                      OUT     (FDDATA),A
F90A'  4F                         LD      C,A             ; SAVE FOR COMPARISON
F90B'  DB 11                      IN      A,(FDTRK)       ; READ BACK FOR COMPARISON
F90D'  B9                         CP      C
F90E'  20 0A                      JR      NZ,DCERR
F910'  DB 12                      IN      A,(FDSEC)
F912'  B9                         CP      C
F913'  20 05                      JR      NZ,DCERR
F915'  DB 13                      IN      A,(FDDATA)
F917'  B9                         CP      C
F918'  28 04                      JR      Z,TSTDON
F91A'                          DCERR:
F91A'  3E 10                      LD      A,10H           ; SET DISK CONTROLLER ERROR
F91C'  ED 47                      LD      I,A
F91E'                          TSTDON:
F91E'  DD 21 F77E'                LD      IX,SYSOK        ; POINT TO SYSTEM OK MESSAGE
F922'  ED 57                      LD      A,I             ; GET ERROR INDICATOR
F924'  E6 3F                      AND     3FH
F926'  B7                         OR      A       ; CHECK IF ANY ERROR
F927'  28 04                      JR      Z,PMSG10
F929'  DD 21 F7A7'                LD      IX,SYSBAD       ; POINT TO SYSTEM BAD MESSAGE
F92D'                          PMSG10:
```

```
F92D'  CD F84E'                CALL    OUTMSG
F930'  ED 57                   LD      A,I             ; GET ERROR INDICATOR
F932'  E6 3F                   AND     3FH
F934'  B7                      OR      A
F935'  C2 F9B7'                JP      NZ,INDBAD       ; INDICATOR ERROR CONDITION
                       ;
                       ;       ACTIVATE BELL TO SIGNAL ALL OK
                       ;
F938'                  BOOT0:
F938'  11 0160                 LD      DE,160H         ; SET LOOP LIMIT
F93B'                  BEEPL0:
F93B'  3E 54                   LD      A,VTMODE        ; GET DEFAULT MODE REGISTER VALUE
F93D'  F6 80                   OR      BEEP            ; ACTIVATE BEEP
F93F'  D3 1D                   OUT     (VTCMOD),A
F941'  06 78                   LD      B,120           ; DELAY LOOP
F943'                  BEEPL1:
F943'  10 FE                   DJNZ    BEEPL1
F945'  3E 54                   LD      A,VTMODE        ; RESET BEEP
F947'  D3 1D                   OUT     (VTCMOD),A
F949'  06 78                   LD      B,120           ; DELAY LOOP
F94B'                  BEEPL2:
F94B'  10 FE                   DJNZ    BEEPL2
F94D'  1B                      DEC     DE      ; REPEAT LOOP
F94E'  7A                      LD      A,D
F94F'  B3                      OR      E
F950'  20 E9                   JR      NZ,BEEPL0
F952'  3E 54                   LD      A,VTMODE        ; SAVE MODE REGISTER FOR FUTURE
F954'  32 FFE0'                LD      (MODREG),A
F957'  3E FB                   LD      A,ITRVEC        ; INITIALIZE INTERRUPT VECTOR
F959'  ED 47                   LD      I,A             ; HIGH BYTE
F95B'  32 FFE1'                LD      (ITRREG),A
                       ;       IM2
F95E'  ED 5E                   DB      0EDH,5EH
F960'  FB                      EI
F961'  AF                      XOR     A
F962'  CB EF                   SET     TIMER,A         ; ENABLE TIMER
F964'  D3 18                   OUT     (CMDREG),A
F966'  11 0000                 LD      DE,0            ; INITIALIZE TIMER COUNT
F969'  ED 53 FFA6'             LD      (TIMCNT),DE
F96D'  3E 32                   LD      A,50            ; DELAY FOR 50 MSEC
F96F'                  TIMLP:
F96F'  CD FD60'                CALL    DELAY
F972'  3D                      DEC     A
F973'  20 FA                   JR      NZ,TIMLP
F975'  ED 5B FFA6'             LD      DE,(TIMCNT)     ; GET TIMER COUNT
F979'  7A                      LD      A,D             ; SHOULD BE NON ZERO
F97A'  B3                      OR      E
F97B'  20 07                   JR      NZ,REBOOT
F97D'  F3                              DI              ; DISABLE FURTHER INTERRUPT
F97E'  3E 08                   LD      A,08H           ; SET ERROR BIT
F980'  ED 47                   LD      I,A
F982'  18 33                   JR      INDBAD
F984'                  REBOOT:
F984'  3E 01                   LD      A,DSKA          ; SELECT DRIVE A
F986'  CD FAA5'                CALL    CHECK
F989'                  BOOT1:
```

```
F989'  B7                        OR      A
F98A'  28 1B                     JR      Z,LOCAL         ; ERROR IF BOOT FAILURE
F98C'  CD FA60'                  CALL    SPEED           ; CHECK DRIVE SPEED
F98F'  B7                        OR      A
F990'  20 15                     JR      NZ,LOCAL        ; BRANCH IF ERROR
F992'                    BOOT2:
F992'  3E 01                     LD      A,1             ; SECTOR 1
F994'  D3 12                     OUT     (FDSEC),A
F996'  06 0A                     LD      B,RETRY         ; SET UP MAXIMUM RETRY
F998'                    BOOT3:
F998'  3E 82                     LD      A,RDCMD         ; SET FOR READ
F99A'  21 0080                   LD      HL,BUF          ; SET DMA
F99D'  CD FC78'                  CALL    FDRWS           ; READ BOOT SECTOR
F9A0'  E6 9E                     AND     RDERR
F9A2'  CA 0080                   JP      Z,BUF           ; JUMP TO BOOT SECTOR IF NO ERROR
F9A5'  10 F1                     DJNZ    BOOT3           ; RETRY
F9A7'                    LOCAL:
F9A7'  3E 00                     LD      A,RSCMD         ; GET RESTORE COMMAND
F9A9'  CD FD06'                  CALL    TYPE1
F9AC'  DD 21 F7D8'               LD      IX,BOOTERR      ; PRINT OUT BOOT ERROR MESSAGE
F9B0'  CD F84E'                  CALL    OUTMSG
F9B3'  3E 10                     LD      A,10H           ; SET ERROR BIT
F9B5'  ED 47                     LD      I,A
F9B7'                    INDBAD:
F9B7'  ED 57                     LD      A,I             ; GET ERROR INDICATOR
F9B9'  11 0200                   LD      DE,0200H        ; SET FOR B BLINK A OFF
F9BC'  CB 47                     BIT     0,A             ; CHECK IF PVTC ERROR
F9BE'  20 28                     JR      NZ,FLASH
F9C0'  11 0301                   LD      DE,0301H        ; SET FOR B BLINK A ON
F9C3'  CB 4F                     BIT     1,A             ; CHECK IF DISPLAY RAM ERROR
F9C5'  20 21                     JR      NZ,FLASH
F9C7'  11 0300                   LD      DE,0300H        ; SET FOR A AND B BLINK
F9CA'  CB 57                     BIT     2,A             ; CHECK IF MEMORY ERROR
F9CC'  20 1A                     JR      NZ,FLASH
F9CE'  11 0303                   LD      DE,0303H        ; SET FOR A AND B ON
F9D1'  CB 5F                     BIT     3,A             ; CHECK IF TIMER ERROR
F9D3'  20 13                     JR      NZ,FLASH
F9D5'  11 0102                   LD      DE,0102H        ; SET FOR A AND B ALTERNATE BLINK
F9D8'  CB 67                     BIT     4,A             ; CHECK IF DISK ERROR
F9DA'  20 0C                     JR      NZ,FLASH
F9DC'  11 0302                   LD      DE,0302H        ; SET FOR A BLINK B ON
F9DF'  CB 6F                     BIT     5,A             ; CHECK IF INTERRUPT LINE ERROR
F9E1'  20 05                     JR      NZ,FLASH
F9E3'  11 0000                   LD      DE,0000H        ; SET FOR A AND B ON
F9E6'  18 00                     JR      FLASH
                         ;
                         ;       MANIPULATION OF LED TO INDICATE ERROR CONDITIONS
                         ;
F9E8'                    FLASH:
F9E8'  7A                        LD      A,D
F9E9'  D3 18                     OUT     (CMDREG),A      ; SET STATE 1
F9EB'  01 0000                   LD      BC,0
F9EE'                    FLASH1:
F9EE'  10 FE                     DJNZ    FLASH1
F9F0'  0D                        DEC     C
F9F1'  20 FB                     JR      NZ,FLASH1       ; DELAY LOOP
```

```
F9F3'  7B                      LD      A,E
F9F4'  D3 18                   OUT     (CMDREG),A      ; SET STATE 2
F9F6'                  FLASH2:
F9F6'  10 FE                   DJNZ    FLASH2
F9F8'  0D                      DEC     C
F9F9'  20 FB                   JR      NZ,FLASH2       ; DELAY LOOP
F9FB'  DB 09                   IN      A,(KEYBST)      ; GET KEYBOARD STATUS
F9FD'  E6 0F                   AND     KEYBMK          ; CHECK IF KEY PRESSED
F9FF'  28 E7                   JR      Z,FLASH         ; REPEAT FLASH IF NOT
FA01'  DB 08                   IN      A,(KEYBDA)      ; GET KEY PRESSED
FA03'  FE 00                   CP      PVTCR0          ; CHECK IF DISPLAY RAM READ TEST
FA05'  28 16                   JR      Z,VDORD0
FA07'  FE 01                   CP      PVTCW0          ; CHECK IF DISPLAY RAM WRITE TEST
FA09'  28 34                   JR      Z,VDOWR0
FA0B'  FE 09                   CP      PVTCR1          ; CHECK IF TEST ON BANK 1
FA0D'  28 1C                   JR      Z,VDORD1
FA0F'  FE 0A                   CP      PVTCW1
FA11'  28 23                   JR      Z,VDOWR1
FA13'  FE 02                   CP      RAMRD           ; CHECK IF MEMORY READ TEST
FA15'  28 38                   JR      Z,MEMRD
FA17'  FE 03                   CP      RAMWR           ; CHECK IF MEMORY WRITE TEST
FA19'  28 3E                   JR      Z,MEMWR
FA1B'  18 CB                   JR      FLASH           ; OTHERWISE REPEAT FLASH
                       ;
                       ;       READ FROM VIDEO RAM CONTINUOUSLY
                       ;
FA1D'                  VDORD0:
FA1D'  3E AC                   LD      A,RDCUR         ; READ AT CURSOR POSITION
FA1F'  D3 01                   OUT     (VTCCMD),A
FA21'                  VDORDL:
FA21'  DB 01                   IN      A,(VTCSTA)
FA23'  E6 20                   AND     VTCRDY
FA25'  28 FA                   JR      Z,VDORDL        ; WAIT UNTIL READY
FA27'  DB 0D                   IN      A,(VTIDAT)      ; READ IN DATA
FA29'  18 F2                   JR      VDORD0          ; REPEAT PROCESS
                       ;
                       ;       READ FROM VIDEO RAM BANK 1
                       ;
FA2B'                  VDORD1:
FA2B'  21 0480                 LD      HL,VTCBEG+400H  ; SET CURSOR TO BANK 1
FA2E'  7D                      LD      A,L
FA2F'  D3 04                   OUT     (VTCCU1),A
FA31'  7C                      LD      A,H
FA32'  D3 05                   OUT     (VTCCU2),A
FA34'  18 E7                   JR      VDORD0
                       ;
                       ;       WRITE TO VIDEO RAM BANK 1
                       ;
FA36'                  VDOWR1:
FA36'  21 0480                 LD      HL,VTCBEG+400H  ; SET CURSOR TO BANK 1
FA39'  7D                      LD      A,L
FA3A'  D3 04                   OUT     (VTCCU1),A
FA3C'  7C                      LD      A,H
FA3D'  D3 05                   OUT     (VTCCU2),A
                       ;
                       ;       WRITE TO VIDEO RAM CONTINOUSLY
```

```
                        ;
FA3F'                   VDOWR0:
FA3F'  3E A5                    LD      A,TSTPAT
FA41'  D3 1C                    OUT     (VTODAT),A      ; WRITE OUT DATA
FA43'  3E AA                    LD      A,WRCUR
FA45'  D3 01                    OUT     (VTCCMD),A
FA47'                   VDOWRL:
FA47'  DB 01                    IN      A,(VTCSTA)
FA49'  E6 20                    AND     VTCRDY
FA4B'  28 FA                    JR      Z,VDOWRL        ; WAIT UNTIL READY
FA4D'  18 F0                    JR      VDOWR0          ; REPEAT PROCESS
                        ;
                        ;       READ FROM MEMORY CONTINOUSLY
                        ;
FA4F'                   MEMRD:
FA4F'  3E A5                    LD      A,TSTPAT
FA51'  32 FFFF                  LD      (MEMEND),A      ; WRITE PATTERN TO MEMORY
FA54'                   MEMRD1:
FA54'  3A FFFF                  LD      A,(MEMEND)
FA57'  18 FB                    JR      MEMRD1          ; READ BACK FOREVER
                        ;
                        ;       WRITE TO MEMORY CONTINOUSLY
                        ;
FA59'                   MEMWR:
FA59'  3E A5                    LD      A,TSTPAT
FA5B'  32 FFFF                  LD      (MEMEND),A
FA5E'  18 F9                    JR      MEMWR           ; WRITE PATTERN TO MEMORY
                        ;
                        ;       DISK DRIVE SPEED TEST
                        ;
FA60'                   SPEED:
FA60'  1E 0C                    LD      E,RSCMD+VERIFY+HDLOAD
FA62'  3A FFD1'                 LD      A,(STEPS)       ; get stepping rate
FA65'  B3                       OR      E
FA66'  CD FD06'                 CALL    TYPE1           ; restore the drive
FA69'  06 05                    LD      B,5             ; SET TO WAIT FIVE TURNS AFTER SYNC
FA6B'  DB 18            STUP0:  IN      A,(STAREG)      ; GET STATUS
FA6D'  E6 40                    AND     40H             ; CHECK INDEX BIT
FA6F'  28 FA                    JR      Z,STUP0         ; LOOP UNTIL INDEX=1
FA71'  DB 18            STUP1:  IN      A,(STAREG)
FA73'  E6 40                    AND     40H
FA75'  20 FA                    JR      NZ,STUP1        ; LOOP UNTIL INDEX=0
FA77'  10 F2                    DJNZ    STUP0
FA79'  21 0001                  LD      HL,1            ; INITIALIZE MEASURED VALUE
FA7C'  DB 18            MEAS0:  IN      A,(STAREG)      ;11 T'S
FA7E'  E6 40                    AND     40H             ;7 T'S
FA80'  28 FA                    JR      Z,MEAS0         ;7/12 T,S
FA82'  DB 18            MEAS1:  IN      A,(STAREG)      ; START COUNTING WHEN LOW TO HIGH
FA84'  E6 40                    AND     40H
FA86'  23                       INC     HL      ;6 T'S
FA87'  20 F9                    JR      NZ,MEAS1
FA89'  DB 18            MEAS2:  IN      A,(STAREG)      ; CONTINUE COUNT UNTIL NEXT LOW TO HIGH
FA8B'  E6 40                    AND     40H             ; TRANSITION
FA8D'  23                       INC     HL      ; INCREMENT COUNT
FA8E'  28 F9                    JR      Z,MEAS2
FA90'  E5                       PUSH    HL      ; SAVE VALUE FOR LATER
```

```
FA91'  11 5511                  LD      DE,SPDLB        ; GET LOWER BOUND
FA94'  AF                       XOR     A
FA95'  ED 52                    SBC     HL,DE
FA97'  E1                       POP     HL      ; RETRIEVE MEASURED VALUE
FA98'  38 08                    JR      C,SPDERR        ; ERROR IF LESS THAN
FA9A'  11 588A                  LD      DE,SPDUB        ; GET UPPER BOUND
FA9D'  AF                       XOR     A
FA9E'  ED 52                    SBC     HL,DE
FAA0'  38 02                    JR      C,SPDRET        ; ERROR IF GREATER THAN
FAA2'                   SPDERR:
FAA2'  3E FF                    LD      A,0FFH          ; SET ERROR INDICATION
FAA4'                   SPDRET:
FAA4'  C9                       RET
                        ;
                        ;*****************************************************************************
                        ;
                        ;       CHECK      TEST CONFIGURATION OF DISKETTE IN DRIVE SPECIFIED
                        ;
                        ;       IN:     A - BIT 0 = 1 -- DRIVE A
                        ;                   BIT 1 = 1 -- DRIVE B
                        ;       OUT:    A = 0 IF ERROR
                        ;                   ELSE = SELECT BYTE
                        ;                          b7 - 6   NOT USED
                        ;                      b5           0 = SSIDE     1 = DSIDE
                        ;                      b4 - 2   NOT USED
                        ;                          b1 - 0   DRIVE NUMBER
                        ;       REG(S) MODIFIED:        A
                        ;
FAA5'                   CHECK:
FAA5'  D5                       PUSH    DE      ; SAVE REGISTERS
FAA6'  E5                       PUSH    HL
FAA7'  E6 03                    AND     DSK             ; MASK OUT UNNECESSARY INFORMATION
FAA9'  CD FC62'                 CALL    DSKSEL          ; SELECT DRIVE
FAAC'                   CHKLP:
FAAC'  3E 0C                    LD      A,RSCMD+HDLOAD+VERIFY
FAAE'  1E 03                    LD      E,SLOSTP
FAB0'  B3                       OR      E
FAB1'  CD FD06'                 CALL    TYPE1           ; RESTORE R/W HEAD
FAB4'  CB 7F                    BIT     EXPTIM,A        ; CHECK IF TIMED OUT
FAB6'  20 F4                    JR      NZ,CHKLP        ; TRY AGAIN
FAB8'  E6 98                    AND     RSERR
FABA'  20 33                    JR      NZ,CHKERR       ; BRANCH IF ERROR
FABC'  3A FFD3'                 LD      A,(SELBYT)      ; OBTAIN SELECT BYTE
FABF'  CB D7                    SET     SIDE1,A         ;SELECT SIDE 1
FAC1'  CD FC62'                 CALL    DSKSEL
FAC4'  21 0080                  LD      HL,BUF          ;READ ADDR INTO BUF
FAC7'  3E C0                    LD      A,RACMD         ;SET UP READ ADDR COMMAND
FAC9'  CD FC78'                 CALL    FDRWS           ;READ ADDR FOR VERIFICATION
FACC'  B7                       OR      A
FACD'  20 20                    JR      NZ,CHKERR       ; BRANCH IF ERROR
FACF'  21 0081                  LD      HL,BUF+1        ; POINT TO SIDE NUMBER
FAD2'  7E                       LD      A,(HL)
FAD3'  B7                       OR      A       ; CHECK IF SIDE 0
FAD4'  3A FFD3'                 LD      A,(SELBYT)
FAD7'  20 04                    JR      NZ,DSIDES       ; DOUBLE SIDED IF NOT
FAD9'  CB AF                    RES     DSIDE,A         ;SINGLE-SIDED
```

```
FADB'  18 04                    JR      CHKSAV
FADD'                   DSIDES: 
FADD'  1E 00                    LD      E,DSSTEP        ; SET STEPPING RATE
FADF'  CB EF                    SET     DSIDE,A         ;DOUBLE-SIDED
FAE1'                   CHKSAV: 
FAE1'  CB 97                    RES     SIDE1,A         ;SELECT SIDE 0
FAE3'  CD FC62'                 CALL    DSKSEL
FAE6'  7B                       LD      A,E
FAE7'  32 FFD1'                 LD      (STEPS),A       ; SAVE STEPPING RATE
FAEA'  3A FFD3'                 LD      A,(SELBYT)
FAED'  18 01                    JR      CHKRET
FAEF'                   CHKERR: 
FAEF'  97                       SUB     A       ; SET ERROR RETURN
FAF0'                   CHKRET: 
FAF0'  E1                       POP     HL      ; RESTORE REGISTERS
FAF1'  D1                       POP     DE
FAF2'  C9                       RET
                        ;
                                ORG     0FB00H
                        ;
                        ;       INTERRUPT VECTORS
                        ;
FB00'  FB96'            DSKDRQ: DW      FDIN            ; DISK DATA REQUEST
FB02'  FBA2'            DSKEND: DW      FDEND           ; DISK COMMAND END
FB04'  FBBA'            PVTCI:  DW      TVREG           ; PVTC INTERRUPT
FB06'  FB10'                    DW      UNIMP           ; KEYBOARD INTERRUPT
FB08'  FB12'            TIMERI: DW      TIMITR          ; PRINTER INTERRUPT
FB0A'  FB10'                    DW      UNIMP
FB0C'  FB10'                    DW      UNIMP
FB0E'  FB10'                    DW      UNIMP
                        ;
                        ;       UN-IMPLEMENTED HANDLER
                        ;
FB10'                   UNIMP:  
FB10'  FB                       EI
FB11'  C9                       RET             ; DUMMY INTERRUPT HANDLER
                        ;
                        ;       TIMER INTERRUPT HANDLER
                        ;
FB12'                   TIMITR: 
FB12'  ED 73 FFA4'              LD      (TIMSTK),SP     ; SAVE CURRENT STACK POINTER
FB16'  31 FFB3'                 LD      SP,STKTIM       ; SET UP LOCAL STACK
FB19'  F5                       PUSH    AF      ; SAVE REGISTERS
FB1A'  E5                       PUSH    HL
FB1B'  D5                       PUSH    DE
FB1C'  3A FFE2'                 LD      A,(IOREG)
FB1F'  CB AF                    RES     TIMER,A         ; RESET TIMER
FB21'  D3 18                    OUT     (CMDREG),A
FB23'  3A FFE2'                 LD      A,(IOREG)
FB26'  D3 18                    OUT     (CMDREG),A
FB28'  FB                       EI              ; RE-ENABLE INTERRUPT
FB29'  2A FFCF'                 LD      HL,(TIMCTR)     ; INCREMENT GENERAL COUNTER
FB2C'  23                       INC     HL
FB2D'  22 FFCF'                 LD      (TIMCTR),HL
FB30'  2A FFA6'                 LD      HL,(TIMCNT)     ; INCREMENT TIMER COUNT
FB33'  23                       INC     HL
```

```
FB34'  22 FFA6'                LD      (TIMCNT),HL
FB37'  3A FFA8'                LD      A,(RPTFLG)      ; CHECK IF REPEAT TEST REQUIRED
FB3A'  B7                      OR      A
FB3B'  28 4E                   JR      Z,TIMIRT        ; RETURN IF NOT
FB3D'  ED 5B FF96'             LD      DE,(DLYCTR)     ; GET DELAY LIMIT
FB41'  AF                      XOR     A
FB42'  ED 52                   SBC     HL,DE           ; CHECK IF TIME REACHED YET
FB44'  38 45                   JR      C,TIMIRT        ; RETURN IF NOT
FB46'  3A FFA9'                LD      A,(TSTFLG)      ; CHECK IF REPEAT TEST IN PROGRESS
FB49'  B7                      OR      A
FB4A'  20 2C                   JR      NZ,POLL
FB4C'  DB 09                   IN      A,(KEYBST)      ; CHECK IF KEY HIT RIGHT ON BOUNDARY
FB4E'  E6 0F                   AND     KEYBMK
FB50'  20 35                   JR      NZ,KEYHIT
FB52'  3A FFE2'                LD      A,(IOREG)       ; GET COMMAND REGISTER CONTENT
FB55'  CB FF                   SET     KBDENA,A        ; DISABLE KEYBOARD SCANNER
FB57'  D3 18                   OUT     (CMDREG),A
FB59'                  WAITLO:
FB59'  DB 18                   IN      A,(STAREG)      ; WAIT UNTIL BIT 7 GO LOW
FB5B'  CB 7F                   BIT     KBDENA,A
FB5D'  20 FA                   JR      NZ,WAITLO
FB5F'  3A FFE2'                LD      A,(IOREG)       ; ENABLE KEYBOARD SCANNER
FB62'  D3 18                   OUT     (CMDREG),A
FB64'  3E FF                   LD      A,0FFH          ; SET REPEAT TEST IN PROGRESS
FB66'  32 FFA9'                LD      (TSTFLG),A
FB69'  11 0002                 LD      DE,2            ; SET DELAY COUNT FOR TWO CYCLES
FB6C'  ED 53 FF96'             LD      (DLYCTR),DE
FB70'  21 0000                 LD      HL,0
FB73'  22 FFA6'                LD      (TIMCNT),HL     ; RE-INITIALIZE TIMER COUNT
FB76'  18 13                   JR      TIMIRT          ; AND RETURN
FB78'                  POLL:
FB78'  AF                      XOR     A       ; RESET TEST IN PROGRESS FLAG
FB79'  32 FFA9'                LD      (TSTFLG),A
FB7C'  DB 09                   IN      A,(KEYBST)      ; CHECK KEYBOARD STATUS
FB7E'  E6 0F                   AND     KEYBMK
FB80'  28 05                   JR      Z,KEYHIT        ; BRANCH IF NOT READY
FB82'  3E FF                   LD      A,0FFH          ; SET AUTO REPEAT MODE
FB84'  32 FFAA'                LD      (AUTFLG),A
FB87'                  KEYHIT:
FB87'  AF                      XOR     A       ; DE-ACTIVATE REPEAT TEST
FB88'  32 FFA8'                LD      (RPTFLG),A
FB8B'                  TIMIRT:
FB8B'  D1                      POP     DE
FB8C'  E1                      POP     HL
FB8D'  F1                      POP     AF
FB8E'  ED 7B FFA4'             LD      SP,(TIMSTK)
FB92'  C3 FB95'                JP      TMIRET          ; JUMP VECTOR FOR 3RD PARTY ROUTINE
FB95'                  TMIRET:
FB95'  C9                      RET
                       ;
                       ;       FLOPPY DISK INPUT HANDLER
                       ;
FB96'                  FDIN:
FB96'  08                      EX      AF,AF'
FB97'  ED A2                   INI
FB99'  08                      EX      AF,AF'
```

```
FB9A'  FB                      EI
FB9B'  C9                      RET
                        ;
                        ;      FLOPPY DISK OUTPUT HANDLER
                        ;
FB9C'                   FDOUT:
FB9C'  08                      EX      AF,AF'
FB9D'  ED A3                   OUTI
FB9F'  08                      EX      AF,AF'
FBA0'  FB                      EI
FBA1'  C9                      RET
                        ;
                        ;      FLOPPY DISK COMMAND COMPLETION HANDLER
                        ;
FBA2'                   FDEND:
FBA2'  08                      EX      AF,AF'          ; SAVE FLAGS
FBA3'  DB 10                   IN      A,(FDSTAT)      ; CLEAR INTERRUPT
FBA5'  3E 01                   LD      A,1
FBA7'  32 FFFF'                LD      (DSKFIN),A      ; SET COMMAND END INDICATOR
FBAA'  08                      EX      AF,AF'          ; RESTORE FLAGS
FBAB'  FB                      EI
FBAC'  C9                      RET
                        ;
                        ;      HARD DISK INPUT/OUTPUT HANDLER
                        ;
FBAD'                   HDIO:
FBAD'  FB                      EI              ; RE-ENABLE INTERRUPT
FBAE'  C9                      RET
                        ;
                        ;      HARD DISK COMMAND COMPLETION HANDLER
                        ;
FBAF'                   HDEND:
FBAF'  08                      EX      AF,AF'          ; SAVE FLAGS
FBB0'  DB 37                   IN      A,(HDSTAT)      ; CLEAR INTERRUPT
FBB2'  3E 01                   LD      A,1
FBB4'  32 FFFF'                LD      (DSKFIN),A      ; SET COMMAND END INDICATOR
FBB7'  08                      EX      AF,AF'          ; RESTORE FLAGS
FBB8'  FB                      EI
FBB9'  C9                      RET
                        ;
                        ;      NORMAL PVTC 40 COLUMN INTERRUPT
                        ;
FBBA'                   TVREG:
FBBA'  ED 73 FF9D'             LD      (STKSAV),SP     ; SAVE STACK POINTER
FBBE'  31 FFBB'                LD      SP,STKBAS
FBC1'  F5                      PUSH    AF
FBC2'  DB 01                   IN      A,(VTCSTA)      ; GET STATUS
FBC4'  E6 18                   AND     INTMSK          ; SET UP FOR RESET
FBC6'  F6 40                   OR      RSTMSK
FBC8'  D3 01                   OUT     (VTCCMD),A
FBCA'  D5                      PUSH    DE      ; SAVE REGISTERS
FBCB'  DD E5                   PUSH    IX
FBCD'  DD 21 FBD3'             LD      IX,TVREGRT      ; SET RETURN ADDRESS
FBD1'  18 23                   JR      TVCOMM          ; GO TO COMMON HANDLER
FBD3'                   TVREGRT:
FBD3'  DD E1                   POP     IX      ; RESTORE REGISTERS
```

```
FBD5'  D1                      POP   DE
FBD6'  F1                      POP   AF
FBD7'  ED 7B FF9D'             LD    SP,(STKSAV)     ; GET ORIGINAL STACK ADDRESS
FBDB'  FB                      EI
FBDC'  C9                      RET
                        ;
                        ;      PVTC 40 COLUMN INTERRUPT WITH DISK I/O
                        ;
FBDD'                   TVDSK:
FBDD'  D9                      EXX           ; GET ALTERNATE REGISTER SET
FBDE'  ED 68                   IN    L,(C)           ; GET STATUS
FBE0'  ED 41                   OUT   (C),B           ; RESET INTERRUPT
FBE2'  FB                      EI
FBE3'  D9                      EXX
FBE4'  D5                      PUSH  DE
FBE5'  F5                      PUSH  AF      ; SAVE REGISTER
FBE6'  DD E5                   PUSH  IX
FBE8'  F3                      DI
FBE9'  D9                      EXX           ; SWITCH TO ALTERNATE REGISTER SET
FBEA'  22 FF9B'                LD    (TVSTAT),HL     ; SAVE STATUS
FBED'  FB                      EI
FBEE'  D9                      EXX
FBEF'  3A FF9B'                LD    A,(TVSTAT)      ; RETRIEVE STATUS
FBF2'  DD 21 FC5D'             LD    IX,TVDSKRT      ; SET RETURN ADDRESS
FBF6'                   TVCOMM:
FBF6'  CB 67                   BIT   VBIT,A          ; CHECK IF VBLANK
FBF8'  28 1F                   JR    Z,NOTVB
FBFA'  ED 5B FFD9'             LD    DE,(OFFSET)     ; GET CURRENT OFFSET
FBFE'  ED 53 FFDB'             LD    (OFFTMP),DE     ; AND SAVE FOR LATER
FC02'  ED 5B FFD7'             LD    DE,(SCSTART)    ; GET CURRENT SCREEN START ADDRESS
FC06'  3A FFD9'                LD    A,(OFFSET)
FC09'  83                      ADD   A,E
FC0A'  D3 02                   OUT   (VTCSC1),A      ; UPDATE SCREEN START REGISTERS
FC0C'  3A FFDA'                LD    A,(OFFSET+1)
FC0F'  8A                      ADC   A,D
FC10'  D3 03                   OUT   (VTCSC2),A
FC12'  3E FF                   LD    A,0FFH          ; SET INDICATOR FOR SCROLLING
FC14'  32 FFDD'                LD    (VBFLAG),A
FC17'  18 42                   JR    TVCMRT
FC19'                   NOTVB:
FC19'  CB 5F                   BIT   LINE0,A         ; CHECK IF LINE ZERO
FC1B'  28 3E                   JR    Z,TVCMRT
FC1D'  11 07FF                 LD    DE,VTCEND       ; GET DISPLAY RAM END ADRESS
FC20'  3A FFDB'                LD    A,(OFFTMP)      ; CALCULATE END ADDRESS FOR COMPARISON
FC23'  83                      ADD   A,E
FC24'  32 FF99'                LD    (ENDAD),A       ; AND SAVE FOR LATER
FC27'  3A FFDC'                LD    A,(OFFTMP+1)
FC2A'  8A                      ADC   A,D
FC2B'  32 FF9A'                LD    (ENDAD+1),A
FC2E'  DB 02                   IN    A,(VTCSC1)      ; COMPUTE NEXT LINE START ADDRESS
FC30'  1E 4F                   LD    E,MAXCOL-1
FC32'  83                      ADD   A,E
FC33'  5F                      LD    E,A
FC34'  DB 03                   IN    A,(VTCSC2)
FC36'  16 00                   LD    D,0
FC38'  8A                      ADC   A,D
```

```
FC39' 57                        LD      D,A
FC3A' 3A FF9A'                  LD      A,(ENDAD+1)     ; COMPARE IF WRAP AROUND
FC3D' 92                        SUB     D
FC3E' 20 14                     JR      NZ,NOTWRA
FC40' 3A FF99'                  LD      A,(ENDAD)
FC43' 93                        SUB     E
FC44' 20 0E                     JR      NZ,NOTWRA
FC46' 11 0080                   LD      DE,VTCBEG       ; HANDLE WRAP AROUND
FC49' 3A FFDB'                  LD      A,(OFFTMP)
FC4C' 83                        ADD     A,E
FC4D' 5F                        LD      E,A
FC4E' 3A FFDC'                  LD      A,(OFFTMP+1)
FC51' 8A                        ADC     A,D
FC52' 57                        LD      D,A
FC53' 1B                        DEC     DE
FC54'                   NOTWRA:
FC54' 13                        INC     DE      ; SET UP NEW SCREEN START REGISTERS
FC55' 7B                        LD      A,E
FC56' D3 02                     OUT     (VTCSC1),A
FC58' 7A                        LD      A,D
FC59' D3 03                     OUT     (VTCSC2),A
FC5B'                   TVCMRT:
FC5B' DD E9                     JP      (IX)            ; RETURN TO PROPER HANDLER
FC5D'                   TVDSKRT:
FC5D' DD E1                     POP     IX      ; RESTORE REGISTERS
FC5F' F1                        POP     AF
FC60' D1                        POP     DE
FC61' C9                        RET
                        ;
                        ;*******************************************************************************
                        ;
                        ;       DSKSEL     SELECT DISK THROUGH I/O REGISTER
                        ;
                        ;       IN:     A - SELECT BYTE
                        ;       OUT:    NONE
                        ;       REG(S) MODIFIED:        A
                        ;
FC62'                   DSKSEL:
FC62' C5                        PUSH    BC      ; SAVE REGISTERS
FC63' 32 FFD3'                  LD      (SELBYT),A      ; SAVE SELECT BYTE
FC66' E6 07                     AND     SELMSK          ; MASK OUT IRRELEVANT BITS
FC68' 47                        LD      B,A
FC69' 3A FFE2'                  LD      A,(IOREG)       ; GET PREVIOUS I/O REGISTER VALUE
FC6C' E6 F8                     AND     IOMSK           ; DIS-SELECT ALL DRIVE
FC6E' D3 18                     OUT     (SEL),A
FC70' B0                        OR      B       ; SET UP NEW SELECT BYTE
FC71' D3 18                     OUT     (SEL),A         ; SELECT NEW DRIVE
FC73' 32 FFE2'                  LD      (IOREG),A       ; SAVE I/O REGISTER VALUE
FC76' C1                        POP     BC      ; RESTORE REGISTER
FC77' C9                        RET
                        ;
                        ;*******************************************************************************
                        ;
                        ;       FDRWS      FLOPPY DISK SECTOR I/O INTERFACE
                        ;
                        ;       IN:     A - COMMAND
```

```
                        ;               HL - DMA ADDRESS
                        ;       OUT:    A - TYPE 2 STATUS
                        ;               HL - NEXT DMA ADDRESS
                        ;       REGISTER(S) MODIFIED:   A,H,L
                        ;
FC78'                   FDRWS:
FC78'  F3               DI              ; DISABLE INTERRUPT
FC79'  ED 73 FFCD'      LD      (DSKSTK),SP     ; SAVE CURRENT STACK
FC7D'  31 FFCD'         LD      SP,STKDSK       ; SET UP NEW STACK
FC80'  D9               EXX             ; SAVE ALTERNATE REGISTER SET
FC81'  C5               PUSH    BC
FC82'  E5               PUSH    HL
FC83'  06 58            LD      B,RSTMSK+INTMSK ; SET UP RESET COMMAND
FC85'  0E 01            LD      C,VTCCMD
FC87'  D9               EXX
FC88'  08               EX      AF,AF'
FC89'  F5               PUSH    AF
FC8A'  3A FFE2'         LD      A,(IOREG)       ; DISABLE TIMER
FC8D'  CB AF            RES     TIMER,A
FC8F'  32 FFE2'         LD      (IOREG),A
FC92'  D3 18            OUT     (CMDREG),A
FC94'  08               EX      AF,AF'
FC95'  C5               PUSH    BC      ; SAVE REGISTERS
FC96'  D5               PUSH    DE
FC97'  01 FB96'         LD      BC,FDIN         ; SET FOR INPUT HANDLER
FC9A'  CB 6F            BIT     WRCMND,A        ; CHECK IF WRITE COMMAND
FC9C'  28 03            JR      Z,SETVEC
FC9E'  01 FB9C'         LD      BC,FDOUT        ; SET FOR OUTPUT HANDLER
FCA1'                   SETVEC:
FCA1'  ED 43 FB00'      LD      (DSKDRQ),BC     ; SET UP VECTOR
FCA5'  01 FBA2'         LD      BC,FDEND
FCA8'  ED 43 FB02'      LD      (DSKEND),BC
FCAC'  01 FBDD'         LD      BC,TVDSK
FCAF'  ED 43 FB04'      LD      (PVTCI),BC
FCB3'  0E 13            LD      C,FDDATA        ; SET UP I/O PORT ADDRESS
FCB5'  47               LD      B,A             ; SAVE COMMAND
FCB6'  AF               XOR     A
FCB7'  32 FFFF'         LD      (DSKFIN),A      ; SET WAITING FOR COMMAND TO COMPLETE
FCBA'  57               LD      D,A             ; SET UP SOFTWARE TIMER
FCBB'  5A               LD      E,D
FCBC'  3E 28            LD      A,40
FCBE'  32 FF98'         LD      (DLYCNT),A
FCC1'  78               LD      A,B             ; RETRIEVE COMMAND
FCC2'  FB               EI              ; MAKE SURE INTERRUPT IS ENABLED
FCC3'  D3 10            OUT     (FDCMD),A       ; SEND COMMAND TO CONTROLLER
FCC5'                   FDRWLP:
FCC5'  3A FFFF'         LD      A,(DSKFIN)      ; GET COMMAND STATUS INDICATOR
FCC8'  1F               RRA             ; WAIT UNTIL COMMAND FINISH
FCC9'  38 1A            JR      C,RWFIN
FCCB'  1D               DEC     E
FCCC'  C2 FCC5'         JP      NZ,FDRWLP
FCCF'  15               DEC     D
FCD0'  C2 FCC5'         JP      NZ,FDRWLP
FCD3'  3A FF98'         LD      A,(DLYCNT)
FCD6'  3D               DEC     A
FCD7'  32 FF98'         LD      (DLYCNT),A
```

```
FCDA'  C2 FCC5'                    JP     NZ,FDRWLP
FCDD'  3E D0                       LD     A,FITRPT        ; CANCEL COMMAND
FCDF'  D3 10                       OUT    (FDCMD),A
FCE1'  3E 80                       LD     A,TIMEXP        ; SET FOR TIMED OUT ERROR
FCE3'  18 02                       JR     FDRWRT
FCE5'                         RWFIN:
FCE5'  DB 10                       IN     A,(FDSTAT)      ; GET STATUS
FCE7'                         FDRWRT:
FCE7'  01 FBBA'                    LD     BC,TVREG        ; RESET TO NORMAL PVTC 40 COL. HANDLING
FCEA'  ED 43 FB04'                 LD     (PVTCI),BC
FCEE'  D1                          POP    DE
FCEF'  C1                          POP    BC      ; RESTORE REGISTERS
FCF0'  08                          EX     AF,AF'
FCF1'  3A FFE2'                    LD     A,(IOREG)       ; RE-ENABLE TIMER
FCF4'  CB EF                       SET    TIMER,A
FCF6'  32 FFE2'                    LD     (IOREG),A
FCF9'  D3 18                       OUT    (CMDREG),A
FCFB'  F1                          POP    AF
FCFC'  08                          EX     AF,AF'
FCFD'  D9                          EXX            ; RESTORE ALTERNATE REGISTER SET
FCFE'  E1                          POP    HL
FCFF'  C1                          POP    BC
FD00'  D9                          EXX
FD01'  ED 7B FFCD'                 LD     SP,(DSKSTK)     ; RETRIEVE PREVIOUS STACK
FD05'  C9                          RET
                              ;
                              ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                              ;
                              ;       TYPE1      HANDLE ALL TYPE I COMMAND FOR FLOPPY CONTROLLER
                              ;
                              ;       IN:     A - TYPE I COMMAND
                              ;       OUT:    A - TYPE I STATUS
                              ;       REG(S) MODIFIED:        A
                              ;
FD06'                         TYPE1:
FD06'  ED 73 FFCD'                 LD     (DSKSTK),SP
FD0A'  31 FFCD'                    LD     SP,STKDSK
FD0D'  08                          EX     AF,AF'
FD0E'  F5                          PUSH   AF
FD0F'  08                          EX     AF,AF'
FD10'  D5                          PUSH   DE      ; SAVE REGISTERS
FD11'  C5                          PUSH   BC
FD12'  F5                          PUSH   AF      ; SAVE COMMAND
FD13'  3E 02                       LD     A,2             ; 2 MS DELAY
FD15'                         TYPE12:
FD15'  CD FD60'                    CALL   DELAY
FD18'  3D                          DEC    A
FD19'  C2 FD15'                    JP     NZ,TYPE12
                              ;
FD1C'  11 FBA2'                    LD     DE,FDEND        ; SET UP INTERRUPT VECTOR
FD1F'  ED 53 FB02'                 LD     (DSKEND),DE
FD23'  57                          LD     D,A             ; SET UP SOFTWARE TIMER
FD24'  5F                          LD     E,A
FD25'  0E 50                       LD     C,80
FD27'  32 FFFF'                    LD     (DSKFIN),A
FD2A'  F1                          POP    AF      ; RESTORE COMMAND
```

```
FD2B'  47                        LD      B,A
FD2C'  AF                        XOR     A       ; CLEAR CARRY FLAG
FD2D'  78                        LD      A,B
FD2E'  FB                        EI              ; MAKE SURE INTERRUPT IS ENABLED
FD2F'  D3 10                     OUT     (FDCMD),A       ; ISSUE TYPE 1 COMMAND
FD31'                    TYPE10:
FD31'  3A FFFF'                  LD      A,(DSKFIN)      ; CHECK COMMAND STATUS INDICATOR
FD34'  1F                        RRA
FD35'  38 14                     JR      C,TYPE14
FD37'  1D                        DEC     E
FD38'  C2 FD31'                  JP      NZ,TYPE10
FD3B'  15                        DEC     D
FD3C'  C2 FD31'                  JP      NZ,TYPE10
FD3F'  0D                        DEC     C
FD40'  C2 FD31'                  JP      NZ,TYPE10
FD43'                    TIMOUT:
FD43'  3E D0                     LD      A,FITRPT        ; CANCEL COMMAND
FD45'  D3 10                     OUT     (FDCMD),A
FD47'  3E 80                     LD      A,TIMEXP        ; SET ERROR
FD49'  18 0B                     JR      TYPE15
                         ;
FD4B'                    TYPE14:
FD4B'  3E 10                     LD      A,16            ;16 MS DELAY
FD4D'                    TYPE13:
FD4D'  CD FD60'                  CALL    DELAY
FD50'  3D                        DEC     A
FD51'  C2 FD4D'                  JP      NZ,TYPE13
                         ;
FD54'  DB 10                     IN      A,(FDSTAT)
FD56'                    TYPE15:
FD56'  C1                        POP     BC      ; RESTORE REGISTERS
FD57'  D1                        POP     DE
FD58'  08                        EX      AF,AF'
FD59'  F1                        POP     AF
FD5A'  08                        EX      AF,AF'
FD5B'  ED 7B FFCD'               LD      SP,(DSKSTK)
FD5F'  C9                        RET
                         ;
                         ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                         ;
                         ;       DELAY     DELAY FOR 1 MINI-SECOND
                         ;
                         ;       IN:     NONE
                         ;       OUT:    NONE
                         ;       REG(S) MODIFIED:        NONE
                         ;
FD60'                    DELAY:
FD60'  C5                        PUSH    BC
FD61'  06 01                     LD      B,1
FD63'                    DL1:
FD63'  0E FF                     LD      C,0FFH
FD65'                    DL0:
FD65'  0D                        DEC     C
FD66'  C2 FD65'                  JP      NZ,DL0
FD69'  05                        DEC     B
FD6A'  C2 FD63'                  JP      NZ,DL1
```

```
FD6D'  C1                      POP     BC
FD6E'  C9                      RET
                       ;
                       ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                       ;
                       ;       HDHOME     HARD DISK RESTORE
                       ;
                       ;       IN:     A - COMMAND
                       ;       OUT:    A - 0 IF NO ERROR
                       ;                 - ERROR REGISTER CONTENT OTHERWISE
                       ;
FD6F'                  HDHOME:
FD6F'  D5                      PUSH    DE      ; SAVE REGISTERS
                       ;       LD      DE,HDEND        ; POINT TO HANDLER
                       ;       LD      (DSKEND),DE     ; PUT INTO VECTOR
                       ;       PUSH    AF      ; SAVE COMMAND
                       ;       XOR     A
                       ;       LD      (DSKFIN),A      ; SET COMMAND STATUS TO WAIT
                       ;       LD      D,A             ; SET UP SOFTWARE TIMER
                       ;       LD      E,A
                       ;       POP     AF      ; RETRIEVE COMMAND
                       ;       EI              ; MAKE SURE INTERRUPT IS ENABLED
                       ;       OUT     HDCMD,A         ; SEND OUT COMMAND
                       ;HDHOM1:
                       ;       LD      A,(DSKFIN)      ; CHECK IF COMMAND FINISH
                       ;       OR      A
                       ;       JR      NZ,HDHOM2
                       ;       DEC     DE      ; DOWN COUNT SOFTWARE TIMER
                       ;       LD      A,D
                       ;       OR      E
                       ;       JR      NZ,HDHOM1
                       ;       LD      A,TIMEXP        ; SET TIMED OUT ERROR
                       ;       JR      HDHOM3
                       ;HDHOM2:
FD70'  D3 37                   OUT     (HDCMD),A               ; SEND OUT COMMAND
FD72'                  HDHOM1:
FD72'  DB 37                   IN      A,(HDSTAT)      ; WAIT UNTIL COMMAND FINISH
FD74'  CB 7F                   BIT     7,A
FD76'  28 FA                   JR      Z,HDHOM1
FD78'                  HDHOM2:
FD78'  DB 37                   IN      A,(HDSTAT)
FD7A'  CB 7F                   BIT     7,A
FD7C'  20 FA                   JR      NZ,HDHOM2
FD7E'  DB 37                   IN      A,(HDSTAT)      ; GET STATUS
FD80'  CB 47                   BIT     HDERRS,A        ; CHECK IF ERROR
FD82'  20 03                   JR      NZ,HDHOM4
FD84'  AF                      XOR     A       ; SET NO ERROR
FD85'  18 02                   JR      HDHOM3
FD87'                  HDHOM4:
FD87'  DB 31                   IN      A,(HDERR)       ; GET ERROR REGISTER CONTENT
FD89'                  HDHOM3:
FD89'  D1                      POP     DE      ; RESTORE REGISTERS
FD8A'  C9                      RET
                       ;
                       ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                       ;
```

```
                        ;       HDRWS     HARD DISK INTERFACE
                        ;
                        ;       IN:     HL - BUFFER
                        ;               A - COMMAND
                        ;       OUT:    A - 0 IF NO ERROR
                        ;                 - ERROR REGISTER CONTENT OTHERWISE
                        ;               HL - NEXT BUFFER ADDRESS
                        ;
FD8B'                   HDRWS:
FD8B'   C5                      PUSH    BC      ; SAVE REGISTERS
FD8C'   D5                      PUSH    DE
FD8D'   F5                      PUSH    AF      ; SAVE COMMAND
FD8E'   1E 02                   LD      E,2             ; SET FOR 512 BYTES
FD90'   DB 36                   IN      A,(HDSDH)       ; GET SELECT BYTE
FD92'   CB 6F                   BIT     5,A
FD94'   20 01                   JR      NZ,FIVE12
FD96'   1D                      DEC     E       ; SET FOR 256 BYTES
FD97'                   FIVE12:
FD97'   F1                      POP     AF      ; RETRIEVE COMMAND
                        ;       LD      DE,HDEND        ; POINT TO HANDLER
                        ;       LD      (DSKEND),DE     ; PUT INTO VECTOR
                        ;       LD      DE,HDIO
                        ;       LD      (DSKDRQ),DE
FD98'   0E 30                   LD      C,HDDATA        ; SET UP DATA REGISTER
                        ;       XOR     A
                        ;       LD      (DSKFIN),A      ; SET COMMAND STATUS TO WAIT
                        ;       LD      D,A             ; SET UP SOFTWARE TIMER
                        ;       LD      E,A
                        ;       LD      A,B             ; RETRIEVE COMMAND
                        ;       EI              ; MAKE SURE INTERRUPT IS ENABLED
                        ;       OUT     HDCMD,A         ; SEND OUT COMMAND
                        ;HDRWLP:
                        ;       LD      A,(DSKFIN)      ; CHECK IF COMMAND FINISH
                        ;       OR      A
                        ;       JR      NZ,HDRWEN
                        ;       DEC     DE      ; DOWN COUNT SOFTWARE TIMER
                        ;       LD      A,D
                        ;       OR      E
                        ;       JR      NZ,HDRWLP
                        ;       LD      A,TIMEXP        ; SET TIMED OUT ERROR
                        ;       JR      HDRWRT
                        ;HDRWEN:
FD9A'   D3 37                   OUT     (HDCMD),A
FD9C'   FE 20                   CP      HDREAD          ; CHECK IF READ
FD9E'   28 07                   JR      Z,HDRD
FDA0'                   HDWRLP:
FDA0'   06 00                   LD      B,0             ; SET UP COUNTER
FDA2'   ED B3                   OTIR            ; SEND DATA TO CONTROLLER
FDA4'   1D                      DEC     E
FDA5'   20 F9                   JR      NZ,HDWRLP
FDA7'                   HDRD:
FDA7'   47                      LD      B,A             ; SAVE COMMAND
FDA8'                   HDRW2:
FDA8'   DB 37                   IN      A,(HDSTAT)
FDAA'   CB 7F                   BIT     7,A
FDAC'   28 FA                   JR      Z,HDRW2
```

```
FDAE'                   HDRW3:
FDAE'  DB 37                    IN      A,(HDSTAT)
FDB0'  CB 7F                    BIT     7,A
FDB2'  20 FA                    JR      NZ,HDRW3
FDB4'  78                       LD      A,B             ; RETRIEVE COMMAND
FDB5'  FE 20                    CP      HDREAD          ; CHECK IF READ
FDB7'  20 07                    JR      NZ,HDNORD
FDB9'                   HDRDLP:
FDB9'  06 00                    LD      B,0             ; SET UP COUNTER
FDBB'  ED B2                    INIR            ; GET DATA FROM CONTROLLER
FDBD'  1D                       DEC     E
FDBE'  20 F9                    JR      NZ,HDRDLP
FDC0'                   HDNORD:
FDC0'  DB 37                    IN      A,(HDSTAT)      ; GET STATUS
FDC2'  CB 47                    BIT     HDERRS,A        ; CHECK IF ERROR
FDC4'  20 03                    JR      NZ,HDRWER
FDC6'  AF                       XOR     A       ; SET NO ERROR
FDC7'  18 02                    JR      HDRWRT
FDC9'                   HDRWER:
FDC9'  DB 31                    IN      A,(HDERR)       ; GET ERROR REGISTER CONTENT
FDCB'                   HDRWRT:
FDCB'  D1                       POP     DE      ; RESTORE REGISTERS
FDCC'  C1                       POP     BC
FDCD'  C9                       RET
                        ;
                        ;*******************************************************************************
                        ;
                        ;       DATAST     CONSOLE STATUS
                        ;
                        ;       IN:     NONE
                        ;       OUT:    NO CARRY IF NOT READY
                        ;               CARRY IF READY AND A=DECODED VALUE
                        ;       REG(S) MODIFIED:        A
                        ;
FDCE'                   DATAST:
FDCE'  D5                       PUSH    DE      ;SAVE REGISTERS
FDCF'  3A FFA9'                 LD      A,(TSTFLG)      ; CHECK IF REPEAT TEST IN PROGRESS
FDD2'  B7                       OR      A
FDD3'  20 4D                    JR      NZ,DATNRD       ; SKIP PROCESSING IF IT IS
FDD5'  DB 09                    IN      A,(KEYBST)      ; CHECK STATUS
FDD7'  E6 0F                    AND     KEYBMK
FDD9'  28 47                    JR      Z,DATNRD        ; RETURN IF NOT READY
FDDB'  3E FF                    LD      A,0FFH          ; ACTIVATE REPEAT KEY TEST
FDDD'  32 FFA8'                 LD      (RPTFLG),A
FDE0'  3A FFAA'                 LD      A,(AUTFLG)      ; CHECK IF IN AUTO REPEAT MODE
FDE3'  B7                       OR      A
FDE4'  20 0B                    JR      NZ,AUTORP       ; AND PROCESS ACCORDINGLY
FDE6'  DB 08                    IN      A,(KEYBDA)      ; GET KEY PRESSED
FDE8'  CD FE25'                 CALL    DECODE
FDEB'  CB 7F                    BIT     7,A             ; CHECK IF FUNCTION KEY
FDED'  20 22                    JR      NZ,FUNKEY
FDEF'  18 24                    JR      DIFKEY
FDF1'                   AUTORP:
FDF1'  AF                       XOR     A
FDF2'  32 FFAA'                 LD      (AUTFLG),A      ; RESET AUTO REPEAT MODE
FDF5'  3A FF94'                 LD      A,(LASTKY)      ; GET LAST KEY PRESSED
```

```
FDF8'  E6 3F                    AND     KEYMSK          ; IGNORE CONTROL AND SHIFT BIT
FDFA'  5F                       LD      E,A             ; SAVE FOR LATER
FDFB'  DB 08                    IN      A,(KEYBDA)      ; GET KEY PRESSED
FDFD'  CD FE25'                 CALL    DECODE
FE00'  CB 7F                    BIT     7,A             ; CHECK IF FUNCTION KEY
FE02'  20 0D                    JR      NZ,FUNKEY
FE04'  3A FF94'                 LD      A,(LASTKY)      ; GET KEY PRESSED
FE07'  E6 3F                    AND     KEYMSK          ; STRIP SHIFT AND CONTROL BITS
FE09'  BB                       CP      E       ; CHECK IF SAME AS PREVIOUS KEY
FE0A'  20 09                    JR      NZ,DIFKEY       ; BRANCH IF NOT
FE0C'  11 0005                  LD      DE,PT05         ; SET AUTO REPEAT TIMER
FE0F'  18 07                    JR      SETDLY
FE11'                   FUNKEY:
FE11'  AF                       XOR     A
FE12'  32 FFA8'                 LD      (RPTFLG),A      ; RESET REPEAT FLAG
FE15'                   DIFKEY:
FE15'  11 0050                  LD      DE,ONESEC       ; SET DELAY FOR ONE SECOND
FE18'                   SETDLY:
FE18'  ED 53 FF96'              LD      (DLYCTR),DE
FE1C'  3A FF95'                 LD      A,(KEYVAL)      ; GET DECODED VALUE
FE1F'  37                       SCF             ; SET READY
FE20'  18 01                    JR      DATIRT
FE22'                   DATNRD:
FE22'  AF                       XOR     A       ; SET NOT READY
FE23'                   DATIRT:
FE23'  D1                       POP     DE
FE24'  C9                       RET
                        ;
                        ;       DECODE KEY ENTERED TO ASCII
                        ;
                        ;       INPUT:  A - KEY VALUE
                        ;       OUTPUT: A - DECODED VALUE
                        ;
FE25'                   DECODE:
FE25'  E5                       PUSH    HL      ; SAVE REGISTERS
FE26'  C5                       PUSH    BC
FE27'  21 0000                  LD      HL,0            ; RE-INITIALIZE TIMER COUNT
FE2A'  22 FFA6'                 LD      (TIMCNT),HL
FE2D'  32 FF94'                 LD      (LASTKY),A      ; SAVE KEY PRESSED
FE30'  21 FE53'                 LD      HL,KEYTBL       ; POINT TO DECODING TABLE
FE33'  4F                       LD      C,A
FE34'  CB 77                    BIT     SHIFT,A         ; CHECK IF SHIFT KEY DEPRESSED
FE36'  28 11                    JR      Z,SHFTEN
FE38'  CB 7F                    BIT     CNTL,A          ; CHECK IF CONTROL KEY DEPRESSED
FE3A'  28 0D                    JR      Z,SHFTEN
FE3C'  DB 18                    IN      A,(CAPREG)      ; CHECK IF CAPITAL LOCK ON
FE3E'  CB 67                    BIT     CAPLCK,A
FE40'  20 07                    JR      NZ,SHFTEN
FE42'  21 FF53'                 LD      HL,CAPTBL       ; POINT TO CAPITAL LOCK TABLE
FE45'  79                       LD      A,C
FE46'  E6 3F                    AND     KEYMSK          ; STRIP CONTROL AND SHIFT BITS
FE48'  4F                       LD      C,A
FE49'                   SHFTEN:
FE49'  06 00                    LD      B,0
FE4B'  09                       ADD     HL,BC
FE4C'  7E                       LD      A,(HL)          ; GET ASCII VALUE
```

```
FE4D'                           DCODRT:
FE4D'  32 FF95'                         LD      (KEYVAL),A      ; SAVE DECODED VALUE
FE50'  C1                               POP     BC      ; RESTORE REGISTERS
FE51'  E1                               POP     HL
FE52'  C9                               RET
                                ;
                                ;       KEYBOARD DECODING TABLE
                                ;
FE53'                           KEYTBL:
                                ;
                                ;       FUNCTION = SHIFT AND CONTROL
                                ;
FE53'  1B 00 B5 1E                      DB      01BH,000H,0B5H,01EH,0B9H,0B0H,060H,07FH         ;0
FE57'  B9 B0 60 7F
FE5B'  B1 B3 B4 B7                      DB      0B1H,0B3H,0B4H,0B7H,0B8H,01FH,03DH,080H         ;8
FE5F'  B8 1F 3D 80
FE63'  09 C5 D2 D5                      DB      009H,0C5H,0D2H,0D5H,0C9H,07BH,07DH,080H         ;10
FE67'  C9 7B 7D 80
FE6B'  D1 D7 D4 D9                      DB      0D1H,0D7H,0D4H,0D9H,0CFH,0D0H,07CH,008H         ;18
FE6F'  CF D0 7C 08
FE73'  80 C1 C6 C7                      DB      080H,0C1H,0C6H,0C7H,0CBH,0CCH,00DH,00AH         ;20
FE77'  CB CC 0D 0A
FE7B'  80 D3 C4 C8                      DB      080H,0D3H,0C4H,0C8H,0CAH,03BH,027H,080H         ;28
FE7F'  CA 3B 27 80
FE83'  80 D8 C3 CE                      DB      080H,0D8H,0C3H,0CEH,0CDH,02FH,093H,080H         ;30
FE87'  CD 2F 93 80
FE8B'  20 DA D6 C2                      DB      020H,0DAH,0D6H,0C2H,02CH,02EH,098H,080H         ;38
FE8F'  2C 2E 98 80
                                ;
                                ;       NO SHIFT AND CONTROL
                                ;
FE93'  1B 32 35 36                      DB      01BH,032H,035H,036H,039H,030H,060H,07FH         ;40
FE97'  39 30 60 7F
FE9B'  31 33 34 37                      DB      031H,033H,034H,037H,038H,02DH,03DH,000H         ;48
FE9F'  38 2D 3D 00
FEA3'  09 05 12 15                      DB      009H,005H,012H,015H,009H,01BH,01DH,000H         ;50
FEA7'  09 1B 1D 00
FEAB'  11 17 14 19                      DB      011H,017H,014H,019H,00FH,010H,01CH,008H         ;58
FEAF'  0F 10 1C 08
FEB3'  00 01 06 07                      DB      000H,001H,006H,007H,00BH,00CH,00DH,00AH         ;60
FEB7'  0B 0C 0D 0A
FEBB'  00 13 04 08                      DB      000H,013H,004H,008H,00AH,03BH,027H,000H         ;68
FEBF'  0A 3B 27 00
FEC3'  00 18 03 0E                      DB      000H,018H,003H,00EH,00DH,02FH,013H,000H         ;70
FEC7'  0D 2F 13 00
FECB'  20 1A 16 02                      DB      020H,01AH,016H,002H,02CH,02EH,018H,000H         ;78
FECF'  2C 2E 18 00
                                ;
                                ;       SHIFT AND NO CONTROL
                                ;
FED3'  1B 40 25 5E                      DB      01BH,040H,025H,05EH,028H,029H,07EH,07FH         ;80
FED7'  28 29 7E 7F
FEDB'  21 23 24 26                      DB      021H,023H,024H,026H,02AH,05FH,02BH,000H         ;88
FEDF'  2A 5F 2B 00
FEE3'  09 45 52 55                      DB      009H,045H,052H,055H,049H,07BH,07DH,000H         ;90
FEE7'  49 7B 7D 00
```

```
FEEB' 51 57 54 59          DB   051H,057H,054H,059H,04FH,050H,07CH,008H   ;98
FEEF' 4F 50 7C 08
FEF3' 00 41 46 47          DB   000H,041H,046H,047H,04BH,04CH,00DH,00AH   ;A0
FEF7' 4B 4C 0D 0A
FEFB' 00 53 44 48          DB   000H,053H,044H,048H,04AH,03AH,022H,000H   ;A8
FEFF' 4A 3A 22 00
FF03' 00 58 43 4E          DB   000H,058H,043H,04EH,04DH,03FH,010H,000H   ;B0
FF07' 4D 3F 10 00
FF0B' 20 5A 56 42          DB   020H,05AH,056H,042H,03CH,03EH,00EH,000H   ;B8
FF0F' 3C 3E 0E 00
                           ;
                           ;    NO SHIFT AND NO CONTROL
                           ;
FF13' 1B 32 35 36          DB   01BH,032H,035H,036H,039H,030H,060H,07FH   ;C0
FF17' 39 30 60 7F
FF1B' 31 33 34 37          DB   031H,033H,034H,037H,038H,02DH,03DH,000H   ;C8
FF1F' 38 2D 3D 00
FF23' 09 65 72 75          DB   009H,065H,072H,075H,069H,05BH,05DH,000H   ;D0
FF27' 69 5B 5D 00
FF2B' 71 77 74 79          DB   071H,077H,074H,079H,06FH,070H,05CH,008H   ;D8
FF2F' 6F 70 5C 08
FF33' 00 61 66 67          DB   000H,061H,066H,067H,06BH,06CH,00DH,00AH   ;E0
FF37' 6B 6C 0D 0A
FF3B' 00 73 64 68          DB   000H,073H,064H,068H,06AH,03BH,027H,000H   ;E8
FF3F' 6A 3B 27 00
FF43' 00 78 63 6E          DB   000H,078H,063H,06EH,06DH,02FH,002H,000H   ;F0
FF47' 6D 2F 02 00
FF4B' 20 7A 76 62          DB   020H,07AH,076H,062H,02CH,02EH,006H,000H   ;F8
FF4F' 2C 2E 06 00
                           ;
                           ;    CAPITAL LOCK DECODING TABLE
                           ;
FF53'                      CAPTBL:
FF53' 1B 32 35 36          DB   01BH,032H,035H,036H,039H,030H,060H,07FH   ;80
FF57' 39 30 60 7F
FF5B' 31 33 34 37          DB   031H,033H,034H,037H,038H,02DH,03DH,000H   ;88
FF5F' 38 2D 3D 00
FF63' 09 45 52 55          DB   009H,045H,052H,055H,049H,05BH,05DH,000H   ;90
FF67' 49 5B 5D 00
FF6B' 51 57 54 59          DB   051H,057H,054H,059H,04FH,050H,05CH,008H   ;98
FF6F' 4F 50 5C 08
FF73' 00 41 46 47          DB   000H,041H,046H,047H,04BH,04CH,00DH,00AH   ;A0
FF77' 4B 4C 0D 0A
FF7B' 00 53 44 48          DB   000H,053H,044H,048H,04AH,03BH,027H,000H   ;A8
FF7F' 4A 3B 27 00
FF83' 00 58 43 4E          DB   000H,058H,043H,04EH,04DH,02FH,002H,000H   ;B0
FF87' 4D 2F 02 00
FF8B' 20 5A 56 42          DB   020H,05AH,056H,042H,02CH,02EH,006H,000H   ;B8
FF8F' 2C 2E 06 00
                           ;
FF93' 00                   CHKSUM: DB   0       ; CHECKSUM
FF94' 00                   LASTKY: DB   0       ; LAST KEY ENTERED
FF95' 00                   KEYVAL: DB   0       ; LAST DECODED VALUE
FF96' 0050                 DLYCTR: DW   ONESEC          ; AUTO REPEAT DELAY COUNT
FF98' 00                   DLYCNT: DB   0       ; DELAY COUNT
FF99' 0000                 ENDAD:  DW   0
```

```
FF9B'  0000              TVSTAT: DW    0
FF9D'  0000              STKSAV: DW    0
FF9F'  0000              MAXTRK: DW    0       ; DISK TEST VARIABLES
FFA1'  0000              TRKNBR: DW    0
FFA3'  00                SECNBR: DB    0
FFA4'  0000              TIMSTK: DW    0       ; STACK SAVE FOR TIMER INTERRUPT
FFA6'  0000              TIMCNT: DW    0       ; TIMER COUNTER
FFA8'  00                RPTFLG: DB    0       ; REPEAT FLAG
FFA9'  00                TSTFLG: DB    0       ; REPEAT IN PROGRESS FLAG
FFAA'  00                AUTFLG: DB    0       ; AUTO REPEAT MODE
FFAB'  0000                      DW    0       ; EXTRA STACK ENTRY FOR SECONDARY ITRPT
FFAD'  0000 0000                 DW    0,0,0
FFB1'  0000
FFB3'                    STKTIM  EQU   $       ; LOCAL STACK FOR TIMER INTERRUPT
FFB3'  0000                      DW    0       ; EXTRA STACK ENTRY FOR SECONDARY ITRPT
FFB5'  0000 0000                 DW    0,0,0
FFB9'  0000
FFBB'                    STKBAS  EQU   $       ; LOCAL STACK FOR 40 COLUMN
FFBB'  0000 0000                 DW    0,0,0,0,0       ; RESERVE STACK FOR DISK I/O
FFBF'  0000 0000
FFC3'  0000
                                 ORG   0FFCDH
FFCD'                    STKDSK:                 ; LOCAL STACK FOR FLOPPY DISK I/O
FFCD'  0000              DSKSTK: DW    0
FFCF'  0000              TIMCTR: DW    0       ; GENERAL TIMER COUNT
                                 ORG   0FFD1H
                         ;
                         ;       BOOT SECTOR CONSTANT
                         ;
                         ;       FFD1    STEPS
                         ;       FFD2    DSKCMD
                         ;       FFD3    SELBYT
                         ;
FFD1'  00                STEPS:  DB    0       ; STEPPING RATE
FFD2'  00                DSKCMD: DB    0       ; DISK COMMAND
FFD3'  00                SELBYT: DB    0       ; SELECT BYTE
                         ;
                         ;       BOOT PROM TO BIOS COMMUNICATION AREA
                         ;
                         ;       FFD4    COLNO
                         ;       FFD6    LINENO
                         ;       FFD7    SCSTART
                         ;       FFD9    OFFSET
                         ;       FFDB    OFFTMP
                         ;       FFDD    VBFLAG
                         ;       FFDE    NOT USED
                         ;       FFDF    SWITCH
                         ;       FFE0    MODREF
                         ;       FFE1    ITRREG
                         ;       FFE2    IOREG
                         ;       FFE3    DTRNTB
                         ;       FFE5    DPBASE
                         ;
FFD4'  0000              COLNO:  DW    0       ; COLUMN NUMBER
FFD6'  00                LINENO: DB    0       ; LINE NUMBER
FFD7'  0000              SCSTART:DW    0       ; SCREEN START ADDRESS
```

```
FFD9'  0000              OFFSET: DW    0       ; OFFSET
FFDB'  0000              OFFTMP: DW    0       ; TEMPORARY STORAGE FOR OFFSET
FFDD'  00                VBFLAG: DB    0       ; VBLANK FLAG
FFDE'  00                        DB    0       ; NOT USED
FFDF'  00                SYSCWD: DB    0       ; SYSTEM CONTROL WORD
FFE0'  00                MODREG: DB    0       ; PVTC MODE REGISTER
FFE1'  00                ITRREG: DB    0       ; I REGISTER
FFE2'  00                IOREG:  DB    0       ; CONTROL REGISTER
FFE3'  0000              DTRNTB: DW    0       ; LOGICAL TO PHYSICAL DISK TRANS. TABLE ADR.
FFE5'  0000              DPBASE: DW    0       ; ADDRESS OF DISK PARM. TABLE
                         ;
                         ;       FFE7  JUMP    CONST
                         ;       FFEA  JUMP    CONIN
                         ;       FFED  JUMP    HDRWS
                         ;       FFF0  JUMP    HDHOME
                         ;       FFF3  JUMP    FDRWS
                         ;       FFF6  JUMP    DSKSEL
                         ;       FFF9  JUMP    TYPE1
                         ;       FFFC  JUMP    DELAY
                         ;
FFE7'  C3 FDCE'          CONST:  JP.     DATAST
FFEA'  00 00 00          CONIN:  DB      0,0,0
FFED'  C3 FD8B'                  JP      HDRWS
FFF0'  C3 FD6F'                  JP      HDHOME
FFF3'  C3 FC78'                  JP      FDRWS
FFF6'  C3 FC62'                  JP      DSKSEL
FFF9'  C3 FD06'                  JP      TYPE1
FFFC'  C3 FD60'                  JP      DELAY
FFFF'  00                DSKFIN: DB      00H             ; DISK COMMAND STATUS INDICATOR
0000'                    STOP:
                                 END
```

Macros:

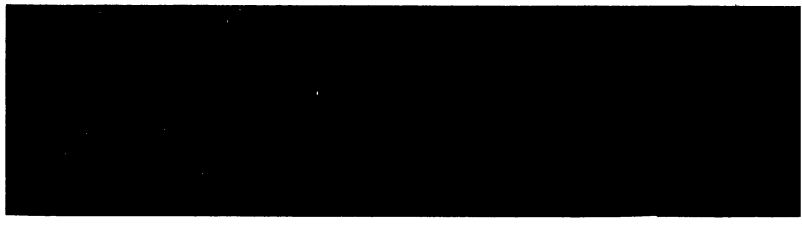
Symbols:

F4A0'	ACTTST	F697'	AFMOVE	FFAA'	AUTFLG
FDF1'	AUTORP	0446'	B2H	0460'	B2HEXE
046E'	B2HLP	0455'	B2HLSN	045E'	B2HRET
0315'	BADSYS	0080	BEEP	F93B'	BEEPL0
F943'	BEEPL1	F94B'	BEEPL2	0020	BLANK
0020	BONW	F8C9'	BOOT	F938'	BOOT0
F989'	BOOT1	F992'	BOOT2	F998'	BOOT3
F902'	BOOT4	F7D8'	BOOTERR	0008	BS
0080	BUF	0005	CAPBIT	0004	CAPLCK
0018	CAPREG	FF53'	CAPTBL	F8AA'	CARRTN
0008	CHAR2K	F874'	CHAROUT	FAA5'	CHECK
01BC'	CHECK3	F5BC'	CHECK4	01FD'	CHECK5
F5FD'	CHECK6	F86E'	CHKCR	FAEF'	CHKERR
FAAC'	CHKLP	03D0'	CHKLP1	F6C7'	CHKLP2
01C2'	CHKLP3	F5C2'	CHKLP4	0203'	CHKLP5
F603'	CHKLP6	FAF0'	CHKRET	FAE1'	CHKSAV
FF93'	CHKSUM	F886'	CHORET	0018	CMDREG
0007	CNTL	0002	CNTLB	F865'	CNTLCH
0001	COL40	FFD4'	COLND	FFEA'	CONIN
FFE7'	CONST	0346'	CONTENT	0072'	CPUEND
0071'	CPUERR	006E'	CPUTST	000D	CR
0030	CUROFF	0031	CURON	F85C'	DATAOUT
FDCE'	DATAST	FE23'	DATIRT	FE22'	DATNRD
F873'	DATORT	F91A'	DCERR	FE4D'	DCODRT
FE25'	DECODE	FD60'	DELAY	FE15'	DIFKEY
0468'	DIGIT	0029	DISPON	FD65'	DL0
FD63'	DL1	FF98'	DLYCNT	FF96'	DLYCTR
0040	DOT7	FFE5'	DPBASE	0005	DSIDE
FADD'	DSIDES	0003	DSK	0001	DSKA
FFD2'	DSKCMD	FB00'	DSKDRQ	FB02'	DSKEND
FFFF'	DSKFIN	FC62'	DSKSEL	FFCD'	DSKSTK
002A	DSKTST	0000	DSSTEP	F4C6'	DTLP1
F518'	DTLP2	F72E'	DTMSG	FFE3'	DTRNTB
F531'	DVFY1	F536'	DVFY2	F509'	DVFY3
F543'	DVFY4	F57F'	DVFY5	F572'	DVFY6
F578'	DVFY7	F58D'	DVFY8	F514'	DVFY9
FF99'	ENDAD	F7FF'	ERRITR	0007	EXPTIM
F7AF'	FAILMSG	0010	FDBASE	0010	FDCMD
0013	FDDATA	FBA2'	FDEND	FB96'	FDIN
FB9C'	FDOUT	FCC5'	FDRWLP	FCE7'	FDRWRT
FC78'	FDRWS	0012	FDSEC	0010	FDSTAT
0011	FDTRK	00D0	FITRPT	FD97'	FIVE12
F9E8'	FLASH	F9EE'	FLASH1	F9F6'	FLASH2
02A1'	FLHL10	029D'	FLHLP	02A4'	FLHLP1
02AC'	FLHLP2	FE11'	FUNKEY	F889'	GETCUR
0037	HDCMD	0032	HDCNT	0035	HDCYLH
0034	HDCYLL	0030	HDDATA	FBAF'	HDEND
0031	HDERR	0000	HDERRS	FD72'	HDHOM1
FD78'	HDHOM2	FD89'	HDHOM3	FD87'	HDHOM4
FD6F'	HDHOME	FBAD'	HDIO	0008	HDLOAD
FDC0'	HDNORD	FDA7'	HDRD	FDB9'	HDRDLP
0020	HDREAD	FDA8'	HDRW2	FDAE'	HDRW3

FDC9'	HDRWER	FDCB'	HDRWRT	FD8B'	HDRWS
0036	HDSDH	0033	HDSEC	0037	HDSTAT
F72B'	HDTEST	0031	HDWPC	FDA0'	HDWRLP
0020	HLD	F9B7'	INDBAD	A2ED	INICOD
002E'	INILP1	0003'	INITAB	0004	INTENS
0010	INTERL	0018	INTMSK	00F8	IOMSK
FFE2'	IOREG	0044	IR0	0032	IR1
0080	IR10	000E	IR2	0066	IR3
0017	IR4	004F	IR5	0008	IR6
0009	IR7	0080	IR8	0010	IR9
0028'	ITABEN	F827'	ITR0	F82B'	ITR1
F831'	ITR2	F836'	ITR3	F83F'	ITR4
F845'	ITR5	F848'	ITR6	F84B'	ITR7
F8FE'	ITRERR	FFE1'	ITRREG	F817'	ITRTAB
00FB	ITRVEC	0007	KBDENA	0000	KBMOD1
002A	KBMOD2	0009	KEYBCM	0008	KEYBD
0008	KEYBDA	000F	KEYBMK	0009	KEYBST
FB87'	KEYHIT	003F	KEYMSK	0180'	KEYRDY
FE53'	KEYTBL	FF95'	KEYVAL	04A0'	LAST
FF94'	LASTKY	03BE'	LDLP1	F6B5'	LDLP2
01AC'	LDLP3	F5AC'	LDLP4	01ED'	LDLP5
F5ED'	LDLP6	000A	LF	F89B'	LFEED
F89D'	LFEED1	F8A7'	LFRET	0003	LINE0
FFD6'	LINEN0	0008	LINZER	F9A7'	LOCAL
8000	M1BEG	03E9'	M1ERR	03FE'	M1ERR1
0405'	M1ERR3	040C'	M1ERR4	0417'	M1ERR5
041E'	M1ERR6	0429'	M1ERR7	0430'	M1ERR8
043B'	M1ERR9	032A'	M1HDG	0476'	M1OK
8000	M1SIZE	0000	M2BEG	F6DF'	M2ERR
F6F5'	M2ERR1	F6FD'	M2ERR3	F705'	M2ERR4
F710'	M2ERR5	F718'	M2ERR6	F723'	M2ERR7
F72B'	M2OK	F000	M2SIZE	FFFF	M3BEG
0286'	M3ER10	022C'	M3ERR	023D'	M3ERR1
0245'	M3ERR3	024D'	M3ERR4	0258'	M3ERR5
0260'	M3ERR6	026B'	M3ERR7	0273'	M3ERR8
027E'	M3ERR9	02B8'	M3OK	EFFF	M4BEG
F680'	M4ER10	F625'	M4ERR	F636'	M4ERR1
F63E'	M4ERR3	F646'	M4ERR4	F651'	M4ERR5
F659'	M4ERR6	F664'	M4ERR7	F66C'	M4ERR8
F677'	M4ERR9	0050	MAXCOL	0017	MAXLIN
FF9F'	MAXTRK	FA7C'	MEAS0	FA82'	MEAS1
FA89'	MEAS2	FFFF	MEMEND	FA4F'	MEMRD
FA54'	MEMRD1	0034	MEMTST	FA59'	MEMWR
FFE0'	MODREG	0476'	MOVE	F6AD'	MTEST2
0384'	NODISP	FC19'	NOTVB	FC54'	NOTWRA
01DC'	NXADR3	F5DC'	NXADR4	021D'	NXADR5
F61D'	NXADR6	03B8'	NXPAT1	F6AF'	NXPAT2
01A6'	NXPAT3	F5A6'	NXPAT4	01E7'	NXPAT5
F5E7'	NXPAT6	01CD'	NXTAD3	F5CD'	NXTAD4
020E'	NXTAD5	F60E'	NXTAD6	F694'	NXTBWD
029A'	NXTFWD	0055'	NXTTST	0028	OFFDIS
FFD9'	OFFSET	FFDB'	OFFTMP	F85B'	OMSGRT
003D	ONDICU	0050	ONESEC	A3ED	OUTICD
F84E'	OUTMSG	0354'	PATTERN	02E2'	PMSG
F92D'	PMSG10	02F0'	PMSGL1	02FC'	PMSGL2
0306'	PMSGRT	FB78'	POLL	F000	PROMAD

0360'	PROMPT	0000	PROMST	0005	PT05
0096'	PVTC	009D'	PVTC1	00C3'	PVTC2
00CE'	PVTCER	FB04'	PVTCI	0000	PVTCR0
0009	PVTCR1	0001	PVTCW0	000A	PVTCW1
00C0	RACMD	0002	RAMRD	0003	RAMWR
0082	RDCMD	00AC	RDCUR	009E	RDERR
00AD	RDINC	F984'	REBOOT	00C9	RETCOD
000A	RETRY	0002	REVVID	0010	RNF
0000'	ROMBEG	F4A0	ROMSAD	1000	ROMSIZ
FFA8'	RPTFLG	0000	RSCMD	0098	RSERR
0048	RSTLN0	0040	RSTMSK	0050	RSTVBL
FCE5'	RWFIN	0000	SCRLOF	0001	SCRLON
FFD7'	SCSTART	FFA3'	SECNBR	0018	SEL
FFD3'	SELBYT	0007	SELMSK	F892'	SETCUR
FE18'	SETDLY	FCA1'	SETVEC	FE49'	SHFTEN
0006	SHIFT	037E'	SHOW	0050	SICMD
0002	SIDE1	0308'	SIGNON	0010	SKCMD
0010	SKERR	0003	SLOSTP	FAA2'	SPDERR
5511	SPDLB	FAA4'	SPDRET	588A	SPDUB
FA60'	SPEED	0004	SPLITS	0018	STAREG
FFD1'	STEPS	FFBB'	STKBAS	FFCD'	STKDSK
FF9D'	STKSAV	FFB3'	STKTIM	0000'	STOP
FA6B'	STUP0	FA71'	STUP1	F7A7'	SYSBAD
FFDF'	SYSCWD	F77E'	SYSOK	F5A1'	TESTM
0196'	TESTM1	FFA6'	TIMCNT	FFCF'	TIMCTR
0005	TIMER	FB08'	TIMERI	0080	TIMEXP
FB8B'	TIMIRT	FB12'	TIMITR	F96F'	TIMLP
FD43'	TIMOUT	FFA4'	TIMSTK	FB95'	TMIRET
0319'	TMMSG	FFA1'	TRKNBR	F91E'	TSTDON
FFA9'	TSTFLG	008A'	TSTLP1	01A1'	TSTMEM
00A5	TSTPAT	FC5B'	TVCMRT	FBF6'	TVCOMM
FBDD'	TVDSK	FC5D'	TVDSKRT	FBBA'	TVREG
FBD3'	TVREGRT	FF9B'	TVSTAT	FD06'	TYPE1
FD31'	TYPE10	FD15'	TYPE12	FD4D'	TYPE13
FD4B'	TYPE14	FD56'	TYPE15	FB10'	UNIMP
FFDD'	VBFLAG	0004	VBIT	0010	VBLANK
FA1D'	VDORD0	FA2B'	VDORD1	FA21'	VDORDL
FA3F'	VDOWR0	FA36'	VDOWR1	FA47'	VDOWRL
0004	VERIFY	8000	VFYBUF	F597'	VFYERR
0000	VTCBAS	0080	VTCBEG	0001	VTCCMD
0004	VTCCU1	0005	VTCCU2	07FF	VTCEND
0110'	VTCERR	0122'	VTCERW	0000	VTCINI
001D	VTCMOD	0132'	VTCOK	0006	VTCPT1
0007	VTCPT2	0020	VTCRDY	0000	VTCRST
0002	VTCSC1	0003	VTCSC2	0001	VTCSTA
012B'	VTCWAIT	000D	VTIDAT	0054	VTMODE
001C	VTODAT	00E0'	VTSTLP	F8C2'	WAIT
00D7'	WAITL0	00E9'	WAITL1	00FA'	WAITL2
0103'	WAITL3	0155'	WAITL4	0164'	WAITL5
016F'	WAITL6	03A2'	WAITL7	0172'	WAITL8
03B6'	WAITL9	FB59'	WAITLO	0030	WHDBAS
00A2	WRCMD	0005	WRCMND	00BB	WRCP
00AA	WRCUR	00AB	WRINC		

No Fatal error(s)



E. Boot Sector Listing

```
                        .Z80

                        ;
                        ;       BOOT SECTOR FOR PIED PIPER
                        ;
                        ;       SEMI-TECH MICROELECTRONICS CORP. COPYRIGHT
                        ;
003A                    MSIZE   EQU     58
                        ;BIAS   EQU     (MSIZE-20)*1024
9800                    BIAS    EQU     09800H
CC00                    CCP     EQU     3400H+BIAS
D406                    BDOS    EQU     CCP+806H
E200                    BIOS    EQU     CCP+1600H
                        ;
                        ;       COMMON AREA FOR PARAMETERS AND JUMP TABLES IN PROM
                        ;
FFDB                    PBASE   EQU     0FFDBH
FFCE                    LOCAL   EQU     0FFCEH          ; DISK ERROR LOCAL MODE
FFD1                    STEPS   EQU     0FFD1H          ; STEPPING RATE OF BOOT DRIVE
FFD2                    FDCMD   EQU     0FFD2H          ; FLOPPY DISK COMMAND
FFD3                    SELBYT  EQU     0FFD3H          ; BOOT DISK SELECT BYTE INFORMATION
FFE1                    ITRREG  EQU     0FFE1H
FFE2                    IOREG   EQU     0FFE2H          ; I/O REGISTER
FFF3                    FDRWS   EQU     PBASE+18H       ; FLOPPY DISK SECTOR READ/WRITE ROUTINE
FFF6                    DSKSEL  EQU     PBASE+1BH       ; DISK SELECT ROUTINE
FFF9                    TYPE1   EQU     PBASE+1EH       ; FLOPPY DISK TYPE I COMMAND ROUTINE
FFFC                    DELAY   EQU     PBASE+21H       ; 1 MS DELAY ROUTINE
                        ;
0082                    RDCMD   EQU     10000010B       ; FLOPPY DISK READ SECTOR COMMAND
0008                    SBIT    EQU     00001000B       ; FLOPPY DISK SIDE SELECT BIT
0058                    SICMD   EQU     01011000B       ; FLOPPY DISK STEP IN COMMAND
0002                    SIDE1   EQU     2       ; SIDE SELECT BIT
0005                    SIDE    EQU     5       ; SIDE INFORMATION BIT
0080                    BOOT    EQU     80H             ; ORIGIN OF BOOTSTRAP
0010                    SIDLY   EQU     16      ; 16 MS DELAY FOR STEP-IN COMMAND
000A                    RETRY   EQU     10
0002                    TRK0ST  EQU     2       ; START SECTOR FOR TRACK 0
000A                    TRK0EN  EQU     10      ; END SECTOR FOR TRACK 0
0001                    TRK1ST  EQU     1       ; START SECTOR FOR TRACK 1
000A                    TRK1EN  EQU     10      ; END SECTOR FOR TRACK 1
0012                    SEC     EQU     12H             ; SECTOR REGISTER
                        ;
                                .PHASE  BOOT
0080    0E 02           BOOTS:  LD      C,TRK0ST        ; START FROM SECTOR 2
0082    1E 0A                   LD      E,TRK0EN        ; UP TO SECTOR 10
0084    21 CC00                 LD      HL,CCP          ; SET UP DMA ADDRESS
0087    3E 82                   LD      A,RDCMD         ; SET FOR READ
0089    32 FFD2                 LD      (FDCMD),A
008C    CD 00DF                 CALL    RDSECS          ; READ IN TRACK 0 SECTORS
008F    B7                      OR      A
0090    20 32                   JR      NZ,BOOTERR
0092    3A FFD3                 LD      A,(SELBYT)      ; GET SELECT BYTE
0095    CB 6F                   BIT     SIDE,A          ; CHECK IF DOUBLE SIDED
0097    20 13                   JR      NZ,DSIDE
0099    3A FFD1                 LD      A,(STEPS)       ; GET STEPPING RATE OF BOOT DISK
```

```
009C  5F                    LD      E,A
009D  3E 58                 LD      A,SICMD         ; PERFORM STEP IN IF SINGLE SIDED
009F  B3                    OR      E
00A0  CD FFF9               CALL    TYPE1
00A3  06 10                 LD      B,SIDLY         ; SET TO DELAY 16 MS
00A5                  DLYLP:
00A5  CD FFFC               CALL    DELAY
00A8  10 FB                 DJNZ    DLYLP
00AA  18 0D                 JR      RDTRK1
00AC                  DSIDE:
00AC  CB D7                 SET     SIDE1,A         ; SELECT SIDE 1
00AE  CD FFF6               CALL    DSKSEL
00B1  3A FFD2               LD      A,(FDCMD)       ; MODIFY READ COMMAND
00B4  F6 08                 OR      SBIT
00B6  32 FFD2               LD      (FDCMD),A
00B9                  RDTRK1:
00B9  0E 01                 LD      C,TRK1ST        ; START FROM SECTOR 1
00BB  1E 0A                 LD      E,TRK1EN        ; END AT SECTOR 5
00BD  CD 00DF               CALL    RDSECS          ; READ IN SECTORS ON TRACK 1
00C0  B7                    OR      A
00C1  CA 00C8               JP      Z,BIOS1         ; START EXECUTION OF BIOS
00C4                  BOOTERR:

                      ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx;
                      ;       THE FOLLOWING INSTRUCTION MUST      ;
                      ;       BE ADDED:                   ;
                      ;                                   ;
                      ;       DI                          ;
                      ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx;

00C4  F3                    DI
00C5  76                    HALT            ; ERROR, GO TO LOCAL MODE
00C6  0000                  DW      0

00C8                  bios1:
00C8  3E 10                 ld      a,10h
00CA  D3 01                 out     (1),a
00CC  3E 44                 ld      a,44H
00CE  D3 00                 out     (0),a
00D0  3E 26                 ld      a,26h
00D2  D3 00                 out     (0),a
00D4  3E 25                 ld      a,25h
00D6  D3 00                 out     (0),a
00D8  3E 6C                 Ld      a,6ch
00DA  D3 00                 aUt     (0),a
00DC  C3 E200               jp      bios
                      ;
                      ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                      ;
                      ;       RDSECS     MULTIPLY SECTOR READ
                      ;
                      ;       IN:     C - STARTING SECTOR NUMBER
                      ;               E - ENDING SECTOR NUMBER
                      ;               HL - STARTING DMA ADDRESS
                      ;       OUT:    A - TYPE II STATUS
                      ;       REG(S) MODIFIED:        A,B,C,H,L
```

```
                        ;
00DF                    RDSECS:
00DF  06 0A                     LD      B,RETRY         ; SET FOR RETYR 10 TIMES
00E1                    FDRTRY:
00E1  E5                        PUSH    HL      ; SAVE CURRENT DMA
00E2  79                        LD      A,C
00E3  D3 12                     OUT     (SEC),A
00E5  3A FFD2                   LD      A,(FDCMD)       ; GET COMMAND
00E8  CD FFF3                   CALL    FDRWS           ; READ A SECTOR
00EB  B7                        OR      A
00EC  28 05                     JR      Z,NXTSEC        ;READ NEXT SECTOR IF NO ERROR
00EE  E1                        POP     HL      ;RESTORE OLD DMA
00EF  10 F0                     DJNZ    FDRTRY          ;RETRY IF ERROR WITHIN 10 TRIALS
00F1  18 07                     JR      RDSRET
00F3                    NXTSEC:
00F3  F1                        POP     AF      ; SYNCHRONIZE PUSH POP
00F4  0C                        INC     C       ; NEXT SECTOR
00F5  7B                        LD      A,E             ; GET MAXIMUM NUMBER OF SECTOR
00F6  B9                        CP      C
00F7  30 E8                     JR      NC,FDRTRY       ;READ NEXT SECTOR IF NOT GREATER
                                        ;THAN SECTOR SIZE
00F9  97                        SUB     A       ; SET NO ERROR
00FA                    RDSRET:
00FA  C9                        RET
                                END
```

Macros:

Symbols:

D406	BDOS	9800	BIAS	E200	BIOS
00C8	BIOS1	0080	BOOT	00C4	BOOTERR
0080	BOOTS	CC00	CCP	FFFC	DELAY
00A5	DLYLP	00AC	DSIDE	FFF6	DSKSEL
FFD2	FDCMD	00E1	FDRTRY	FFF3	FDRWS
FFE2	IOREG	FFE1	ITRREG	FFCE	LOCAL
003A	MSIZE	00F3	NXTSEC	FFDB	PBASE
0082	RDCMD	00DF	RDSECS	00FA	RDSRET
00B9	RDTRK1	000A	RETRY	0008	SBIT
0012	SEC	FFD3	SELBYT	0058	SICMD
0005	SIDE	0002	SIDE1	0010	SIDLY
FFD1	STEPS	000A	TRK0EN	0002	TRK0ST
000A	TRK1EN	0001	TRK1ST	FFF9	TYPE1

No Fatal error(s)

F. BIOS Listing

```
                        .Z80

                        ;
                        ;       BIOS FOR CP/M 2.2       ( PIED PIPER VERSION 1.01 )
                        ;
                        ;          SEMI-TECH MICROELECTRONICS CORP.
                        ;

003A                    MSIZE   EQU     58      ; CPM MEMORY SIZE IN KILOBYTES

                        ;
                        ;       "BIAS" IS ADDRESS OFFSET FROM 3400H FOR MEMORY SYSTEMS
                        ;       TH@N 16K
                        ;

000A                    PADDING EQU     10              ; PADDING
000C                    FMFEED  EQU     0CH             ; FORMFEED
9800                    BIAS    EQU     09800H          ; (MSIZE-20)*1024
CC00                    CCP     EQU     3400H+BIAS      ; BASE OF CCP
D406                    BDOS    EQU     CCP+806H        ; BASE OF BDOS
E200                    BIOS    EQU     CCP+1600H       ; BASE OF BIOS
0004                    CDISK   EQU     0004H           ; CURRENT DISK NUMBER 0=A,...,15=P
0003                    IOBYTE  EQU     0003H           ; INTEL I/O BYTE
0000                    WARMST  EQU     0000H           ; WARM START VECTOR
0001                    REBOOT  EQU     0001H
0005                    SYSCAL  EQU     0005H           ; BDOS FUNCTION CALL VECTOR
0006                    BDOSAD  EQU     0006H

                        ;
                        ;       COMMON AREA FOR UTILITIES
                        ;

FFD4                    COLNO   EQU     0FFD4H          ; COLUMN NUMBER
FFD6                    LINENO  EQU     0FFD6H          ; LINE NUMBER
FFD7                    SCSTART EQU     0FFD7H          ; SCREEN START ADDRESS
FFD9                    OFFSET  EQU     0FFD9H          ; SCREEN OFFSET
FFDB                    OFFTMP  EQU     0FFDBH          ; SCREEN TEMPORARY OFFSET
FFDD                    VBFLAG  EQU     0FFDDH          ; VBLANK FLAG
FFDF                    SWITCH  EQU     0FFDFH          ; SYSTEM SWITCH
FFE0                    MODREG  EQU     0FFE0H          ; MODE REGISTER
FFE1                    IREG    EQU     0FFE1H          ; I REGISTER
FFE2                    IOREG   EQU     0FFE2H          ; I/O CONTROL REGISTER
FFE3                    DMAPTP  EQU     0FFE3H          ; DISK NUMBER TRANSLATION TABLE
FFE5                    DPBTP   EQU     0FFE5H          ; DISK PARAMETER BASE TABLE

                        ;
                        ;       COMMON AREA FOR SUBROUTINES TABLES IN PROM
                        ;

FFE7                    PBASE   EQU     0FFE7H
FFE7                    CONSTA  EQU     PBASE+00H       ; CONSOLE STATUS ROUTINE
FFEA                    CONINP  EQU     PBASE+03H       ; CONSOLE INPUT ROUTINE
FFED                    HDRW    EQU     PBASE+06H       ; HARD DISK READ/WRITE SECTOR ROUTINE
FFF0                    HDHCMD  EQU     PBASE+09H       ; HARD DISK COMMAND ROUTINE
FFF3                    FDRW    EQU     PBASE+0CH       ; FLOPPY DISK READ/WRITE SECTOR ROUTINE
FFF6                    FDSEL   EQU     PBASE+0FH       ; SELECT FLOPPY DISK ROUTINE
```

```
FFF9                    FDHCMD  EQU     PBASE+12H       ; FLOPPY DISK TYPE 1 COMMAND ROUTINE
FFFC                    DELAY   EQU     PBASE+15H       ; DELAY 1 MS ROUTINE

                                ;
                                ;       GENERAL EQUATES
                                ;

000D                    PTRDAT  EQU     0DH             ; PARALLEL PRINTER DATA REGISTER
0008                    KEYBD   EQU     08H             ; KEYBOARD REGISTER
0018                    STATUS  EQU     18H             ; STATUS REGISTER
0018                    CONTRL  EQU     18H             ; CONTROL REGISTER
0003                    MAXDSK  EQU     3       ; MAXIMUM NUMBER OF DISK DRIVES SUPPORTED
00C3                    JMPCOD  EQU     0C3H            ; JUMP INSTRUCTION CODE
0080                    RECLEN  EQU     80H             ; LOGICAL RECORD LENGTH
0000                    DISKTY  EQU     0       ; DISK TYPE BIT  0 = FLOPPY DISK
                                ;                               1 = HARD DISK
00FF                    ON      EQU     0FFH            ; FLAG ON
0000                    OFF     EQU     0       ; FLAG OFF
0000                    READ    EQU     0       ; READ OPERATION
0001                    WRITE   EQU     1       ; WRITE OPERATION
000D                    CR      EQU     0DH             ; CARRIAGE RETURN
000A                    LF      EQU     0AH             ; LINEFEED
0000                    EOL     EQU     00H             ; END OF LINE INDICATOR
0000                    WRALL   EQU     0       ; WRITE TO ALLOCATED
0001                    WRDIR   EQU     1       ; WRITE TO DIRECTORY
0002                    WRULA   EQU     2       ; WRITE TO UNALLOCATED
0002                    SEKRTY  EQU     2       ; SEEK RETRY
000A                    RETRY   EQU     10      ; READ/WRITE SECTOR RETRY
FFFF                    TIMING  EQU     0FFFFH          ; TIMEOUT VALUE FOR LIST
000A                    TVCNT   EQU     10      ; TV COUNTER

                                ;
                                ;       FLOPPY DISK PARAMETERS
                                ;

0010                    FDBASE  EQU     10H             ; FLOPPY CONTROLLER BASE ADDRESS
0010                    FDCMD   EQU     FDBASE+0        ; FLOPPY CONTROLLER COMMAND REGISTER
0010                    FDSTAT  EQU     FDBASE+0        ; FLOPPY CONTROLLER STATUS REGISTER
0011                    FDTRK   EQU     FDBASE+1        ; FLOPPY CONTROLLER TRACK REGISTER
0012                    FDSEC   EQU     FDBASE+2        ; FLOPPY CONTROLLER SECTOR REGISTER
0013                    FDDATA  EQU     FDBASE+3        ; FLOPPY CONTROLLER DATA REGISTER

000C                    HOMEFD  EQU     00001100B       ; FLOPPY DISK HOME COMMAND
001C                    SEEKFD  EQU     00011100B       ; FLOPPY DISK SEEK COMMAND
005C                    STEPFD  EQU     01011100B       ; FLOPPY DISK STEP IN COMMAND
0082                    READFD  EQU     10000010B       ; FLOPPY DISK READ SECTOR COMMAND
00A2                    WRITFD  EQU     10100010B       ; FLOPPY DISK WRRITE SECTOR COMMAND

0003                    SIDE    EQU     3       ; FLOPPY DISK COMMAND SIDE BIT
0007                    FDERR   EQU     7       ; FLOPPY DISK ERROR
0006                    WRPROT  EQU     6       ; WRITE PROTECTED
0090                    SKERR   EQU     10010000B       ; SEEK ERROR
0004                    RNFERR  EQU     4

0002                    FDISK1  EQU     00000010B       ; SSDD, 40 TRACKS
0004                    FDISK2  EQU     00000100B       ; DSDD, 40 TRACKS
```

```
0008                    FDISK3  EQU     00001000B       ; DSDD, 80 TRACKS

                                ;
                                ;       WARM BOOT PARAMETERS
                                ;

0002                    TRK0ST  EQU     2       ; WARM BOOT START SECTOR ON TRACK 0
000A                    TRK0EN  EQU     10      ; WARM BOOT END SECTOR ON TRACK 0
0001                    TRK1ST  EQU     1       ; WARM BOOT START SECTOR ON TRACK 1
0002                    TRK1EN  EQU     2       ; WARM BOOT END SECTOR ON TRACK 1

                                ;
                                ;       HARD DISK PARAMETERS
                                ;

0030                    HDBASE  EQU     30H             ; HARD DISK CONTROLLER BASE ADDRESS
0030                    HDDATA  EQU     HDBASE+0        ; HARD DISK DATA REGISTER
0031                    HDPRCM  EQU     HDBASE+1        ; HARD DISK PRECOMPENSATION REGISTER
0033                    HDSEC   EQU     HDBASE+3        ; HARD DISK SECTOR REGISTER
0034                    HDCYLL  EQU     HDBASE+4        ; HARD DISK CYLINDER HIGH BYTE REGISTER
0035                    HDCYLH  EQU     HDBASE+5        ; HARD DISK CYLINDER LOW BYTE REGISTER
0036                    HDSEL   EQU     HDBASE+6        ; HARD DISK SECTOR/HEAD/DRIVE REGISTER
0037                    HDCMD   EQU     HDBASE+7        ; HARD DISK COMMAND REGISTER
0037                    HDSTAT  EQU     HDBASE+7        ; HARD DISK STATUS REGISTER

0010                    HOMEHD  EQU     00010000B       ; HARD DISK RESTORE COMMAND
0020                    READHD  EQU     00100000B       ; HARD DISK READ COMMAND
0030                    WRITHD  EQU     00110000B       ; HARD DISK WRITE COMMAND

0020                    PRECMP  EQU     20H             ; HARD DISK PRECOMPENSATION VALUE
0007                    HEAD    EQU     00000111B       ; HEAD SELECT
0007                    HDERRZ  EQU     7       ; HARD DISK ERROR

0003                    HDISK1  EQU     00000011B       ; 5MB HARD DISK
0005                    HDISK2  EQU     00000101B       ; 10MB HARD DISK

                                ;
                                ;       CONTROL REGISTER BIT ASSIGNMENT
                                ;

0000                    SELFA   EQU     0       ; SELECT FLOPPY DRIVE A
0001                    SELFB   EQU     1       ; SELECT FLOPPY DRIVE B
0002                    SELSID  EQU     2       ; SELECT FLOPPY SIDE
0003                    PTRSTR  EQU     3       ; PRINTER STROBE
0004                    KBDINT  EQU     4       ; KEYBOARD INTERRUPT MASK
0005                    PRINT   EQU     5       ; PRINTER INTERRUPT MASK
0006                    HDINT   EQU     6       ; HARD DISK INTERRUPT MASK

                                ;
                                ;       STATUS REGISTER BIT ASSIGNMENT
                                ;

0005                    PTRBSY  EQU     5       ; PRINTER BUSY
0004                    KCAPLK  EQU     4       ; KEYBOARD CAPITAL LOCK
000E                    INTREQ  EQU     00001110B       ; INTERRUPT REQUEST ID
0000                    INTRPT  EQU     0       ; INTERRUPT REQUEST
```

```
                        ;
                        ;       SYSTEM  SWITCHES
                        ;

0000                    PROMPT  EQU     0       ; PROMPT SUPPRESS
0001                    TV      EQU     1       ; TV/MONITOR

                        ;
                        ; PROGRAMMABLE VIDEO TIMING CONTROLLER PARAMETERS
                        ;

                        ;
                        ; REGISTER ASSIGNMENTS
                        ;

0000                    VTCBASE         EQU     00H
0000                    VTCINI          EQU     VTCBASE+00H     ; INITALIALIZATION REGISTER
0001                    VTCSTA          EQU     VTCBASE+01H     ; STATUS REGISTER
0001                    VTCCMD          EQU     VTCBASE+01H     ; COMMAND REGISTER
0002                    VTCSC1          EQU     VTCBASE+02H     ; SCREEN START REGISTER LOWER BYTE
0003                    VTCSC2          EQU     VTCBASE+03H     ; SCREEN START REGISTER UPPER BYTE
0004                    VTCCU1          EQU     VTCBASE+04H     ; CURSOR REGISTER LOWER BYTE
0005                    VTCCU2          EQU     VTCBASE+05H     ; CURSOR REGISTER UPPER BYTE
0006                    VTCPT1          EQU     VTCBASE+06H     ; POINTER REGISTER LOWER BYTE
0007                    VTCPT2          EQU     VTCBASE+07H     ; POINTER REGISTER UPPER BYTE
000D                    VTCINP          EQU     VTCBASE+0DH     ; INPUT DATA REGISTER
001C                    VTCDAT          EQU     VTCBASE+1CH     ; OUTPUT DATA REGISTER
001D                    VTCMOD          EQU     VTCBASE+1DH     ; MODE REGISTER

                        ;
                        ; PVTC COMMANDS
                        ;

0000                    RESET           EQU     00000000B       ; MASTER RESET
00AB                    WRINC           EQU     10101011B       ; WRITE AT CURSOR AND INCREMENT
00BB                    WRCP            EQU     10111011B       ; WRITE FROM CURSOR TO POINTER
00AA                    WRCUR           EQU     10101010B       ; WRITE AT CURSOR
00AC                    RDCUR           EQU     10101100B       ; READ AT CURSOR
00A4                    RDPT            EQU     10100100B       ; READ AT POINTER
004C                    MODE            EQU     01001100B       ; MODE REGISTER
003D                    ONDICU          EQU     00111101B       ; TURN ON CURSOR & DISPLAY
0031                    ONCUR           EQU     00110001B       ; TURN ON CURSOR
0030                    OFFCUR          EQU     00110000B       ; TURN OFF CURSOR
0029                    ONDIS           EQU     00101001B       ; TURN ON DISPLAY
0028                    OFFDIS          EQU     00101000B       ; TURN OFF DISPLAY
0040                    RSTFLG          EQU     01000000B       ; RESET FLAGS

                        ;
                        ; INITIALIZATION REGISTERS
                        ;

0044                    IR0             EQU     01000100B
0032                    IR1             EQU     00110010B
000C                    IR2             EQU     00001100B
00AA                    IR3             EQU     10101010B
```

```
0017                    IR4         EQU     00010111B
004F                    IR5         EQU     01001111B
0009                    IR6         EQU     00001001B
0029                    IR7         EQU     00101001B
0080                    IR8         EQU     10000000B
0010                    IR9         EQU     00010000B
0097                    IR10        EQU     10010111B

                        ;
                        ; MODE REGISTER BIT ASSIGNMENT
                        ;

0000                    COL40       EQU     0           ; 40 COLUMN
0001                    RVERSE      EQU     1           ; REVERSE VIDEO
0002                    HGHINT      EQU     2           ; HIGH INTENSITY
0003                    GRAPHIC     EQU     3           ; GRAPHIC
0004                    INTLCE      EQU     4           ; INTERLACE
0005                    BWSCN       EQU     5           ; BLACK & WHITE SCREEN
0006                    BLOCK       EQU     6           ; BLOCK
0007                    BEEP        EQU     7           ; ALARM SOUND

                        ;
                        ; FLAGS
                        ;

0005                    RDYFLG      EQU     5           ; PVTC READY FOR COMMAND
0004                    VBLANK      EQU     4           ; VBLANK PERIOD
0003                    LINE0       EQU     3           ; LINE ZERO
0002                    SPLIT       EQU     2           ; SPLIT SCREEN
0080                    FORGND      EQU     10000000B   ; FOREGROUND CHARACTER BIT
0018                    INTMSK      EQU     00011000B   ; VBLANK & LINE 0 INTERRUPTS

                        ;
                        ; ADDRESS
                        ;

0080                    STADR       EQU     080H        ; SCREEN START ADDRESS
07FF                    ENDADR      EQU     7FFH        ; SCREEN END ADDRESS

                        ;
                        ; CONSTANTS
                        ;

0020                    BLANK       EQU     20H
007E                    TILDA       EQU     7EH
0037                    COLBW       EQU     37H
0038                    COLFW       EQU     38H
0039                    LEFT        EQU     39H
0030                    RIGHT       EQU     30H

0050                    MAXCOL      EQU     80          ; MAXIMUM COLUMNS PER ROW
0017                    MAXLIN      EQU     23          ; MAXIMUM ROW - 1
0028                    MAXTVC      EQU     40          ; MAXIMUM TV COLUMNS PER ROW
0028                    MAXOFF      EQU     MAXCOL-MAXTVC   ; SCREEN OFFSET FOR TV
0005                    MARGIN      EQU     5           ; SCREEN WINDOW MARGINS
001D                    WIDTH       EQU     MAXTVC-MARGIN-MARGIN-1
```

```
                                                ; OFFSET BETWEEN LEFT & RIGHT LIMITS

                                        ORG     BIOS

                                        ;
                                        ;       JUMP VECTOR FOR INDIVIDUAL SUBROUTINES
                                        ;

E200'   C3 E378'                        JP      BOOT            ;COLD START
E203'   C3 E3E2'                WBOOTE: JP      WBOOT           ;WARM BOOT
E206'   C3 E4A8'                        JP      CONST           ;CONSOLE STATUS
E209'   C3 E4E4'                        JP      CONIN           ;CONSOLE CHARACTER IN
E20C'   C3 E59D'                        JP      CONOUT          ;CONSOLE CHARACTER OUT
E20F'   C3 EB5B'                        JP      LIST            ;LIST CHARACTER OUT
E212'   C3 EBE2'                        JP      PUNCH           ;PUNCH CHARACTER OUT
E215'   C3 EBE5'                        JP      READER          ;READER CHARACTER OUT
E218'   C3 EBE8'                        JP      HOME            ;MOVE HEAD TO HOME POSITION
E21B'   C3 EBF0'                        JP      SELDSK          ;SELECT DISK
E21E'   C3 EBEB'                        JP      SETTRK          ;SET TRACK NUMBER
E221'   C3 EC82'                        JP      SETSEC          ;SET SECTOR NUMBER
E224'   C3 EC99'                        JP      SETDMA          ;SET DMA ADDRESS
E227'   C3 ECB1'                        JP      RDDISK          ;READ DISK
E22A'   C3 EC9F'                        JP      WRDISK          ;WRITE DISK
E22D'   C3 EB94'                        JP      LISTST          ;RETURN LIST STATUS
E230'   C3 EC87'                        JP      SECTRAN         ;SECTOR TRANSLATE

                                        ;
                                        ;       LOGICAL TO PHYSICAL DISK TRANSLATION TABLE
                                        ;

E233'   02 FF FF FF             DSKMAP: DB      2,0FFH,0FFH,0FFH

                                        ;
                                        ;       DISK PARAMETER HEADERS FOR UP TO 5 TYPES OF ON-LINE DRIVES
                                        ;


                                        ;
                                        ;       DISK PARAMETER HEADER FOR DISK 00
                                        ;       5" SINGLE-SIDED, DOUBLE-DENSITY, 40 TRACKS
                                        ;

E237'   E29B' 0000              DPBASE: DW      TRANS0,0000H
E23B'   0000 0000                       DW      0000H,0000H
E23F'   F0C2' E2C4'                     DW      DIRBF,DPBLK0
E243'   F296' F142'                     DW      CHK00,ALL00
E247'   E31F' E30F'                     DW      DISK0,TAB0

                                        ;
                                        ;       DISK PARAMETER HEADER FOR DISK 01
                                        ;       5" DOUBLE-SIDED, DOUBLE DENSITY, 80 TRACKS
                                        ;

E24B'   E29B' 0000                      DW      TRANS0,0000H
E24F'   0000 0000                       DW      0000H,0000H
E253'   F0C2' E2D3'                     DW      DIRBF,DPBLK1
E257'   F2A6' F14E'                     DW      CHK01,ALL01
```

```
E25B' E32A' E30F'                DW      DISK1,TAB0

                                 ;
                                 ;       DISK PARAMETER HEADER FOR DISK 02
                                 ;       5" DOUBLE-SIDED, DOUBLE DENSITY, 80 TRACKS
                                 ;

E25F' E29B' 0000                 DW      TRANS0,0000H
E263' 0000 0000                  DW      0000H,0000H
E267' F0C2' E2E2'                DW      DIRBF,DPBLK2
E26B' F2E6' F17F'                DW      CHK02,ALL02
E26F' E335' E30F'                DW      DISK2,TAB0

                                 ;
                                 ;       DISK PARAMETER HEADER FOR DISK 03
                                 ;       5 MB WINCHESTER HARD DISK
                                 ;

E273' 0000 0000                  DW      0000H,0000H
E277' 0000 0000                  DW      0000H,0000H
E27B' F0C2' E2F1'                DW      DIRBF,DPBLK3
E27F' 0000 F1B0'                 DW      0000H,ALL03
E283' E340' E317'                DW      DISK3,TAB3

                                 ;
                                 ;       DISK PARAMETER HEADER FOR DISK 04
                                 ;       10 MB WINCHESTER HARD DISK
                                 ;

E287' 0000 0000                  DW      0000H,0000H
E28B' 0000 0000                  DW      0000H,0000H
E28F' F0C2' E300'                DW      DIRBF,DPBLK4
E293' 0000 F1FD'                 DW      0000H,ALL04
E297' E34B' E317'                DW      DISK4,TAB3

                                 ;
                                 ;       SECTOR TRANSLATION TABLE FOR 5" DOUBLE DENSITY
                                 ;

E29B' 28                 TRANS0: DB      40      ; LAST SECTOR ON DISK
E29C' 01 02 03 04                DB      1,2,3,4         ; SKEW FACTOR = 2
E2A0' 09 0A 0B 0C                DB      9,10,11,12
E2A4' 11 12 13 14                DB      17,18,19,20
E2A8' 19 1A 1B 1C                DB      25,26,27,28
E2AC' 21 22 23 24                DB      33,34,35,36
E2B0' 05 06 07 08                DB      5,6,7,8
E2B4' 0D 0E 0F 10                DB      13,14,15,16
E2B8' 15 16 17 18                DB      21,22,23,24
E2BC' 1D 1E 1F 20                DB      29,30,31,32
E2C0' 25 26 27 28                DB      37,38,39,40

                                 ;
                                 ;       DISK PARAMETER BLOCK FOR 5" SINGLE-SIDED DOUBLE DENSITY
                                 ;       512 BYTES/SECTOR, 10 SECTORS/TRACK, 40 TRACKS
                                 ;       2 SYSTEM TRACKS
                                 ;
```

```
E2C4'  0028          DPBLK0: DW   40 ; SECTORS PER TRACK (SPT)
E2C6'  04                    DB   4  ; BLOCK SHIFT FACTOR (BSH) , BLS = 2048
E2C7'  0F                    DB   15 ; BLOCK MASK (BLM)
E2C8'  01                    DB   1  ; EXTENT MASK (EXM)
E2C9'  005E                  DW   94 ; DISK SIZE-1 (DSM)
E2CB'  003F                  DW   63 ; DIRECTORY MAX (DRM)
E2CD'  80                    DB   128    ; ALLOC 0 (AL0)
E2CE'  00                    DB   0  ; ALLOC 1 (AL1)
E2CF'  0010                  DW   16 ; CHECK SIZE (CKS)
E2D1'  0002                  DW   2  ; RESERVED TRACKS (OFF)

                             ;
                             ;       DISK PARAMETER BLOCK FOR 5" DOUBLE-SIDED DOUBLE DENSITY
                             ;       512 BYTES/SECTOR, 10 SECTORS/TRACK, 80 TRACKS/SIDE
                             ;       2 SYSTEM TRACKS
                             ;

E2D3'  0028          DPBLK1: DW   40
E2D5'  04                    DB   4  ; BLS = 2048
E2D6'  0F                    DB   15
E2D7'  00                    DB   0
E2D8'  018A                  DW   394
E2DA'  007F                  DW   127
E2DC'  C0                    DB   192
E2DD'  00                    DB   0
E2DE'  0040                  DW   64
E2E0'  0002                  DW   2

                             ;
                             ;       DISK PARAMETER BLOCK FOR 5" DOUBLE-SIDED DOUBLE DENSITY
                             ;       512 BTYES/SECTOR, 10 SECTORS/TRACK, 80 TRACKS/SIDE
                             ;       3 SYSTEM TRACKS
                             ;

E2E2'  0028          DPBLK2: DW   40
E2E4'  04                    DB   4  ; BLS = 2048
E2E5'  0F                    DB   15
E2E6'  00                    DB   0
E2E7'  0187                  DW   391
E2E9'  00FF                  DW   255
E2EB'  F0                    DB   240
E2EC'  00                    DB   0
E2ED'  0040                  DW   64
E2EF'  0003                  DW   3

                             ;
                             ;       DISK PARAMETER BLOCK FOR 5 MB WINCHESTER HARD DISK
                             ;       512 BYTES/SECTOR, 16 SECTORS/TRACK, 154 TRACKS, 4 HEADS
                             ;

E2F1'  0040          DPBLK3: DW   64
E2F3'  06                    DB   6  ; BLS = 8192
E2F4'  3F                    DB   63
E2F5'  03                    DB   3
E2F6'  0265                  DW   613
```

```
E2F8'  03FF                    DW      1023
E2FA'  F0                      DB      240
E2FB'  00                      DB      0
E2FC'  0000                    DW      0
E2FE'  0002                    DW      2

                               ;
                               ;       DISK PARAMETER BLOCK FOR 5 MB WINCHESTER HARD DISK
                               ;       512 BYTES/SECTOR, 16 SECTORS/TRACK, 154 TRACKS, 4 HEADS
                               ;

E300'  0040            DPBLK4: DW      64
E302'  06                      DB      6  ; BLS = 8192
E303'  3F                      DB      63
E304'  03                      DB      3
E305'  04C2                    DW      1218
E307'  07FF                    DW      2047
E309'  FF                      DB      255
E30A'  00                      DB      0
E30B'  0000                    DW      0
E30D'  0002                    DW      2

                               ;
                               ;       FLOPPY DOUBLE DENSITY TABLE
                               ;

E30F'  0200            TAB0:   DW      512     ; HOST BUFFER SIZE
E311'  0800                    DW      2048    ; HOST ALLOCATION BLOCK SIZE
E313'  04                      DB      4  ; CP/M SECTOR PER HOST BUFFER ( HSTBLK )
E314'  28                      DB      40 ; LOGICAL SECTOR PER TRACK
E315'  03                      DB      3  ; SECTOR MASK ( HSTBLK - 1 )
E316'  02                      DB      2  ; LOG2( HSTBLK )

                               ;
                               ;       HARD DISK TABLE
                               ;

E317'  0200            TAB3:   DW      512
E319'  2000                    DW      8192
E31B'  04                      DB      4
E31C'  40                      DB      64
E31D'  03                      DB      3
E31E'  02                      DB      2

                               ;
                               ;       CURRENT VALUES FOR THE 5 DISKS
                               ;

E31F'  02              DISK0:  DB      00000010B       ; FLOPPY DISK SELECT BYTE
                                       ; BIT 2 = SIDE
                                       ; BIT 1 = SELECT FLOPPY DRIVE B
                                       ; BIT 0 = SELECT FLOPPY DRIVE A
E320'  0200                    DW      512             ; SECTOR SIZE
E322'  03                      DB      3       ; STEPPING RATE
E323'  01                      DB      1       ; SIDES
E324'  02                      DB      FDISK1          ; DISK TYPE
```

```
E325' 0000                    DW    0       ; LAST TRACK ACCESSED
E327' 0A                      DB    10      ; PHYSICAL SECTORS/TRACK
E328' 0028                    DW    40      ; CYLINDERS

E32A' 02              DISK1:  DB    00000010B
E32B' 0200                    DW    512
E32D' 00                      DB    0
E32E' 02                      DB    2
E32F' 04                      DB    FDISK2
E330' 0000                    DW    0
E332' 0A                      DB    10
E333' 0050                    DW    80

E335' 01              DISK2:  DB    00000001B
E336' 0200                    DW    512
E338' 00                      DB    0
E339' 02                      DB    2
E33A' 08                      DB    FDISK3
E33B' 0000                    DW    0
E33D' 0A                      DB    10
E33E' 0050                    DW    80

E340' 20              DISK3:  DB    00100000B       ; HARD DISK SELECT BYTE
                                    ; BIT 7 = 0
                                    ; BIT 6,5 = SECTOR SIZE
                                    ; BIT 4 - 3 = DRIVE
                                    ; BIT 2 - 0 = HEAD
E341' 0200                    DW    512
E343' 06                      DB    06H             ; 3 MS
E344' 04                      DB    4
E345' 03                      DB    HDISK1
E346' 0000                    DW    0
E348' 10                      DB    16
E349' 009A                    DW    154

E34B' 20              DISK4:  DB    00100000B       ; HARD DISK SELECT BYTE
                                    ; BIT 7 = 0
                                    ; BIT 6,5 = SECTOR SIZE
                                    ; BIT 4 - 3 = DRIVE
                                    ; BIT 2 - 0 = HEAD
E34C' 0200                    DW    512
E34E' 06                      DB    06H             ; 3 MS
E34F' 04                      DB    4
E350' 05                      DB    HDISK2
E351' 0000                    DW    0
E353' 10                      DB    16
E354' 0131                    DW    305

                              ;
                              ;     END OF FIXED TABLES
                              ;

E356' 30 30 6B 20             DB    '00k '
E35A' 43 50 2F 4D     SIGNON: DB    'CP/M vers 2.2 (STM REL. 1.01)',EOL
E35E' 20 76 65 72
E362' 73 20 32 2E
```

```
E366'  32 20 28 53
E36A'  54 4D 20 52
E36E'  45 4C 2E 20
E372'  31 2E 30 31
E376'  29 00

                                ;
                                ;       INDIVIDUAL SUBROUTINES TO PERFORM EACH FUNCTION
                                ;

                                ;
                                ;       COLD BOOT
                                ;

                                ;
                                ;       PARAMETER INITIALIZATION
                                ;

E378'  3E 20            BOOT:   LD      A,BLANK         ; CLEAR SCREEN
E37A'  D3 1C                    OUT     (VTCDAT),A
E37C'  21 0080                  LD      HL,STADR
E37F'  7D                       LD      A,L
E380'  D3 04                    OUT     (VTCCU1),A
E382'  7C                       LD      A,H
E383'  D3 05                    OUT     (VTCCU2),A
E385'  21 07FF                  LD      HL,ENDADR
E388'  7D                       LD      A,L
E389'  D3 06                    OUT     (VTCPT1),A
E38B'  7C                       LD      A,H
E38C'  D3 07                    OUT     (VTCPT2),A
E38E'  3E BB                    LD      A,WRCP
E390'  CD EB2D'                 CALL    WAIT
E393'  21 0080                  LD      HL,STADR
E396'  22 FFD7                  LD      (SCSTART),HL    ; SAVE SCREEN START ADDRESS
E399'  7D                       LD      A,L
E39A'  D3 04                    OUT     (VTCCU1),A
E39C'  D3 02                    OUT     (VTCSC1),A
E39E'  7C                       LD      A,H
E39F'  D3 05                    OUT     (VTCCU2),A
E3A1'  D3 03                    OUT     (VTCSC2),A

E3A3'  AF                       XOR     A
E3A4'  32 0003                  LD      (IOBYTE),A      ; CLEAR THE IOBYTE
E3A7'  32 0004                  LD      (CDISK),A       ; SELECT DISK ZERO
E3AA'  32 FFDF                  LD      (SWITCH),A      ; CLEAR SWITCHES
E3AD'  32 FFD4                  LD      (COLNO),A       ; SET TO COLUMN 0
E3B0'  32 FFD5                  LD      (COLNO+1),A
E3B3'  32 FFD6                  LD      (LINENO),A      ; SET TO LINE 0
E3B6'  32 FFD9                  LD      (OFFSET),A      ; SET SCREEN OFFSET TO 0
E3B9'  32 FFDA                  LD      (OFFSET+1),A
E3BC'  21 E233'                 LD      HL,DSKMAP       ; GET UP DISK TRANLATION TABLE ADDRESS
E3BF'  22 FFE3                  LD      (DMAPTP),HL     ; SAVE VAUE
E3C2'  21 E237'                 LD      HL,DPBASE       ; POINT TO DISK PARM BASE
E3C5'  22 FFE5                  LD      (DPBTP),HL      ; SAVE VALUE

E3C8'  21 E35A'                 LD      HL,SIGNON       ; PRINT SIGN ON MESSAGE
```

```
E3CB'  CD E3D1'                  CALL   PMSG

E3CE'  C3 E454'                  JP     GOCPM          ; INITIALIZE AND GO TO CP/M

                                 ;
                                 ;      TO PRINT OUT A MESSAGE
                                 ;
                                 ;         IN:         HL = ADDRESS OF MESSAGE
                                 ;         OUT:        NONE
                                 ;         CHANGED:    NONE
                                 ;

E3D1'  F5                PMSG:   PUSH   AF
E3D2'  C5                        PUSH   BC
E3D3'  E5                        PUSH   HL
E3D4'  7E                        LD     A,(HL)
E3D5'  4F                PMSG1:  LD     C,A            ; GET BYTE FROM MESSAGE
E3D6'  CD E59D'                  CALL   CONOUT         ; SEND BYTE OUT
E3D9'  23                        INC    HL
E3DA'  7E                        LD     A,(HL)         ; CHECK IF END OF MESSAGE
E3DB'  B7                        OR     A
E3DC'  20 F7                     JR     NZ,PMSG1
E3DE'  E1                        POP    HL
E3DF'  C1                        POP    BC
E3E0'  F1                        POP    AF
E3E1'  C9                        RET

                                 ;
                                 ;      WARM BOOT
                                 ;

E3E2'  3A FFE1           WBOOT:  LD     A,(IREG)       ; GET I REGISTER VALUE
E3E5'  ED 47                     LD     I,A            ; SET I REGISTER
E3E7'  FB                        EI             ; ENABLE INTERRUPTT

E3E8'  31 0080                   LD     SP,80H         ; SET STACK POINTER

E3EB'  3A F0A7'                  LD     A,(BUFACT)     ; CHECK IF BUFFER ACTIVE
E3EE'  FE FF                     CP     ON
E3F0'  CC EDA5'                  CALL   Z,WRHST        ; FLUSH BUFFER

E3F3'  16 0A                     LD     D,RETRY        ; SET WARM BOOT RETRY COUNT

                                 ;
                                 ;      SELECT DRIVE 0
                                 ;

E3F5'  0E 00             WBOOT1: LD     C,0            ; SELECT DISK 0
E3F7'  59                        LD     E,C            ; RESTORING DISK 0
E3F8'  CD EBF0'                  CALL   SELDSK
E3FB'  38 4C                     JR     C,WBERR        ; SELECT ERROR, RETRY

E3FD'  0E 02                     LD     C,TRKOST       ; READ TRACK 0
E3FF'  1E 0A                     LD     E,TRKOEN
E401'  21 CC00                   LD     HL,CCP         ; LOAD AT CCP
E404'  3E 82                     LD     A,READFD       ; GET READ COMMAND
```

```
E406'  32 F0AD'                LD     (DSKCMD),A     ; SAVE COMMAND
E409'  CD E48B'                CALL   RDSECS         ; READ SECTORS
E40C'  38 3B                   JR     C,WBERR        ; READ ERROR, RETRY

E40E'  3A F0B2'                LD     A,(SIDES)      ; GET NUMBER OF SIDES
E411'  FE 01                   CP     1       ; CHECK IF SINGLE-SIDED
E413'  20 0E                   JR     NZ,TWOSID      ; NO, DOUBLE-SIDED

E415'  3A F0B1'                LD     A,(STEPS)      ; GET STEPPING RATE OF BOOT DISK
E418'  F6 5C                   OR     STEPFD         ; PUT INTO STEP IN COMMAND
E41A'  CD FFF9                 CALL   FDHCMD         ; STEP IN
E41D'  E6 90                   AND    SKERR          ; CHECK IF ERROR
E41F'  20 28                   JR     NZ,WBERR       ; ERROR, RETRY
E421'  18 10                   JR     RDTRK1         ; READ TRACK 1

E423'  3A F0AE'          TWOSID: LD     A,(SELBYT)     ; GET SELECT BYTE
E426'  CB D7                   SET    SELSID,A       ; SELECT SIDE 1
E428'  CD EDD6'                CALL   DSKSEL
E42B'  3A F0AD'                LD     A,(DSKCMD)     ; MODIFY READ COMMAND
E42E'  CB DF                   SET    SIDE,A
E430'  32 F0AD'                LD     (DSKCMD),A

E433'  0E 01             RDTRK1: LD     C,TRK1ST       ; READ TRACK 1
E435'  1E 02                   LD     E,TRK1EN
E437'  CD E48B'                CALL   RDSECS         ; READ SECTORS
E43A'  38 0D                   JR     C,WBERR        ; READ ERROR, RETRY
E43C'  AF                      XOR    A
E43D'  32 CC07                 LD     (CCP+7),A      ; CLEAR AUTO EXECUTION

E440'  2A F0BC'                LD     HL,(LSTADR)    ; GET LAST TRACK ADDRESS
E443'  36 01                   LD     (HL),1         ; UPDATE LAST TRACK
E445'  23                      INC    HL
E446'  77                      LD     (HL),A
E447'  18 0B                   JR     GOCPM

E449'  15                WBERR: DEC    D       ; WARM BOOT ERROR
E44A'  20 A9                   JR     NZ,WBOOT1

E44C'  21 EFE3'                LD     HL,ERMSG5
E44F'  CD E3D1'                CALL   PMSG
E452'  F3                      DI
E453'  76                      HALT           ; HALT

                               ;
                               ;      SET UP VECTORS
                               ;         A = 0
                               ;

E454'  21 F09D'          GOCPM: LD     HL,WRTYPE      ; INITIALIZE DATA AREA
E457'  06 11                   LD     B,17
E459'  77                LOOP0: LD     (HL),A
E45A'  23                      INC    HL
E45B'  10 FC                   DJNZ   LOOP0

E45D'  01 FFFF                 LD     BC,TIMING
E460'  ED 43 F576'             LD     (TIMER),BC     ; INITIALIZE LIST DEVICE TIMER
```

```
E464'  3E C3                    LD      A,JMPCOD        ; GET JUMP INSTRUCTION
E466'  32 0000                  LD      (WARMST),A      ; SAVE IN WARM START VECTOR
E469'  21 E203'                 LD      HL,WBOOTE       ; POINT TO WARM BOOT ENTRY POINT
E46C'  22 0001                  LD      (REBOOT),HL
E46F'  32 0005                  LD      (SYSCAL),A      ; SET UP SYSTEM CALL VECTOR
E472'  21 D406                  LD      HL,BDOS         ; POINT TO BDOS ENTRY POINT
E475'  22 0006                  LD      (BDOSAD),HL
E478'  21 0E2D'                 LD      HL,SWLEAD-BIOS
E47B'  22 FAFE                  LD      (0FAFEH),HL
E47E'  01 0080                  LD      BC,80H          ; DEFAULT DMA ADDRESS IS 80H
E481'  CD EC99'                 CALL    SETDMA
E484'  3A 0004                  LD      A,(CDISK)       ; GET CURRENT DISK NUMBER
E487'  4F                       LD      C,A             ; SEND TO THE CCP
E488'  C3 CC00                  JP      CCP             ; GO TO CP/M FOR FURTHER PROCESSING

                                ;
                                ;       MULTIPLY SECTOR FLOPPY READ
                                ;
                                ;          IN:          C = STARTING SECTOR NUMBER
                                ;                       E = ENDING SECTOR NUMBER
                                ;                       HL = STARTING DMA ADDRESS
                                ;          OUT:                A = TYPE II STATUS
                                ;                       CARRY ON = ERROR RETURN
                                ;          CHANGED:     A,B,C,H,L
                                ;

E48B'  06 0A            RDSECS: LD      B,RETRY         ; GET RETRY COUNT
E48D'  E5               RDSTRY: PUSH    HL      ; SAVE CURRENT DMA
E48E'  79                       LD      A,C             ; SET SECTOR REGISTER
E48F'  D3 12                    OUT     (FDSEC),A
E491'  3A F0AD'                 LD      A,(DSKCMD)      ; GET READ COMMAND
E494'  CD FFF3                  CALL    FDRW            ; READ A SECTOR
E497'  B7                       OR      A
E498'  28 06                    JR      Z,NXTSEC        ; NO ERROR, READ NEXT SECTOR
E49A'  E1                       POP     HL      ; ERROR, RESTORE OLD DMA
E49B'  10 F0                    DJNZ    RDSTRY          ; RETRY
E49D'  37                       SCF             ; SET CARRY FLAG
E49E'  18 07                    JR      RDSRET          ; RETURN

E4A0'  F1               NXTSEC: POP     AF      ; SYNCHRONIZE PUSH POP
E4A1'  0C                       INC     C       ; INCREMENT TO SECTOR
E4A2'  7B                       LD      A,E             ; CHECK IF WITHIN LIMIT
E4A3'  B9                       CP      C
E4A4'  30 E5                    JR      NC,RDSECS       ; YES, READ NEXT SECTOR
E4A6'  3F                       CCF             ; CLEAR CARRY FLAG

E4A7'  C9               RDSRET: RET

                                ;
                                ;       GET CONSOLE STATUS
                                ;          A = 0FFH, CHARACTER READY
                                ;            = 000H, CHARACTER NOT READY
                                ;

E4A8'                   CONST:
E4A8'  3A FFDF                  LD      A,(SWITCH)
```

```
E4AB'  CB 4F                    BIT   TV,A          ; CHECK IF TV IS ON
E4AD'  28 0D                    JR    Z,CONST3      ; IGNORE IF NOT
E4AF'  3A F038'                 LD    A,(COUNTER)
E4B2'  B7                       OR    A
E4B3'  28 07                    JR    Z,CONST3
E4B5'  3D                       DEC   A
E4B6'  32 F038'                 LD    (COUNTER),A
E4B9'  CC E570'                 CALL  Z,SCNUPD
E4BC'                   CONST3: 
E4BC'  3A F035'                 LD    A,(KEYSW)
E4BF'  D6 FF                    SUB   ON
E4C1'  C8                       RET   Z

E4C2'  3A F032'                 LD    A,(CHRCNT)
E4C5'  B7                       OR    A
E4C6'  3E FF                    LD    A,0FFH
E4C8'  C0                       RET   NZ

E4C9'  CD FFE7                  CALL  CONSTA
E4CC'  30 0B                    JR    NC,CONST1     ; NO INPUT

E4CE'  CB 7F                    BIT   7,A           ; CHECK FOR FUNCTION KEYS
E4D0'  28 09                    JR    Z,CONST2      ; NOT FUNCTION KEY

E4D2'  CD E50A'                 CALL  INFUNC        ; PROCESS FUNCTION KEY
E4D5'  30 D1                    JR    NC,CONST      ; FUNCTON KEY TRAPPED
E4D7'  38 02                    JR    C,CONST2

E4D9'  AF               CONST1: XOR   A       ; NO CHARACTER
E4DA'  C9                       RET

E4DB'  32 F578'         CONST2: LD    (CHRBUF),A    ; SAVE CHARACTER
E4DE'  3E 01                    LD    A,1           ; TURN ON CHARACTER SWITCH
E4E0'  32 F032'                 LD    (CHRCNT),A
E4E3'  C9                       RET

                                ;
                                ;       GET CONSOLE CHARACTER
                                ;          A = CHARACTER
                                ;


E4E4'  CD E4A8'         CONIN:  CALL  CONST         ; CHECK FOR CHARACTER
E4E7'  B7                       OR    A
E4E8'  28 FA                    JR    Z,CONIN

E4EA'  2A F036'         CONIN2: LD    HL,(BUFADR)
E4ED'  7E                       LD    A,(HL)
E4EE'  F5                       PUSH  AF
E4EF'  23                       INC   HL
E4F0'  22 F036'                 LD    (BUFADR),HL
E4F3'  3A F032'                 LD    A,(CHRCNT)
E4F6'  3D                       DEC   A
E4F7'  32 F032'                 LD    (CHRCNT),A
E4FA'  B7                       OR    A
E4FB'  20 0B                    JR    NZ,LOOP22
```

```
E4FD'  21 F578'              LD      HL,CHRBUF
E500'  22 F036'              LD      (BUFADR),HL
E503'  3E 0A                 LD      A,TVCNT
E505'  32 F038'              LD      (COUNTER),A
E508'  F1              LOOP22: POP     AF
E509'  C9                    RET             ; RETURN

                             ;
                             ;       PROCESS FUNCTION KEYS
                             ;          A = CHARACTER
                             ;

E50A'  F5              INFUNC: PUSH    AF
E50B'  3A FFDF               LD      A,(SWITCH)      ; CHECK IF TV SWITCH ON
E50E'  CB 4F                 BIT     TV,A
E510'  28 2A                 JR      Z,INRET1        ; NO, RETURN
E512'  F1                    POP     AF
E513'  CB BF                 RES     7,A             ; CLEAR HIGH BIT

E515'  FE 38           IN06: CP      COLFW           ; CHECK IF COLUMN SCROLL FORWARD
E517'  20 05                 JR      NZ,IN07

E519'  CD E541'              CALL    COLFOR          ; COLUMN SCROLL FORWARD
E51C'  18 1C                 JR      INRET           ; RETURN

E51E'  FE 37           IN07: CP      COLBW           ; CHECK IF COLUMN SCROLL BACKWARD
E520'  20 05                 JR      NZ,IN08

E522'  CD E556'              CALL    COLBCK          ; COLUMN SCROLL BACKWARD
E525'  18 13                 JR      INRET           ; RETURN

E527'  FE 39           IN08: CP      LEFT            ; CHECK IF SCROLL LEFT MARGIN
E529'  20 06                 JR      NZ,IN09

E52B'  AF                    XOR     A
E52C'  32 FFD9               LD      (OFFSET),A      ; SET SCREEN OFFSET TO BEGINNING

E52F'  18 09                 JR      INRET           ; RETURN

E531'  FE 30           IN09: CP      RIGHT           ; CHECK IF SCROLL RIGHT MARGIN
E533'  20 08                 JR      NZ,INRET2

E535'  3E 28                 LD      A,MAXOFF
E537'  32 FFD9               LD      (OFFSET),A      ; SET SCREEN OFFSET TO END

E53A'  AF              INRET: XOR     A
E53B'  C9                    RET             ; RETURN
E53C'  F1              INRET1: POP     AF
E53D'  CB FF           INRET2: SET     7,A
E53F'  37                    SCF
E540'  C9                    RET

                             ;
                             ; TV SCREEN COLUMN SCROLL FORWARD
                             ;
```

```
E541'  3A FFD9          COLFOR: LD      A,(OFFSET)      ; GET SCREEN OFFSET
E544'  FE 28                    CP      MAXOFF          ; CHECK IF AT END
E546'  28 0A                    JR      Z,COLFRT        ; YES, RETURN

E548'  E6 F8                    AND     0F8H            ; CALCULATE NEXT POSITION
E54A'  C6 08                    ADD     A,8

E54C'  FE 28                    CP      MAXOFF          ; CHECK IF OVER LIMIT
E54E'  38 02                    JR      C,COLFRT        ; NO

E550'  3E 28                    LD      A,MAXOFF        ; YES, SET TO MAXIMUM

E552'  32 FFD9          COLFRT: LD      (OFFSET),A      ; UPDATE OFFSET
E555'  C9                       RET

                                ;
                                ; TV SCREEN COLUMN SCROLL BACKWARD
                                ;

E556'  3A FFD9          COLBCK: LD      A,(OFFSET)      ; GET SCREEN OFFSET
E559'  B7                       OR      A       ; CHECK IF AT BEGINNING
E55A'  28 13                    JR      Z,COLBRT        ; YES, RETURN

E55C'  E6 07                    AND     07H             ; CALCULATE NEXT POSITION
E55E'  20 07                    JR      NZ,COLBK1
E560'  3A FFD9                  LD      A,(OFFSET)
E563'  D6 08                    SUB     8
E565'  18 05                    JR      COLBK2

E567'  3A FFD9          COLBK1: LD      A,(OFFSET)
E56A'  E6 F8                    AND     0F8H
E56C'  32 FFD9          COLBK2: LD      (OFFSET),A      ; UPDATE OFFSET

E56F'  C9               COLBRT: RET

                                ;
                                ; UPDATE TV SCREEN
                                ;

E570'  C5               SCNUPD: PUSH    BC
E571'  3A FFD4                  LD      A,(COLNO)       ; GET COLUMN NUMBER
E574'  47                       LD      B,A
E575'  3A FFD9                  LD      A,(OFFSET)      ; GET CURRENT OFFSET
E578'  0E 05                    LD      C,MARGIN        ; CALCULATE LEFT LIMIT
E57A'  81                       ADD     A,C
E57B'  B8                       CP      B       ; CHECK IF LESS THAN LEFT LIMIT
E57C'  38 07                    JR      C,SCNUP1        ; CHECK IF LESS THAN RIGHT LIMIT

E57E'  78                       LD      A,B
E57F'  91                       SUB     C       ; CALCULATE OFFSET
E580'  30 16                    JR      NC,SCNUP2       ; NOT IN LEFTMOST MARGIN, RETURN

E582'  AF                       XOR     A       ; CLEAR OFFSET
E583'  18 13                    JR      SCNUP2          ; RETURN
```

```
E585'  C6 1D            SCNUP1: ADD   A,WIDTH       ; CALCULATE RIGHT LIMIT
E587'  B8                       CP    B
E588'  30 11                    JR    NC,SCNRET     ; NO CHANGE TO LIMITS

E58A'  4F                       LD    C,A
E58B'  78                       LD    A,B
E58C'  91                       SUB   C
E58D'  4F                       LD    C,A
E58E'  3A FFD9                  LD    A,(OFFSET)
E591'  81                       ADD   A,C
E592'  FE 28                    CP    MAXOFF        ; CHECK IF GREATER THAN MAXIMUM
E594'  38 02                    JR    C,SCNUP2      ; NO
E596'  3E 28                    LD    A,MAXOFF      ; YES, SET TO MAXIMUM

E598'  32 FFD9          SCNUP2: LD    (OFFSET),A

E59B'  C1               SCNRET: POP   BC
E59C'  C9                       RET

                                ;
                                ;       PUT CONSOLE CHARACTER
                                ;          C = CHARACTER
                                ;

E59D'  F5               CONOUT: PUSH  AF
E59E'  C5                       PUSH  BC
E59F'  D5                       PUSH  DE
E5A0'  E5                       PUSH  HL
E5A1'  3E 0A                    LD    A,TVCNT
E5A3'  32 F038'                 LD    (COUNTER),A
E5A6'  79                       LD    A,C           ; STRIP FIRST BIT
E5A7'  E6 7F                    AND   7FH
E5A9'  4F                       LD    C,A
E5AA'  3A F033'                 LD    A,(GRPHSW)    ; CHECK IF GRAPHIC CHARACTER
E5AD'  FE FF                    CP    ON
E5AF'  20 0A                    JR    NZ,CON1       ; NO

E5B1'  3E 00                    LD    A,OFF         ; TURN OFF SWITCH
E5B3'  32 F033'                 LD    (GRPHSW),A

E5B6'  CD E634'                 CALL  CHAROUT       ; SEND CHARACTER
E5B9'  18 62                    JR    CONEND

E5BB'  3A F030'         CON1:   LD    A,(CURCNT)
E5BE'  B7                       OR    A
E5BF'  20 12                    JR    NZ,CON1C

E5C1'  3A F02A'                 LD    A,(LDFLG)
E5C4'  FE FF                    CP    ON
E5C6'  20 19                    JR    NZ,CON2

E5C8'  3E 3D                    LD    A,'='
E5CA'  B9                       CP    C
E5CB'  20 14                    JR    NZ,CON2

E5CD'  CD E8C8'                 CALL  DIRADR
```

```
E5D0'  C3 E61D'              JP    CONEND

E5D3'  FE 01          CON1C: CP    1       ; CHECK IF COLUMN ADDRESS
E5D5'  20 05                 JR    NZ,CON1A        ; NO

E5D7'  CD E8CE'              CALL  DIADR1          ; PROCESS COLUMN ADDRESS
E5DA'  18 41                 JR    CONEND

E5DC'  CD E8FD'       CON1A: CALL  DIADR2          ; PROCESS ROW NUMBER
E5DF'  18 3C                 JR    CONEND

E5E1'  79             CON2:  LD    A,C
E5E2'  FE 20                 CP    BLANK           ; CHECK IF DISPLAYABLE
E5E4'  38 0E                 JR    C,CON4          ; NO

E5E6'  FE 7E                 CP    TILDA           ; CHECK IF TILDA
E5E8'  20 05                 JR    NZ,CON3A        ; NO, SEND CHARACTER

E5EA'  CD E830'       CON3:  CALL  TLDA            ; PROCESS TILDA
E5ED'  18 2E                 JR    CONEND

E5EF'  CD E634'       CON3A: CALL  CHAROUT         ; SEND CHARACTER
E5F2'  18 24                 JR    CON5

E5F4'  21 F079'       CON4:  LD    HL,LEADTB       ; GET LEAD-IN TABLE ADDRESS
E5F7'  06 00                 LD    B,0             ; INDEX INTO TABLE
E5F9'  09                    ADD   HL,BC
E5FA'  7E                    LD    A,(HL)          ; GET LEAD-IN INDICATOR
E5FB'  57                    LD    D,A
E5FC'  3A F029'              LD    A,(LEADFLG)     ; GET LEAD-IN FLAG
E5FF'  BA                    CP    D       ; CHECK IF SAME
E600'  20 0B                 JR    NZ,CON6         ; NO, PROCESS CHARACTER

E602'  21 F039'              LD    HL,CHRTAB       ; GET ROUTINE TABLE ADDRESS
E605'  CB 21                 SLA   C       ; MULITPLY BY 2
E607'  09                    ADD   HL,BC           ; INDEX INTO TABLE
E608'  CD E62D'              CALL  JUMP            ; PROCESS CHARACTER
E60B'  18 0B                 JR    CON5

E60D'  FE FF          CON6:  CP    ON      ; CHECK IF LEAD-IN ON
E60F'  20 0C                 JR    NZ,CONEND

E611'  3E 00                 LD    A,OFF           ; CLEAR LEAD-IN SWITCH
E613'  32 F029'              LD    (LEADFLG),A
E616'  18 DC                 JR    CON4

E618'  3E 00          CON5:  LD    A,OFF           ; CLEAR SWITCHES
E61A'  32 F029'              LD    (LEADFLG),A

E61D'  3A F031'       CONEND: LD   A,(CURSW)       ; GET CURSOR SWITCH
E620'  FE FF                 CP    ON      ; CHECK IF ON
E622'  20 04                 JR    NZ,CONRT        ; NO

E624'  3E 31                 LD    A,ONCUR         ; TURN ON CURSOR
E626'  D3 01                 OUT   (VTCCMD),A
```

```
E628'  E1              CONRT:  POP     HL
E629'  D1                      POP     DE
E62A'  C1                      POP     BC
E62B'  F1                      POP     AF
E62C'  C9                      RET

E62D'  EB              JUMP:   EX      DE,HL
E62E'  1A                      LD      A,(DE)
E62F'  6F                      LD      L,A
E630'  13                      INC     DE
E631'  1A                      LD      A,(DE)
E632'  67                      LD      H,A
E633'  E9                      JP      (HL)

                               ;
                               ; CHARACTER OUTPUT TO SCREEN
                               ;
                               ;       C = CHARACTER
                               ;

E634'                  CHAROUT:
E634'  3A F028'                LD      A,(BKFRFLG)     ; GET BACK/FOREGROUND FLAG
E637'  B1                      OR      C       ; SET CHARACTER TO GROUND TYPE
E638'  D3 1C                   OUT     (VTCDAT),A      ; SEND CHARACTER

E63A'  3A FFD4                 LD      A,(COLNO)       ; CHECK IF END OF LINE
E63D'  3C                      INC     A
E63E'  FE 50                   CP      MAXCOL
E640'  20 1C                   JR      NZ,CHROT        ; NO, SEND CHARACTER

E642'  3A F02B'                LD      A,(SWANL)       ; CHECK IF LINE TRUNCATE
E645'  FE FF                   CP      ON
E647'  20 1F                   JR      NZ,CHROT1       ; YES

E649'  AF                      XOR     A       ; CHECK IF SCROLL
E64A'  32 FFD4                 LD      (COLNO),A
E64D'  3A F027'                LD      A,(SCRLFLG)
E650'  FE 00                   CP      OFF
E652'  28 1B                   JR      Z,CHKLIN        ; NO, UPDATE LINE NUMBER

E654'  3E AB                   LD      A,WRINC         ; WRITE CHARACTER & INCREMENT CURSOR
E656'  CD EB2D'                CALL    WAIT

E659'  CD E73D'                CALL    SCROLL          ; SRCOLL SCREEN
E65C'  18 19                   JR      CHORET          ; RETURN

E65E'  32 FFD4         CHROT:  LD      (COLNO),A       ; UPDATE COLUMN NUMBER
E661'  3E AB                   LD      A,WRINC         ; WRITE CHARACTER & INCREMENT CURSOR
E663'  CD EB2D'                CALL    WAIT
E666'  18 0F                   JR      CHORET          ; RETURN

E668'  3E AA           CHROT1: LD      A,WRCUR         ; WRITE CHARACTER AT CURSOR
E66A'  CD EB2D'                CALL    WAIT
E66D'  18 08                   JR      CHORET          ; RETURN

E66F'  3E AB           CHKLIN: LD      A,WRINC         ; WRITE CHARACTER & INCREMENT CURSOR
```

```
E671'  CD EB2D'                CALL   WAIT
E674'  CD E78E'                CALL   UPDLIN          ; UPDATE LINE NUMBER

E677'  C9              CHORET: RET            ; RETURN

                               ;
                               ; LINE FEED
                               ;

E678'  3A F027'        LFEED:  LD     A,(SCRLFLG)     ; PROCESS SCROLLING
E67B'  FE 00                   CP     0FF
E67D'  28 05                   JR     Z,LFEED1

E67F'  CD E73D'                CALL   SCROLL
E682'  18 31                   JR     LFRET           ; RETURN

E684'  DB 04           LFEED1: IN     A,(VTCCU1)      ; GET CURSOR ADDRESS
E686'  6F                      LD     L,A
E687'  DB 05                   IN     A,(VTCCU2)
E689'  67                      LD     H,A
E68A'  E5                      PUSH   HL      ; SAVE CURSOR
E68B'  ED 5B FFD4              LD     DE,(COLNO)      ; CHECK IF AT LAST LINE
E68F'  AF                      XOR    A
E690'  ED 52                   SBC    HL,DE
E692'  11 004F                 LD     DE,MAXCOL-1
E695'  19                      ADD    HL,DE
E696'  11 07FF                 LD     DE,ENDADR
E699'  AF                      XOR    A
E69A'  ED 52                   SBC    HL,DE
E69C'  E1                      POP    HL
E69D'  28 05                   JR     Z,LFEED2        ; YES

E69F'  11 0050                 LD     DE,MAXCOL       ; CLACULATE NEXT LINE ADDRESS
E6A2'  18 07                   JR     LFEED3

E6A4'  21 0080         LFEED2: LD     HL,STADR        ; CALCULATE NEXT LINE ADDRESS
E6A7'  ED 5B FFD4              LD     DE,(COLNO)
E6AB'  19              LFEED3: ADD    HL,DE
E6AC'  7D                      LD     A,L             ; UPDATE CURSOR
E6AD'  D3 04                   OUT    (VTCCU1),A
E6AF'  7C                      LD     A,H
E6B0'  D3 05                   OUT    (VTCCU2),A

E6B2'  CD E78E'                CALL   UPDLIN          ; UPDATE LINE NUMBER

E6B5'  C9              LFRET:  RET            ; RETURN

                               ;
                               ; CARRIAGE RETURN
                               ;

E6B6'  DB 04           CARRTN: IN     A,(VTCCU1)      ; UPDATE CURSOR & COLUMN NUMBER
E6B8'  6F                      LD     L,A
E6B9'  DB 05                   IN     A,(VTCCU2)
E6BB'  67                      LD     H,A
E6BC'  ED 5B FFD4              LD     DE,(COLNO)
```

```
E6C0'  AF                    XOR   A
E6C1'  32 FFD4               LD    (COLNO),A
E6C4'  ED 52                 SBC   HL,DE
E6C6'  7D                    LD    A,L
E6C7'  D3 04                 OUT   (VTCCU1),A
E6C9'  7C                    LD    A,H
E6CA'  D3 05                 OUT   (VTCCU2),A

E6CC'  3A F02C'              LD    A,(SWALF)      ; CHECK IF AUTO LINEFEED
E6CF'  FE FF                 CP    ON
E6D1'  CC E678'              CALL  Z,LFEED

E6D4'  C9                    RET                  ; RETURN

                             ;
                             ; BACKSPACE
                             ;

E6D5'                  BKSPACE:
E6D5'  3A FFD4               LD    A,(COLNO)      ; CHECK IF AT BEGINNING OF LINE
E6D8'  B7                    OR    A
E6D9'  20 11                 JR    NZ,BSMOVL      ; NO, MOVE LEFT

E6DB'  3A FFD6               LD    A,(LINENO)     ; CHECK IF AT FIRST LINE
E6DE'  B7                    OR    A
E6DF'  28 29                 JR    Z,BSRET        ; YES, RETURN

E6E1'  3D                    DEC   A       ; UPDATE LINE NUMBER
E6E2'  32 FFD6               LD    (LINENO),A
E6E5'  3E 00                 LD    A,OFF          ; TURN OFF SCROLLING
E6E7'  32 F027'              LD    (SCRLFLG),A
E6EA'  3E 50                 LD    A,MAXCOL

E6EC'  3D              BSMOVL: DEC   A
E6ED'  32 FFD4               LD    (COLNO),A      ; UPDATE COLUMN NUMBER

E6F0'  DB 04                 IN    A,(VTCCU1)     ; UPDATE CURSOR NUMBER
E6F2'  5F                    LD    E,A
E6F3'  DB 05                 IN    A,(VTCCU2)
E6F5'  57                    LD    D,A
E6F6'  21 0080               LD    HL,STADR       ; CHECK IF AT START ADDRESS
E6F9'  AF                    XOR   A
E6FA'  ED 52                 SBC   HL,DE
E6FC'  20 05                 JR    NZ,BSMOV1      ; NO

E6FE'  11 07FF               LD    DE,ENDADR      ; SET CURSOR TO END ADDRESS
E701'  18 01                 JR    BSOUT

E703'  1B              BSMOV1: DEC   DE      ; DECREMENT CURSOR
E704'  7B              BSOUT:  LD    A,E
E705'  D3 04                 OUT   (VTCCU1),A
E707'  7A                    LD    A,D
E708'  D3 05                 OUT   (VTCCU2),A

E70A'  C9              BSRET:  RET                  ; RETURN
```

```
                                ;
                                ; CURSOR RIGHT
                                ;

E70B'  3A FFD4          CURIGT: LD      A,(COLNO)       ; CHECK IF AT END OF LINE
E70E'  3C                       INC     A
E70F'  FE 50                    CP      MAXCOL
E711'  20 0C                    JR      NZ,CURIG1

E713'  3A FFD6                  LD      A,(LINENO)      ; CHECK IF AT LAST LINE
E716'  FE 17                    CP      MAXLIN
E718'  28 22                    JR      Z,CURET         ; YES, RETURN

E71A'  3C                       INC     A       ; INCREMENT LINE NUMBER
E71B'  32 FFD6                  LD      (LINENO),A
E71E'  AF                       XOR     A

E71F'  32 FFD4          CURIG1: LD      (COLNO),A
E722'  DB 04                    IN      A,(VTCCU1)
E724'  5F                       LD      E,A
E725'  DB 05                    IN      A,(VTCCU2)
E727'  57                       LD      D,A
E728'  21 07FF                  LD      HL,ENDADR       ; CHECK IF AT END ADDRESS
E72B'  AF                       XOR     A
E72C'  ED 52                    SBC     HL,DE
E72E'  20 05                    JR      NZ,CURIG2       ; NO

E730'  11 0080                  LD      DE,STADR        ; SET TO START ADDRESS
E733'  18 01                    JR      CURIG3

E735'  13               CURIG2: INC     DE
E736'  7B               CURIG3: LD      A,E
E737'  D3 04                    OUT     (VTCCU1),A
E739'  7A                       LD      A,D
E73A'  D3 05                    OUT     (VTCCU2),A

E73C'  C9               CURET:  RET

                                ;
                                ; SCROLL
                                ;

E73D'  3E 30            SCROLL: LD      A,OFFCUR        ; TURN OFF CURSOR
E73F'  D3 01                    OUT     (VTCCMD),A

E741'  ED 5B FFD7               LD      DE,(SCSTART)    ; SET CURSOR TO FIRST LINE
E745'  7B                       LD      A,E
E746'  D3 04                    OUT     (VTCCU1),A
E748'  7A                       LD      A,D
E749'  D3 05                    OUT     (VTCCU2),A
E74B'  21 004F                  LD      HL,MAXCOL-1
E74E'  19                       ADD     HL,DE
E74F'  7D                       LD      A,L             ; SET POINTER TO END OF FIRST LINE
E750'  D3 06                    OUT     (VTCPT1),A
E752'  7C                       LD      A,H
E753'  D3 07                    OUT     (VTCPT2),A
```

```
E755'  3A F028'                LD      A,(BKFRFLG)     ; SEND OUT BLANK
E758'  F6 20                   OR      BLANK
E75A'  D3 1C                   OUT     (VTCDAT),A

E75C'  CD EB28'                CALL    WAIT2           ; CLEAR LINE

E75F'  E5                      PUSH    HL      ; UPDATE CURSOR
E760'  2A FFD4                 LD      HL,(COLNO)
E763'  19                      ADD     HL,DE
E764'  7D                      LD      A,L
E765'  D3 04                   OUT     (VTCCU1),A
E767'  7C                      LD      A,H
E768'  D3 05                   OUT     (VTCCU2),A

E76A'  D1                      POP     DE
E76B'  AF                      XOR     A       ; CHECK IF SECOND LINE WRAP AROUND
E76C'  21 07FF                 LD      HL,ENDADR
E76F'  ED 52                   SBC     HL,DE
E771'  20 05                   JR      NZ,SCRLL1

E773'  11 0080                 LD      DE,STADR        ; YES, SET TO BEGINNING ADDRESS
E776'  18 01                   JR      SCRLL2

E778'  13              SCRLL1: INC     DE
E779'  CD EB38'        SCRLL2: CALL    WAIT1           ; WAIT FOR VBLANK
E77C'  3A FFDF                 LD      A,(SWITCH)      ; CHECK IF TV SWITCH ON
E77F'  CB 4F                   BIT     TV,A
E781'  20 06                   JR      NZ,SCRLL3       ; YES

E783'  7B                      LD      A,E             ; SET SCREEN START ADDRESS
E784'  D3 02                   OUT     (VTCSC1),A
E786'  7A                      LD      A,D
E787'  D3 03                   OUT     (VTCSC2),A

E789'  ED 53 FFD7      SCRLL3: LD      (SCSTART),DE    ; SAVE SCREEN START ADDRESS
E78D'  C9                      RET             ; RETURN


                               ;
                               ; UPDATE LINE NUMBER
                               ;

E78E'  3A FFD6         UPDLIN: LD      A,(LINENO)      ; UPDATE LINE NUMBER
E791'  3C                      INC     A
E792'  32 FFD6                 LD      (LINENO),A
E795'  FE 17                   CP      MAXLIN          ; CHECK IF AT LAST LINE
E797'  20 05                   JR      NZ,UPDLRT       ; NO, RETURN

E799'  3E FF                   LD      A,ON            ; SET SCROLL FLAG ON
E79B'  32 F027'                LD      (SCRLFLG),A

E79E'  C9              UPDLRT: RET

                               ;
                               ; TURN ON BELL
```

```
                                ;
E79F'  11 00A0          BELL:   LD      DE,160
E7A2'  3A FFE0          LOOP16: LD      A,(MODREG)      ; GET MODE REGISTER VALUE
E7A5'  CB FF                    SET     BEEP,A
E7A7'  D3 1D                    OUT     (VTCMOD),A      ; SET BEEP MODE

E7A9'  06 4E                    LD      B,78
E7AB'  10 FE            LOOP13: DJNZ    LOOP13

E7AD'  3A FFE0                  LD      A,(MODREG)
E7B0'  D3 1D                    OUT     (VTCMOD),A      ; CLEAR BEEP MODE
E7B2'  06 4E                    LD      B,78
E7B4'  10 FE            LOOP14: DJNZ    LOOP14          ; DELAY
E7B6'  1B                       DEC     DE
E7B7'  7B                       LD      A,E
E7B8'  B2                       OR      D
E7B9'  20 E7                    JR      NZ,LOOP16
E7BB'  C9                       RET

                                ;
                                ; LOCK KEYBOARD
                                ;

E7BC'  3E FF            KLOCK:  LD      A,ON
E7BE'  32 F035'                 LD      (KEYSW),A
E7C1'  C9                       RET

                                ;
                                ; UNLOCK KEYBOARD
                                ;

E7C2'  3E 00            KUNLK:  LD      A,OFF
E7C4'  32 F035'                 LD      (KEYSW),A
E7C7'  AF                       XOR     A
E7C8'  32 F032'                 LD      (CHRCNT),A

E7CB'  CD E4A8'         LOOP21: CALL    CONST
E7CE'  B7                       OR      A
E7CF'  C8                       RET     Z

E7D0'  CD E4E4'                 CALL    CONIN
E7D3'  18 F6                    JR      LOOP21

                                ;
                                ; SEND CURSOR ADDRESS
                                ;

E7D5'  3E 03            SNDADR: LD      A,3
E7D7'  32 F032'                 LD      (CHRCNT),A

E7DA'  21 F578'                 LD      HL,CHRBUF
E7DD'  3A FFD4                  LD      A,(COLNO)
E7E0'  B7                       OR      A
E7E1'  20 04                    JR      NZ,SNDAD1
```

```
E7E3'  CB 77                    BIT     6,A
E7E5'  CB 6F                    BIT     5,A

E7E7'  77               SNDAD1: LD      (HL),A
E7E8'  23                       INC     HL
E7E9'  3A FFD6                  LD      A,(LINENO)
E7EC'  B7                       OR      A
E7ED'  20 02                    JR      NZ,SNDAD2

E7EF'  CB 6F                    BIT     5,A
E7F1'  77               SNDAD2: LD      (HL),A
E7F2'  23                       INC     HL
E7F3'  3E 00                    LD      A,CR
E7F5'  77                       LD      (HL),A
E7F6'  C9                       RET

                                ;
                                ; TURN ON DISPLAY
                                ;

E7F7'  3E 29            OND:    LD      A,ONDIS
E7F9'  D3 01                    OUT     (VTCCMD),A
E7FB'  C9                       RET

                                ;
                                ; TURN OFF DISPLAY
                                ;

E7FC'  3E 28            OFFD:   LD      A,OFFDIS
E7FE'  D3 01                    OUT     (VTCCMD),A
E800'  C9                       RET

                                ;
                                ; TURN ON CURSOR
                                ;

E801'  3E 31            ONC:    LD      A,ONCUR
E803'  D3 01                    OUT     (VTCCMD),A
E805'  3E FF                    LD      A,ON
E807'  32 F031'                 LD      (CURSW),A
E80A'  C9                       RET

                                ;
                                ; TURN OFF CURSOR
                                ;

E80B'  3E 30            OFFC:   LD      A,OFFCUR
E80D'  D3 01                    OUT     (VTCCMD),A
E80F'  3E 00                    LD      A,OFF
E811'  32 F031'                 LD      (CURSW),A
E814'  C9                       RET

                                ;
                                ; GRAPHIC CHARACTER
                                ;
```

```
E815'  3E FF            GRAPH:  LD      A,ON
E817'  32 F033'                 LD      (GRPHSW),A
E81A'  C9                       RET

                                ;
                                ; SET FOREGROUND MODE
                                ;

E81B'  3E 80            SETFR:  LD      A,FORGND
E81D'  32 F028'                 LD      (BKFRFLG),A
E820'  C9                       RET

                                ;
                                ; SET BACKGROUND MODE
                                ;

E821'  AF               SETBK:  XOR     A
E822'  32 F028'                 LD      (BKFRFLG),A
E825'  C9                       RET

                                ;
                                ; ESCAPE
                                ;

E826'  3A F02D'         ESCAPE: LD      A,(SWLEAD)      ; CHECK IF LEAD-IN CHARACTER
E829'  FE FF                    CP      ON
E82B'  C0                       RET     NZ      ; IGNORE

E82C'  32 F02A'                 LD      (LDFLG),A       ; SET LEAD-IN FLAG
E82F'  C9                       RET

                                ;
                                ; TILDA
                                ;

E830'  3A F02D'         TLDA:   LD      A,(SWLEAD)      ; CHECK IF LEAD-IN CHARACTER
E833'  FE FF                    CP      ON
E835'  28 06                    JR      Z,TLDA1         ; DISPLAY CHARACTER

E837'  3E FF                    LD      A,ON            ; SET LEAD-IN FLAG
E839'  32 F029'                 LD      (LEADFLG),A
E83C'  C9                       RET

E83D'  CD E634'         TLDA1:  CALL    CHAROUT         ; SEND CHARACTER
E840'  C9                       RET

                                ;
                                ; CURSOR UP
                                ;

E841'  3A FFD6          CURUP:  LD      A,(LINENO)      ; CHECK IF AT FIRST LINE
E844'  B7                       OR      A
E845'  28 3C                    JR      Z,CUPRET        ; YES, RETURN

E847'  3D                       DEC     A       ; UPDATE LINE NUMBER
E848'  32 FFD6                  LD      (LINENO),A
```

```
E84B'  3E 00                    LD      A,OFF           ; TURN OFF SCROLLING
E84D'  32 F027'                 LD      (SCRLFLG),A

E850'  DB 04                    IN      A,(VTCCU1)      ; GET CURSOR ADDRESS
E852'  6F                       LD      L,A
E853'  DB 05                    IN      A,(VTCCU2)
E855'  67                       LD      H,A

E856'  E5                       PUSH    HL      ; CHECK IF LINE AT START ADDRESS
E857'  ED 5B FFD4               LD      DE,(COLNO)
E85B'  AF                       XOR     A
E85C'  ED 52                    SBC     HL,DE
E85E'  11 0080                  LD      DE,STADR
E861'  AF                       XOR     A
E862'  ED 52                    SBC     HL,DE
E864'  E1                       POP     HL
E865'  28 08                    JR      Z,CURUP1        ; YES

E867'  AF                       XOR     A       ; NO, CALCULATE PREVIOUS LINE ADDRESS
E868'  11 0050                  LD      DE,MAXCOL
E86B'  ED 52                    SBC     HL,DE
E86D'  18 0E                    JR      CURUP2

E86F'  21 07FF          CURUP1: LD      HL,ENDADR       ; GET END ADDRESS
E872'  11 004F                  LD      DE,MAXCOL-1     ; CALCULATE FIRST ADDRESS OF LINE
E875'  AF                       XOR     A
E876'  ED 52                    SBC     HL,DE
E878'  ED 5B FFD4               LD      DE,(COLNO)      ; CALCULATE CURSOR ADDRESS
E87C'  19                       ADD     HL,DE

E87D'  7D               CURUP2: LD      A,L
E87E'  D3 04                    OUT     (VTCCU1),A
E880'  7C                       LD      A,H
E881'  D3 05                    OUT     (VTCCU2),A

E883'  C9               CUPRET: RET

                                ;
                                ; CURSOR DOWN
                                ;

E884'  3A FFD6          CURDWN: LD      A,(LINENO)      ; CHECK IF AT LAST LINE
E887'  FE 17                    CP      MAXLIN
E889'  28 26                    JR      Z,CDWNRT        ; YES, RETURN

E88B'  3C                       INC     A
E88C'  32 FFD6                  LD      (LINENO),A
E88F'  DB 04                    IN      A,(VTCCU1)
E891'  6F                       LD      L,A
E892'  DB 05                    IN      A,(VTCCU2)
E894'  67                       LD      H,A             ; CHECK IF END ADDRESS LINE
E895'  11 0050                  LD      DE,MAXCOL
E898'  19                       ADD     HL,DE
E899'  EB                       EX      DE,HL
E89A'  21 07FF                  LD      HL,ENDADR
E89D'  AF                       XOR     A
```

```
E89E'  ED 52                    SBC   HL,DE
E8A0'  30 09                    JR    NC,CDWN1        ; NO

E8A2'  21 0080                  LD    HL,STADR        ; CALCULATE NEXT LINE ADDRESS
E8A5'  ED 5B FFD4               LD    DE,(COLNO)
E8A9'  19                       ADD   HL,DE
E8AA'  EB                       EX    DE,HL

E8AB'  7B               CDWN1:  LD    A,E             ; SET CURSOR ADDRESS
E8AC'  D3 04                    OUT   (VTCCU1),A
E8AE'  7A                       LD    A,D
E8AF'  D3 05                    OUT   (VTCCU2),A
E8B1'  C9               CDWNRT: RET

                                ;
                                ; CURSOR HOME
                                ;

E8B2'  2A FFD7          CURHME: LD    HL,(SCSTART)    ; GET SCREEN START ADDRESS
E8B5'  7D                       LD    A,L
E8B6'  D3 04                    OUT   (VTCCU1),A      ; UPDATE CURSOR ADDRESS
E8B8'  7C                       LD    A,H
E8B9'  D3 05                    OUT   (VTCCU2),A
E8BB'  AF                       XOR   A       ; RESET LINE & COLUMN NUMBER
E8BC'  32 FFD6                  LD    (LINENO),A
E8BF'  32 FFD4                  LD    (COLNO),A
E8C2'  3E 00                    LD    A,OFF           ; TURN OFF SCROLLING
E8C4'  32 F027'                 LD    (SCRLFLG),A

E8C7'  C9               CURRET: RET

                                ;
                                ; DIRECT ADDRESSING
                                ;

E8C8'  3E 01            DIRADR: LD    A,1             ; SET COUNT FOR CUROSR ADDRESS
E8CA'  32 F030'                 LD    (CURCNT),A
E8CD'  C9                       RET           ; RETURN

E8CE'  3A F02A'         DIADR1: LD    A,(LDFLG)
E8D1'  FE FF                    CP    ON
E8D3'  20 13                    JR    NZ,DIADRA

E8D5'  79                       LD    A,C
E8D6'  D6 20                    SUB   20H
E8D8'  32 FFD6                  LD    (LINENO),A
E8DB'  FE 17                    CP    MAXLIN
E8DD'  3E FF                    LD    A,ON
E8DF'  28 02                    JR    Z,DIADRC
E8E1'  3E 00                    LD    A,OFF
E8E3'  32 F027'         DIADRC: LD    (SCRLFLG),A
E8E6'  18 0F                    JR    DIADRD

E8E8'  79               DIADRA: LD    A,C
E8E9'  FE 5F                    CP    95      ; TEST IF CONVERSION IS REQD - STRIP OFFSET
E8EB'  F4 E96B'                 CALL  P,CONVRT        ; YES  STRIP BITS 5-7 OFF TO REMOVE OFFSET
```

```
E8EE'  FE 50                    CP      MAXCOL          ; CHECK IF WITHIN LIMIT
E8F0'  38 02                    JR      C,DIADR8        ; YES

E8F2'  3E 4F                    LD      A,MAXCOL-1      ; NO, SET TO MAXIMUM

E8F4'  32 F02F'         DIADR8: LD      (COLTP),A       ; SAVE COLUMN NUMBER
E8F7'  3E 02            DIADRD: LD      A,2             ; INCREMENT COUNTER
E8F9'  32 F030'                 LD      (CURCNT),A
E8FC'  C9                       RET             ; RETURN

E8FD'  3A F02A'         DIADR2: LD      A,(LDFLG)
E900'  FE FF                    CP      ON
E902'  20 0C                    JR      NZ,DIADRB

E904'  79                       LD      A,C
E905'  D6 20                    SUB     20H
E907'  32 FFD4                  LD      (COLNO),A
E90A'  3A FFD6                  LD      A,(LINENO)
E90D'  47                       LD      B,A
E90E'  18 20                    JR      DIADRE

E910'  79               DIADRB: LD      A,C
E911'  E6 1F                    AND     1FH             ; STRIP THE OFFSET
E913'  FE 17                    CP      MAXLIN          ; CHECK IF WITHIN LIMIT
E915'  38 02                    JR      C,DIADR9        ; YES

E917'  3E 17                    LD      A,MAXLIN        ; NO, SET TO MAXIMUM

E919'  F5               DIADR9: PUSH    AF
E91A'  FE 17                    CP      MAXLIN          ; CHECK IF AT LAST LINE
E91C'  3E FF                    LD      A,ON
E91E'  28 02                    JR      Z,DIADR7        ; YES
E920'  3E 00                    LD      A,OFF           ; TURN OFF SCROLLING

E922'  32 F027'         DIADR7: LD      (SCRLFLG),A
E925'  F1                       POP     AF
E926'  32 FFD6                  LD      (LINENO),A      ; UPDATE LINE & COLUMN NUMBER
E929'  47                       LD      B,A
E92A'  3A F02F'                 LD      A,(COLTP)
E92D'  32 FFD4                  LD      (COLNO),A

E930'  11 0050          DIADRE: LD      DE,MAXCOL       ; CALCULATE CURSOR ADDRESS
E933'  21 0000                  LD      HL,0
E936'  78                       LD      A,B             ; CHECK IF AT ROW 0
E937'  B7                       OR      A
E938'  28 03                    JR      Z,DIADR6        ; YES

E93A'  19               LOOP6:  ADD     HL,DE
E93B'  10 FD                    DJNZ    LOOP6

E93D'  ED 5B FFD4       DIADR6: LD      DE,(COLNO)
E941'  19                       ADD     HL,DE

E942'  ED 5B FFD7               LD      DE,(SCSTART)    ; GET START ADDRESS
E946'  19                       ADD     HL,DE
```

```
E947'  E5                      PUSH   HL      ; CHECK IF OVER BOUNDARY
E948'  11 0800                 LD     DE,ENDADR+1
E94B'  AF                      XOR    A
E94C'  ED 52                   SBC    HL,DE
E94E'  30 03                   JR     NC,DIADR3       ; YES

E950'  E1                      POP    HL
E951'  18 05                   JR     DIADR4

E953'  D1              DIADR3: POP    DE
E954'  11 0080                 LD     DE,STADR
E957'  19                      ADD    HL,DE
E958'  7D              DIADR4: LD     A,L             ; UPDATE CURSOR
E959'  D3 04                   OUT    (VTCCU1),A
E95B'  7C                      LD     A,H
E95C'  D3 05                   OUT    (VTCCU2),A

E95E'  AF              DIADR5: XOR    A       ; CLEAR COUNT
E95F'  32 F030'                LD     (CURCNT),A
E962'  3E 00                   LD     A,OFF           ; TURN OFF LEAD-IN FLAG
E964'  32 F029'                LD     (LEADFLG),A
E967'  32 F02A'                LD     (LDFLG),A
E96A'  C9                      RET            ; RETURN

E96B'  E6 1F           CONVRT: AND    1FH             ; STRIP THE HIGH ORDER BITS - 5 TO 7
E96D'  C9                      RET

                               ;
                               ; CLEAR SCREEN
                               ;

E96E'  3E 30           CLRSCN: LD     A,OFFCUR        ; TURN OFF CURSOR
E970'  D3 01                   OUT    (VTCCMD),A
E972'  3E A0                   LD     A,BLANK+FORGND  ; SET TO FOREGROUND BLANKS
E974'  D3 1C                   OUT    (VTCDAT),A
E976'  21 0080                 LD     HL,STADR        ; SET CURSOR TO START ADDRESS
E979'  7D                      LD     A,L
E97A'  D3 04                   OUT    (VTCCU1),A
E97C'  7C                      LD     A,H
E97D'  D3 05                   OUT    (VTCCU2),A
E97F'  21 07FF                 LD     HL,ENDADR       ; SET POINTER TO END ADDRESS
E982'  7D                      LD     A,L
E983'  D3 06                   OUT    (VTCPT1),A
E985'  7C                      LD     A,H
E986'  D3 07                   OUT    (VTCPT2),A
E988'  3E BB                   LD     A,WRCP          ; CLEAR SCREEN
E98A'  CD EB2D'                CALL   WAIT

E98D'  21 0080                 LD     HL,STADR        ; SET CURSOR AND SCREEN TO START ADDRESS
E990'  22 FFD7                 LD     (SCSTART),HL    ; SAVE SCREEN START ADDRESS

E993'  3A FFDF                 LD     A,(SWITCH)      ; CHECK IF TV SWITCH ON
E996'  CB 4F                   BIT    TV,A
E998'  20 06                   JR     NZ,CLRSC1       ; YES

E99A'  7D                      LD     A,L             ; UPDATE SCREEN START ADDRESS
```

```
E99B'  D3 02                    OUT     (VTCSC1),A
E99D'  7C                       LD      A,H
E99E'  D3 03                    OUT     (VTCSC2),A

E9A0'  7D               CLRSC1: LD      A,L
E9A1'  D3 04                    OUT     (VTCCU1),A
E9A3'  7C                       LD      A,H
E9A4'  D3 05                    OUT     (VTCCU2),A
E9A6'  AF                       XOR     A       ; RESET COLUMN & LINE NUMBER
E9A7'  32 FFD4                  LD      (COLNO),A
E9AA'  32 FFD6                  LD      (LINENO),A
E9AD'  3E 00                    LD      A,OFF           ; TURN OFF SCROLLING
E9AF'  32 F027'                 LD      (SCRLFLG),A
E9B2'  C9                       RET

                                ;
                                ; CLEAR TO END OF LINE
                                ;

E9B3'  3E 30            CLREND: LD      A,OFFCUR        ; TURN OFF CURSOR
E9B5'  D3 01                    OUT     (VTCCMD),A
E9B7'  21 004F                  LD      HL,MAXCOL-1     ; CALCULATE END OF LINE ADDRESS
E9BA'  ED 5B FFD4               LD      DE,(COLNO)
E9BE'  AF                       XOR     A
E9BF'  ED 52                    SBC     HL,DE
E9C1'  DB 04                    IN      A,(VTCCU1)
E9C3'  5F                       LD      E,A
E9C4'  DB 05                    IN      A,(VTCCU2)
E9C6'  57                       LD      D,A
E9C7'  19                       ADD     HL,DE
E9C8'  7D                       LD      A,L             ; SET POINTER TO END OF LINE
E9C9'  D3 06                    OUT     (VTCPT1),A
E9CB'  7C                       LD      A,H
E9CC'  D3 07                    OUT     (VTCPT2),A      ; SET TO FOREGROUND BLANKS
E9CE'  3E A0                    LD      A,BLANK+FORGND
E9D0'  D3 1C                    OUT     (VTCDAT),A

E9D2'  CD EB28'                 CALL    WAIT2           ; CLEAR LINE

E9D5'  7B                       LD      A,E             ; RESET CUROSR ADDRESS
E9D6'  D3 04                    OUT     (VTCCU1),A
E9D8'  7A                       LD      A,D
E9D9'  D3 05                    OUT     (VTCCU2),A
E9DB'  C9                       RET

                                ;
                                ; CLEAR TO END OF SCREEN (FOREGROUND BLANKS)
                                ;

E9DC'  3E A0            CLRFSC: LD      A,BLANK+FORGND
E9DE'  18 02                    JR      CLRF1

                                ;
                                ; CLEAR TO END OF SCREEN (BACKGROUND BLANKS)
                                ;
```

```
E9E0'  3E 20           CLRBK:  LD      A,BLANK

E9E2'  D3 1C           CLRF1:  OUT     (VTCDAT),A
E9E4'  3E 30                   LD      A,OFFCUR        ; TURN OFF CURSOR
E9E6'  D3 01                   OUT     (VTCCMD),A
E9E8'  DB 04                   IN      A,(VTCCU1)      ; GET CURRENT CURSOR
E9EA'  6F                      LD      L,A
E9EB'  DB 05                   IN      A,(VTCCU2)
E9ED'  67                      LD      H,A
E9EE'  ED 5B FFD7              LD      DE,(SCSTART)    ; GET SCREEN START ADDRESS

E9F2'  E5                      PUSH    HL
E9F3'  AF                      XOR     A       ; CHECK IF CURSOR GREATER THEN START ADDR
E9F4'  ED 52                   SBC     HL,DE
E9F6'  38 1D                   JR      C,CLRF2         ; NO

E9F8'  11 07FF                 LD      DE,ENDADR       ; GET END ADDRESS
E9FB'  7B                      LD      A,E             ; SET UP POINTER
E9FC'  D3 06                   OUT     (VTCPT1),A
E9FE'  7A                      LD      A,D
E9FF'  D3 07                   OUT     (VTCPT2),A

EA01'  CD EB28'                CALL    WAIT2           ; CLEAR LINE

EA04'  2A FFD7                 LD      HL,(SCSTART)    ; GET SCREEN START ADDRESS
EA07'  11 0080                 LD      DE,STADR        ; CHECK IF SAME AS START ADDRESS
EA0A'  AF                      XOR     A
EA0B'  ED 52                   SBC     HL,DE
EA0D'  28 13                   JR      Z,CLRFRT        ; YES, RETURN

EA0F'  7B                      LD      A,E             ; SET CURSOR TO START ADDRESS
EA10'  D3 04                   OUT     (VTCCU1),A
EA12'  7A                      LD      A,D
EA13'  D3 05                   OUT     (VTCCU2),A

EA15'  2A FFD7         CLRF2:  LD      HL,(SCSTART)    ; GET SCREEN START ADDRESS
EA18'  2B                      DEC     HL      ; DECREMENT TO PREVIOUS LINE
EA19'  7D                      LD      A,L             ; SET POINTER TO END OF LINE
EA1A'  D3 06                   OUT     (VTCPT1),A
EA1C'  7C                      LD      A,H
EA1D'  D3 07                   OUT     (VTCPT2),A

EA1F'  CD EB28'                CALL    WAIT2           ; WAIT FOR VBLANK

EA22'  D1              CLRFRT: POP     DE      ; RESTORE CURSOR
EA23'  7B                      LD      A,E
EA24'  D3 04                   OUT     (VTCCU1),A
EA26'  7A                      LD      A,D
EA27'  D3 05                   OUT     (VTCCU2),A
EA29'  C9                      RET

                               ;
                               ; DELETE LINE
                               ;

EA2A'  3E 30           DELLNE: LD      A,OFFCUR        ; TURN OFF CURSOR
```

```
EA2C'  D3 01                    OUT     (VTCCMD),A
EA2E'  DB 04                    IN      A,(VTCCU1)
EA30'  6F                       LD      L,A
EA31'  DB 05                    IN      A,(VTCCU2)
EA33'  67                       LD      H,A
EA34'  ED 5B FFD4               LD      DE,(COLNO)
EA38'  AF                       XOR     A
EA39'  32 FFD4                  LD      (COLNO),A       ; UPDATE COLUMN NUMBER
EA3C'  ED 52                    SBC     HL,DE
EA3E'  E5                       PUSH    HL      ; SAVE LINE FIRST ADDRESS
EA3F'  3A FFD6                  LD      A,(LINENO)      ; GET LINE NUMBER
EA42'  32 F02E'                 LD      (LINETP),A
EA45'  7D               DEL1:   LD      A,L
EA46'  D3 04                    OUT     (VTCCU1),A
EA48'  7C                       LD      A,H
EA49'  D3 05                    OUT     (VTCCU2),A
EA4B'  3A F02E'                 LD      A,(LINETP)      ; CHECK IF AT LAST LINE
EA4E'  FE 17                    CP      MAXLIN
EA50'  28 3A                    JR      Z,DELCLR        ; YES, CLEAR LAST LINE

EA52'  3C                       INC     A
EA53'  32 F02E'                 LD      (LINETP),A      ; SAVE LINE NUMBER
EA56'  11 004F                  LD      DE,MAXCOL-1     ; CALCULATE NEXT LINE ADDRESS
EA59'  19                       ADD     HL,DE
EA5A'  EB                       EX      DE,HL
EA5B'  21 07FF                  LD      HL,ENDADR
EA5E'  AF                       XOR     A
EA5F'  ED 52                    SBC     HL,DE
EA61'  28 03                    JR      Z,DEL2

EA63'  13                       INC     DE
EA64'  18 03                    JR      DEL3

EA66'  11 0080          DEL2:   LD      DE,STADR
EA69'  7B               DEL3:   LD      A,E             ; SET POINTER TO NEXT LINE ADDRESS
EA6A'  D3 06                    OUT     (VTCPT1),A
EA6C'  7A                       LD      A,D
EA6D'  D3 07                    OUT     (VTCPT2),A
EA6F'  D5                       PUSH    DE      ; SAVE NEXT LINE ADDRESS
EA70'  06 50                    LD      B,MAXCOL
EA72'  3E A4            DEL4:   LD      A,RDPT          ; READ FROM POINTER
EA74'  CD EB2D'                 CALL    WAIT

EA77'  DB 0D                    IN      A,(VTCINP)      ; GET CHARACTER
EA79'  D3 1C                    OUT     (VTCDAT),A      ; SEND CHARACTER

EA7B'  3E AB                    LD      A,WRINC         ; WRITE AT CURSOR & INCREMENT
EA7D'  CD EB2D'                 CALL    WAIT

EA80'  13                       INC     DE      ; INCRMENT POINTER ADDRESS
EA81'  7B                       LD      A,E
EA82'  D3 06                    OUT     (VTCPT1),A
EA84'  7A                       LD      A,D
EA85'  D3 07                    OUT     (VTCPT2),A

EA87'  10 E9                    DJNZ    DEL4            ; REPEAT FOR WHOLE LINE
```

```
EA89'  E1                        POP     HL
EA8A'  18 B9                     JR      DEL1

EA8C'  11 004F            DELCLR: LD      DE,MAXCOL-1     ; SET POINTER TO END OF LINE
EA8F'  19                         ADD     HL,DE
EA90'  7D                         LD      A,L
EA91'  D3 06                      OUT     (VTCPT1),A
EA93'  7C                         LD      A,H
EA94'  D3 07                      OUT     (VTCPT2),A
EA96'  3E A0                      LD      A,BLANK+FORGND
EA98'  D3 1C                      OUT     (VTCDAT),A

EA9A'  CD EB28'                   CALL    WAIT2           ; CLEAR LINE

EA9D'  E1                         POP     HL      ; RESTORE CURSOR
EA9E'  7D                         LD      A,L
EA9F'  D3 04                      OUT     (VTCCU1),A
EAA1'  7C                         LD      A,H
EAA2'  D3 05                      OUT     (VTCCU2),A
EAA4'  C9                         RET

                                  ;
                                  ; INSERT LINE
                                  ;

EAA5'  3E 30              INSLNE: LD      A,OFFCUR        ; TURN OFF CURSOR
EAA7'  D3 01                      OUT     (VTCCMD),A
EAA9'  DB 04                      IN      A,(VTCCU1)      ; CALCULATE LINE BEGIN ADDRESS
EAAB'  6F                         LD      L,A
EAAC'  DB 05                      IN      A,(VTCCU2)
EAAE'  67                         LD      H,A
EAAF'  ED 5B FFD4                 LD      DE,(COLNO)
EAB3'  AF                         XOR     A
EAB4'  32 FFD4                    LD      (COLNO),A
EAB7'  ED 52                      SBC     HL,DE
EAB9'  E5                         PUSH    HL
EABA'  7D                         LD      A,L             ; SET CURSOR TO BEGINNING ADDRESS
EABB'  D3 04                      OUT     (VTCCU1),A
EABD'  7C                         LD      A,H
EABE'  D3 05                      OUT     (VTCCU2),A

EAC0'  06 50                      LD      B,MAXCOL
EAC2'  21 F326'                   LD      HL,BUFTP        ; SET TO TEMPORARY BUFFER

EAC5'  3E AC              INS1:   LD      A,RDCUR         ; READ AT CURSOR
EAC7'  CD EB2D'                   CALL    WAIT
EACA'  DB 0D                      IN      A,(VTCINP)      ; GET CHARACTER
EACC'  77                         LD      (HL),A          ; SAVE IN BUFFER
EACD'  23                         INC     HL
EACE'  3E A0                      LD      A,BLANK+FORGND
EAD0'  D3 1C                      OUT     (VTCDAT),A
EAD2'  3E AB                      LD      A,WRINC         ; WRITE SPACE
EAD4'  CD EB2D'                   CALL    WAIT
EAD7'  10 EC                      DJNZ    INS1            ; BLANK LINE
```

```
EAD9'  3A FFD6                  LD      A,(LINENO)
EADC'  32 F02E'                 LD      (LINETP),A
EADF'  3A F02E'          INS2:  LD      A,(LINETP)      ; CHECK IF AT LAST LINE
EAE2'  FE 17                    CP      MAXLIN
EAE4'  28 3A                    JR      Z,INSRT         ; YES, RETURN

EAE6'  3C                       INC     A
EAE7'  32 F02E'                 LD      (LINETP),A

EAEA'  DB 04                    IN      A,(VTCCU1)      ; CHECK IF END ADDRESS
EAEC'  5F                       LD      E,A
EAED'  DB 05                    IN      A,(VTCCU2)
EAEF'  57                       LD      D,A
EAF0'  1B                       DEC     DE
EAF1'  21 07FF                  LD      HL,ENDADR
EAF4'  AF                       XOR     A
EAF5'  ED 52                    SBC     HL,DE
EAF7'  28 03                    JR      Z,INS3          ; YES
EAF9'  13                       INC     DE
EAFA'  18 03                    JR      INS4
EAFC'  11 0080           INS3:  LD      DE,STADR
EAFF'  7B                INS4:  LD      A,E             ; SET NEXT LINE ADDRESS
EB00'  D3 04                    OUT     (VTCCU1),A
EB02'  7A                       LD      A,D
EB03'  D3 05                    OUT     (VTCCU2),A

EB05'  06 50                    LD      B,MAXCOL
EB07'  21 F326'                 LD      HL,BUFTP
EB0A'  3E AC             INS5:  LD      A,RDCUR         ; MOVE LINE DOWN
EB0C'  CD EB2D'                 CALL    WAIT
EB0F'  DB 0D                    IN      A,(VTCINP)
EB11'  4F                       LD      C,A
EB12'  7E                       LD      A,(HL)
EB13'  D3 1C                    OUT     (VTCDAT),A
EB15'  3E AB                    LD      A,WRINC
EB17'  CD EB2D'                 CALL    WAIT
EB1A'  71                       LD      (HL),C
EB1B'  23                       INC     HL
EB1C'  10 EC                    DJNZ    INS5
EB1E'  18 BF                    JR      INS2

EB20'  E1                INSRT: POP     HL      ; RESTORE CURSOR
EB21'  7D                       LD      A,L
EB22'  D3 04                    OUT     (VTCCU1),A
EB24'  7C                       LD      A,H
EB25'  D3 05                    OUT     (VTCCU2),A
EB27'  C9                       RET

                                ;
                                ; CLEAR LINE
                                ;

EB28'  CD EB38'          WAIT2: CALL    WAIT1           ; WAIT FOR VBLANK
EB2B'  3E BB                    LD      A,WRCP          ; CLEAR LINE

                                ;
```

```
                                ; WAIT FOR END OF PVTC COMMAND
                                ;       A = PVTC COMMAND
                                ;

EB2D'  F3              WAIT:    DI              ; DISABLE INTERRUPT
EB2E'  D3 01                    OUT     (VTCCMD),A      ; SEND COMMAND
EB30'  DB 01           LOOP9:   IN      A,(VTCSTA)      ; GET STATUS
EB32'  CB 6F                    BIT     RDYFLG,A        ; CHECK READY FLAG
EB34'  28 FA                    JR      Z,LOOP9         ; LOOP AND WAIT
EB36'  FB                       EI              ; ENABLE INTERRUPT
EB37'  C9                       RET             ; RETURN

                                ;
                                ; WAIT FOR VBLANK
                                ;

EB38'  3A FFDF         WAIT1:   LD      A,(SWITCH)      ; CHECK IF TV SWITCH ON
EB3B'  CB 4F                    BIT     TV,A
EB3D'  28 0E                    JR      Z,WAITNO        ; NO

EB3F'  3E 00                    LD      A,0FF           ; CLEAR VBLANK FLAG
EB41'  32 FFDD                  LD      (VBFLAG),A

EB44'  3A FFDD         LOOP10:  LD      A,(VBFLAG)      ; GET VBLANK FLAG
EB47'  FE FF                    CP      0N      ; CHECK IF VBLANK PERIOD
EB49'  20 F9                    JR      NZ,LOOP10       ; NO, WAIT TILL VBLANK
EB4B'  18 0C                    JR      WAITRT

EB4D'  3E 40           WAITNO:  LD      A,RSTFLG        ; RESET VBLANK FLAG
EB4F'  CB E7                    SET     VBLANK,A
EB51'  D3 01                    OUT     (VTCCMD),A

EB53'  DB 01           LOOP11:  IN      A,(VTCSTA)      ; GET STATUS
EB55'  CB 67                    BIT     VBLANK,A        ; CHECK IF VBLANK
EB57'  28 FA                    JR      Z,LOOP11        ; NO, WAIT

EB59'  C9              WAITRT:  RET             ; RETURN

                                ;
                                ; DUMMY ROUTINE
                                ;

EB5A'  C9              NONE:    RET

                                ;
                                ;       PUT LIST CHARACTER
                                ;          C = CHARACTER
                                ;

EB5B'  CD EB94'        LIST:    CALL    LISTST          ; CHECK STATUS
EB5E'  B7                       OR      A
EB5F'  28 FA                    JR      Z,LIST          ; WAIT FOR READY

EB61'  CD EB85'                 CALL    LISTOT

EB64'  3E 0D                    LD      A,CR
```

```
EB66'  B9                      CP      C
EB67'  28 0A                   JR      Z,LIST1

EB69'  3E 0C                   LD      A,FMFEED
EB6B'  B9                      CP      C
EB6C'  28 05                   JR      Z,LIST1

EB6E'  3E 0A                   LD      A,LF
EB70'  B9                      CP      C
EB71'  20 11                   JR      NZ,LISTRT

EB73'  C5              LIST1:  PUSH    BC
EB74'  06 0A                   LD      B,PADDING
EB76'  CD EB94'        LOOP20: CALL    LISTST
EB79'  B7                      OR      A
EB7A'  28 FA                   JR      Z,LOOP20
EB7C'  0E 00                   LD      C,0
EB7E'  CD EB85'                CALL    LISTOT
EB81'  10 F3                   DJNZ    LOOP20
EB83'  C1                      POP     BC
EB84'  C9              LISTRT: RET

EB85'  79              LISTOT: LD      A,C
EB86'  D3 0D                   OUT     (PTRDAT),A      ; SEND DATA TO DATA REGISTER
EB88'  3A FFE2                 LD      A,(IOREG)       ; GET PREVIOUS I/O REGISTER VALUE
EB8B'  CB DF                   SET     PTRSTR,A        ; SET STROBE
EB8D'  D3 18                   OUT     (CONTRL),A      ; SEND TO CONTROL REGISTER
EB8F'  CB 9F                   RES     PTRSTR,A        ; RESET STROBE
EB91'  D3 18                   OUT     (CONTRL),A      ; SEND TO CONTROL REGISTER
EB93'  C9                      RET

                               ;
                               ;       RETURN LIST STATUS
                               ;          A = 0FFH, READY
                               ;            = 000H, NOT READY
                               ;

EB94'  C5              LISTST: PUSH    BC
EB95'  DB 18                   IN      A,(STATUS)      ; CHECK IF PRINTER BUSY
EB97'  CB 6F                   BIT     PTRBSY,A
EB99'  28 37                   JR      Z,LISTS3

EB9B'  3A F034'                LD      A,(TIMEOUT)     ; CHECK IF TIMEOUT
EB9E'  FE FF                   CP      ON
EBA0'  28 2C                   JR      Z,LISTS2        ; YES

EBA2'  ED 4B F576'             LD      BC,(TIMER)      ; DECREMENT TIMER
EBA6'  0B                      DEC     BC
EBA7'  ED 43 F576'             LD      (TIMER),BC
EBAB'  78                      LD      A,B
EBAC'  B1                      OR      C
EBAD'  20 1C                   JR      NZ,LISTS1

EBAF'  21 EF97'                LD      HL,ERMSG3
EBB2'  CD E3D1'                CALL    PMSG
```

```
EBB5'  CD E4E4'          LOOP17: CALL   CONIN          ; PROMPT FOR RESPONSE
EBB8'  FE 0D                     CP     CR
EBBA'  20 F9                     JR     NZ,LOOP17

EBBC'  0E 0D                     LD     C,CR          ; SEND CR & LF
EBBE'  CD E59D'                  CALL   CONOUT
EBC1'  0E 0A                     LD     C,LF
EBC3'  CD E59D'                  CALL   CONOUT
EBC6'  3E FF                     LD     A,ON          ; TURN ON TIMEOUT
EBC8'  32 F034'                  LD     (TIMEOUT),A

EBCB'  AF                LISTS1: XOR    A       ; NOT READY
EBCC'  18 12                     JR     LSTRET

EBCE'  3E FF             LISTS2: LD     A,0FFH        ; TIMEOUT ALREADY
EBD0'  18 0E                     JR     LSTRET

EBD2'  3E 00             LISTS3: LD     A,0FF         ; READY
EBD4'  32 F034'                  LD     (TIMEOUT),A
EBD7'  01 FFFF                   LD     BC,TIMING
EBDA'  ED 43 F576'               LD     (TIMER),BC
EBDE'  3E FF                     LD     A,0FFH
EBE0'  C1                LSTRET: POP    BC
EBE1'  C9                        RET

                                 ;
                                 ;      PUT PUNCH CHARACTER
                                 ;         C = CHARACTER
                                 ;

EBE2'  C9                PUNCH:  RET

EBE3'  00 00                     DB     0,0           ; SPARE

                                 ;
                                 ;      GET READER CHARACTER
                                 ;         A = CHARACTER
                                 ;

EBE5'  C9                READER: RET

EBE6'  00 00                     DB     0,0           ; SPARE

                                 ;
                                 ;      I/O DRIVERS FOR THE DISK
                                 ;

                                 ;
                                 ;      HOME DISK
                                 ;

EBE8'  01 0000           HOME:   LD     BC,0

                                 ;
                                 ;      SET TRACK NUMBER
                                 ;         BC = LOGICAL TRACK NUMBER
```

```
                                ;
EBEB'  ED 43 F09A'      SETTRK: LD      (SEKTRK),BC
EBEF'  C9                       RET
                                ;
                                ;       SELECT DISK
                                ;          C = DISK NUMBER
                                ;          E = 0, DISK OFFLINE
                                ;
EBF0'  79               SELDSK: LD      A,C             ; GET DISK SELECTED
EBF1'  FE 03                    CP      MAXDSK          ; CHECK IF OVER LIMIT
EBF3'  30 35                    JR      NC,SELBAD       ; YES, BAD SECLECTION

EBF5'  21 E233'                 LD      HL,DSKMAP       ; GET PHYSCAL DISK NUMBER
EBF8'  06 00                    LD      B,0
EBFA'  09                       ADD     HL,BC
EBFB'  7E                       LD      A,(HL)
EBFC'  FE FF                    CP      0FFH            ; CHECK IF DISK EXISTS
EBFE'  28 2A                    JR      Z,SELBAD        ; NO, BAD SELECTION

EC00'  32 F099'                 LD      (SEKDSK),A      ; SAVE DISK NUMBER
EC03'  CD EC39'                 CALL    GETPARM         ; GET DISK PARAMETERS

                                ;
                                ;       RESTORE DISK DRIVE
                                ;

EC06'  7B               SELHME: LD      A,E             ; GET OFF-LINE FLAG
EC07'  B7                       OR      A       ; CHECK IF OFF-LINE
EC08'  20 1A                    JR      NZ,SELOK        ; NO

EC0A'  3A F0A7'                 LD      A,(BUFACT)
EC0D'  FE FF                    CP      ON
EC0F'  28 0E                    JR      Z,SELHM1

EC11'  C5                       PUSH    BC
EC12'  D5                       PUSH    DE
EC13'  21 F09D'                 LD      HL,WRTYPE
EC16'  06 11                    LD      B,17
EC18'  AF                       XOR     A
EC19'  77               LOOP12: LD      (HL),A
EC1A'  23                       INC     HL
EC1B'  10 FC                    DJNZ    LOOP12
EC1D'  D1                       POP     DE
EC1E'  C1                       POP     BC

EC1F'  CD EDEC'         SELHM1: CALL    RSTORE          ; RESTORE DRIVE IF OFF-LINE
EC22'  38 0C                    JR      C,SELRST        ; RESTORE ERROR

                                ;
                                ;       RETURN DISK PARAMETER HEADER ADDRESS
                                ;

EC24'  2A F0C0'         SELOK:  LD      HL,(DPADDR)     ; RETURN DPH ADDR IN HL
```

```
EC27'  AF                      XOR   A
EC28'  18 0E                   JR    SELRET

EC2A'  21 EFCA'          SELBAD: LD    HL,ERMSG4
EC2D'  CD E3D1'                  CALL  PMSG
EC30'  AF                SELRST: XOR   A       ; RESET CURRENT DISK TO A
EC31'  32 0004                   LD    (CDISK),A
EC34'  21 0000                   LD    HL,0            ; SET ERROR RETURN
EC37'  37                        SCF

EC38'  C9                SELRET: RET

                                 ;
                                 ;       GET DISK PARAMETERS
                                 ;          IN:          A = PHYSICAL DISK NUMBER
                                 ;          OUT:         NONE
                                 ;          CHANGED:     NONE
                                 ;

EC39'                    GETPARM:
EC39'  F5                        PUSH  AF
EC3A'  C5                        PUSH  BC
EC3B'  D5                        PUSH  DE
EC3C'  E5                        PUSH  HL
EC3D'  CB 27                     SLA   A       ; TIMES 2
EC3F'  CB 27                     SLA   A       ; TIMES 4
EC41'  4F                        LD    C,A             ; SAVE FOR LATER
EC42'  CB 27                     SLA   A       ; TIMES 8
EC44'  CB 27                     SLA   A       ; TIMES 16
EC46'  6F                        LD    L,A
EC47'  26 00                     LD    H,0
EC49'  06 00                     LD    B,0
EC4B'  09                        ADD   HL,BC           ; TIMES 20
EC4C'  01 E237'                  LD    BC,DPBASE
EC4F'  09                        ADD   HL,BC           ; HL=DPBASE(DISKNO*20)
EC50'  22 F0C0'                  LD    (DPADDR),HL     ; SAVE THE DPH ADDR

                                 ;
                                 ;       MOVE DISK TABLE INFORMATION
                                 ;

EC53'  01 0010                   LD    BC,16           ; CALCULATE OFFSET TO DISK TABLE ADDRESS
EC56'  09                        ADD   HL,BC
EC57'  5E                        LD    E,(HL)          ; GET DISK TABLE ADDRESS
EC58'  23                        INC   HL
EC59'  56                        LD    D,(HL)
EC5A'  ED 53 F0BE'               LD    (DSKTAB),DE     ; SAVE DISK TABLE ADDRESS
EC5E'  01 0006                   LD    BC,6            ; SET LENGTH OF MOVE
EC61'  21 F0AE'                  LD    HL,SELBYT       ; MOVE INTO DATA AREA
EC64'  EB                        EX    DE,HL
EC65'  ED B0                     LDIR          ; MOVE DISK TABLE INFORMATION
EC67'  22 F0BC'                  LD    (LSTADR),HL     ; SAVE LAST TRACK ADDRESS

                                 ;
                                 ;       MOVE TAB INFORMATION
                                 ;
```

```
EC6A' 2A F0C0'              LD    HL,(DPADDR)    ; GET DP HEADER ADDRESS
EC6D' 01 0012               LD    BC,18          ; CALCULATE OFFSET TO TAB ADDRESS
EC70' 09                    ADD   HL,BC
EC71' 5E                    LD    E,(HL)         ; GET TAB ADDRESS
EC72' 23                    INC   HL
EC73' 56                    LD    D,(HL)
EC74' 01 0008               LD    BC,8           ; SET LENGTH OF MOVE
EC77' 21 F0B4'              LD    HL,HSTSIZ      ; MOVE INTO DATA AREA
EC7A' EB                    EX    DE,HL
EC7B' ED B0                 LDIR          ; MOVE TAB INFORMATION
EC7D' E1                    POP   HL
EC7E' D1                    POP   DE
EC7F' C1                    POP   BC
EC80' F1                    POP   AF
EC81' C9                    RET

                            ;
                            ;       SET SECTOR NUMBER
                            ;          C = SECTOR NUMBER
                            ;

EC82' 79                SETSEC: LD    A,C
EC83' 32 F09C'                  LD    (SEKSEC),A
EC86' C9                        RET

                            ;
                            ;       TRANSLATE SECTOR NUMBER
                            ;          BC = LOGICAL SECTOR NUMBER
                            ;          DE = TRANSLATION TABLE ADDRESS
                            ;          HL = PHYSICAL SECTOR NUMBER

EC87'                   SECTRAN:
EC87' F5                        PUSH  AF
EC88' 7A                        LD    A,D
EC89' B3                        OR    E
EC8A' 28 08                     JR    Z,SECTN1       ; NO TRANSLATION

EC8C' EB                        EX    DE,HL
EC8D' 09                        ADD   HL,BC          ; INDEX INTO TRANSLATION TABLE
EC8E' 23                        INC   HL      ; OFFSET BY 1
EC8F' 6E                        LD    L,(HL)         ; GET SECTOR NUMBER
EC90' 26 00                     LD    H,0
EC92' 18 03                     JR    SECTRT         ; RETURN

EC94' C5                SECTN1: PUSH  BC
EC95' E1                        POP   HL
EC96' 23                        INC   HL

EC97' F1                SECTRT: POP   AF
EC98' C9                        RET

                            ;
                            ;       SET DMA ADDRESS
                            ;          BC = DMA ADDRESS
                            ;
```

```
EC99'  69                  SETDMA: LD      L,C
EC9A'  60                          LD      H,B
EC9B'  22 F0AB'                    LD      (DMAAD),HL      ; SAVE DMA ADDRESS
EC9E'  C9                          RET

                                   ;
                                   ;       READ/WRITE ROUTINE
                                   ;          C = WRITE TYPE
                                   ;

                                   ;
                                   ;       WRITE A SECTOR
                                   ;

EC9F'                      WRDISK:
EC9F'  ED 73 F582'                 LD      (DOSSTK),SP     ; USE LOCAL STACK
ECA3'  31 F5C0'                    LD      SP,LOCSTK
ECA6'  79                          LD      A,C             ; GET WRITE TYPE
ECA7'  32 F09D'                    LD      (WRTYPE),A      ; SAVE TYPE
ECAA'  3E 01                       LD      A,WRITE         ; SET WRITE OPERATION
ECAC'  32 F09E'                    LD      (RDWROP),A      ; SAVE OPERATION
ECAF'  18 0C                       JR      RWOPER

                                   ;
                                   ;       READ A SECTOR
                                   ;

ECB1'                      RDDISK:
ECB1'  ED 73 F582'                 LD      (DOSSTK),SP
ECB5'  31 F5C0'                    LD      SP,LOCSTK
ECB8'  3E 00                       LD      A,READ          ; SET READ OPERATION
ECBA'  32 F09E'                    LD      (RDWROP),A      ; SAVE OPERATION

ECBD'                      RWOPER:

                                   ;
                                   ; COMPUTE PHYSICAL SECTOR, TRACK IS THE SAME
                                   ;

ECBD'  AF                          XOR     A
ECBE'  32 F0AA'                    LD      (ERFLAG),A      ; CLEAR ERROR FLAG

                                   ;
                                   ; CALCULATE PHYSICAL SECTOR NUMBER
                                   ;

ECC1'  3A F0BB'                    LD      A,(SECSHF)      ; LOG2 OF BLOCKING FACTOR
ECC4'  47                          LD      B,A
ECC5'  3A F09C'                    LD      A,(SEKSEC)      ; GET LOGICAL SECTOR NUMBER
ECC8'  3D                          DEC     A
ECC9'  04                          INC     B
ECCA'  05                  RW1:    DEC     B
ECCB'  28 05                       JR      Z,RW2
ECCD'  37                          SCF
ECCE'  3F                          CCF
```

```
ECCF' 1F                        RRA             ; DIVIDE LOGICAL SECTOR BY 2
ECD0' 18 F8                     JR      RW1
ECD2' 3C                RW2:    INC     A
ECD3' 32 F0A8'                  LD      (HSTSEC),A      ; SAVE PHYSICAL SECTOR NUMBER
ECD6' 3A F09C'                  LD      A,(SEKSEC)
ECD9' 3D                        DEC     A
ECDA' 47                        LD      B,A
ECDB' 3A F0BA'                  LD      A,(SECMSK)      ; GET SECTOR MASK
ECDE' A0                        AND     B
ECDF' 32 F0A9'                  LD      (LOGOFF),A      ; SAVE LOGICAL OFFSET OF WITHIN SECTOR
ECE2' 3A F0A3'                  LD      A,(BUFDSK)      ; GET DISK FOR HOST BUFFER
ECE5' 47                        LD      B,A
ECE6' 3A F099'                  LD      A,(SEKDSK)      ; GET DISK FOR R/W
ECE9' B8                        CP      B       ; CHECK IF SAME
ECEA' 20 16                     JR      NZ,DIFBUF       ; NO, DIFFERENT BUFFER

ECEC' ED 4B F0A4'               LD      BC,(BUFTRK)     ; GET TRACK FOR HOST BUFFER
ECF0' 2A F09A'                  LD      HL,(SEKTRK)     ; GET TRACK FOR R/W
ECF3' AF                        XOR     A
ECF4' ED 42                     SBC     HL,BC           ; CHECK IF SAME TRACK
ECF6' 20 0A                     JR      NZ,DIFBUF       ; NO, DIFFERENT BUFFER

ECF8' 3A F0A6'                  LD      A,(BUFSEC)      ; GET SECTOR FOR BUFFER
ECFB' 47                        LD      B,A
ECFC' 3A F0A8'                  LD      A,(HSTSEC)      ; GET SECTOR FOR R/W
ECFF' B8                        CP      B       ; CHECK IF SAME SECTOR
ED00' 28 1C                     JR      Z,RWLAST        ; SAME

                                ;
                                ;       DIFFERENT BUFFER
                                ;

ED02' 3A F0A7'          DIFBUF: LD      A,(BUFACT)      ; CHECK IF BUFFER ACTIVE
ED05' FE FF                     CP      ON
ED07' 3E 00                     LD      A,OFF
ED09' 32 F0A7'                  LD      (BUFACT),A      ; SET BUFFER INACTIVE
ED0C' CC EDA5'                  CALL    Z,WRHST         ; WRITE HOST WHEN ACTIVE

ED0F' 3A F09E'                  LD      A,(RDWROP)      ; GET OPERATION CODE
ED12' FE 00                     CP      READ            ; CHECK IF READ OPERATION
ED14' 28 05                     JR      Z,OPREAD        ; YES, READ SECTOR

ED16' 3A F09D'                  LD      A,(WRTYPE)      ; CHECK IF WRITE TO ALLOCATED SECTOR
ED19' FE 00                     CP      WRALL
ED1B' CC ED78'          OPREAD: CALL    Z,RDHST         ; YES, READ HOST

ED1E' CD ED3F'          RWLAST: CALL    TRANS           ; TRANSFER CONTENT

ED21' 3A F09E'                  LD      A,(RDWROP)      ; GET OPERATION CODE
ED24' FE 00                     CP      READ            ; CHECK IF READ
ED26' 28 0F                     JR      Z,RWRET         ; YES, RETURN

ED28' 3A F09D'                  LD      A,(WRTYPE)      ; GET WRITE TYPE
ED2B' FE 01                     CP      WRDIR           ; CHECK IF DIRECTORY ACTIVITY
ED2D' 20 08                     JR      NZ,RWRET        ; NO, RETURN
```

```
ED2F'  CD EDA5'                CALL   WRHST          ; WRITE HOST
ED32'  3E 00                   LD     A,OFF          ; SET BUFFER TO INACTIVE
ED34'  32 F0A7'                LD     (BUFACT),A

ED37'  3A F0AA'          RWRET: LD     A,(ERFLAG)     ; RETURN ERROR CODE
ED3A'  ED 7B F582'             LD     SP,(DOSSTK)
ED3E'  C9                      RET

                               ;
                               ;      TRANSFER CONTENT OF BUFFER
                               ;

ED3F'  AF                TRANS: XOR    A       ; CLEAR CARRY
ED40'  3A F0A9'                LD     A,(LOGOFF)     ; GET OFFSET IN SECTORS
                                      ; CALCULATE OFFSET VALUE WITHIN BUFFER
                                      ; USING LOGICAL RECORD LENGTH OF 128
ED43'  1F                      RRA           ; ROTATE RIGHT, INTO CARRY
ED44'  67                      LD     H,A
ED45'  3E 00                   LD     A,0
ED47'  1F                      RRA           ; ROTATE CARRY INTO HIGH BIT
ED48'  6F                      LD     L,A            ; GET OFFSET VALUE
ED49'  11 F376'                LD     DE,HSTBUF      ; GET BUFFER ADDRESS
ED4C'  19                      ADD    HL,DE          ; CALCULATE LOGICAL RECORD ADDRESS IN BUFFER
ED4D'  3A F0A8'                LD     A,(HSTSEC)     ; UPDATE BUFFER SECTOR NUMBER
ED50'  32 F0A6'                LD     (BUFSEC),A
ED53'  3A F099'                LD     A,(SEKDSK)     ; UPDATE BUFFER DISK NUMBER
ED56'  32 F0A3'                LD     (BUFDSK),A
ED59'  ED 5B F09A'             LD     DE,(SEKTRK)    ; UPDATE BUFFER TRACK NUMBER
ED5D'  ED 53 F0A4'             LD     (BUFTRK),DE
ED61'  ED 5B F0AB'             LD     DE,(DMAAD)     ; GET TRANSFER ADDRESS
ED65'  01 0080                 LD     BC,RECLEN      ; GET LENGTH OF TRANSFER
ED68'  3A F09E'                LD     A,(RDWROP)     ; GET OPERATION CODE
ED6B'  FE 00                   CP     READ           ; CHECK IF READ
ED6D'  28 06                   JR     Z,TRAN1        ; YES

ED6F'  3E FF                   LD     A,ON           ; SET BUFFER ACTIVE
ED71'  32 F0A7'                LD     (BUFACT),A
ED74'  EB                      EX     DE,HL          ; EXCHANGE SOURCE AND DESTINATION
ED75'  ED B0             TRAN1: LDIR          ; MOVE LOGICAL RECORD
ED77'  C9                      RET

                               ;
                               ;      READ A PHYSICAL SECTOR
                               ;         CARRY ON = ERROR RETURN
                               ;         CHANGED:     A
                               ;

ED78'  E5                RDHST: PUSH   HL
ED79'  3A F099'                LD     A,(SEKDSK)     ; GET DISK NUMBER
ED7C'  CD EC39'                CALL   GETPARM        ; GET PARAMETERS

ED7F'  2A F09A'                LD     HL,(SEKTRK)    ; GET LOGICAL TRACK NUMBER
ED82'  22 F09F'                LD     (RWTRK),HL

ED85'  3A F0A8'                LD     A,(HSTSEC)     ; GET PHYSICAL SECTOR NUMBER
ED88'  32 F0A1'                LD     (RWSEC),A
```

```
ED8B'  CD EE1F'              CALL   SEEK          ; SEEK TRACK
ED8E'  38 43                 JR     C,HSTERR      ; ERROR

ED90'  3A F0B3'              LD     A,(DSKTYP)    ; GET DISK TYPE
ED93'  CB 47                 BIT    DISKTY,A      ; CHECK IF HARD DISK
ED95'  20 07                 JR     NZ,RDHST1     ; READ FROM HARD DISK

ED97'  CD EED7'              CALL   RDFDSK        ; READ FROM FLOPPY DISK
ED9A'  38 37                 JR     C,HSTERR      ; ERROR
ED9C'  18 32                 JR     HSTNOR        ; NORMAL RETURN

ED9E'  CD EF2A'    RDHST1:   CALL   RDHDSK        ; READ FROM HARD DISK
EDA1'  38 30                 JR     C,HSTERR      ; ERROR
EDA3'  18 2B                 JR     HSTNOR        ; NORMAL RETURN

                             ;
                             ;      WRITE A PHYSICAL SECTOR
                             ;         CARRY ON = ERROR RETURN
                             ;

EDA5'  E5          WRHST:    PUSH   HL
EDA6'  3A F0A3'              LD     A,(BUFDSK)    ; GET DISK NUMBER
EDA9'  CD EC39'              CALL   GETPARM       ; GET PARAMETERS

EDAC'  2A F0A4'              LD     HL,(BUFTRK)   ; GET LOGICAL TRACK NUMBER
EDAF'  22 F09F'              LD     (RWTRK),HL

EDB2'  3A F0A6'              LD     A,(BUFSEC)    ; GET PHYSICAL SECTOR NUMBER
EDB5'  32 F0A1'              LD     (RWSEC),A

EDB8'  CD EE1F'              CALL   SEEK          ; SEEK TRACK
EDBB'  38 16                 JR     C,HSTERR      ; ERROR

EDBD'  3A F0B3'              LD     A,(DSKTYP)    ; GET DISK TYPE
EDC0'  CB 47                 BIT    DISKTY,A      ; CHECK IF HARD DISK
EDC2'  20 07                 JR     NZ,WRHST1     ; READ FROM HARD DISK

EDC4'  CD EEDB'              CALL   WRFDSK        ; READ FROM FLOPPY DISK
EDC7'  38 0A                 JR     C,HSTERR      ; ERROR
EDC9'  18 05                 JR     HSTNOR        ; NORMAL RETURN

EDCB'  CD EF2E'    WRHST1:   CALL   WRHDSK        ; READ FROM HARD DISK
EDCE'  38 03                 JR     C,HSTERR      ; ERROR

EDD0'  AF          HSTNOR:   XOR    A
EDD1'  18 01                 JR     HSTRET

EDD3'  37          HSTERR:   SCF

EDD4'  E1          HSTRET:   POP    HL
EDD5'  C9                    RET

                             ;
                             ;      DISK DRIVE SELECT
                             ;         IN:         A = SELECT BYTE
```

```
                                ;       OUT:        NONE
                                ;       CHANGED:    NONE
                                ;

EDD6'  F5              DSKSEL: PUSH   AF      ; SAVE SELECT BYTE
EDD7'  3A F0B3'                LD     A,(DSKTYP)      ; GET DISK TYPE
EDDA'  CB 47                   BIT    DISKTY,A        ; CHECK IF HARD DISK
EDDC'  20 06                   JR     NZ,DSKSL1       ; YES

EDDE'  F1                      POP    AF      ; GET SELECT BYTE
EDDF'  CD FFF6                 CALL   FDSEL           ; SELECT DRIVE
EDE2'  18 07                   JR     DSKRET          ; RETURN

EDE4'  3E 20           DSKSL1: LD     A,PRECMP        ; GET PRECOMPENSATION
EDE6'  D3 31                   OUT    (HDPRCM),A      ; WRITE PRECOMPENSATION
EDE8'  F1                      POP    AF      ; GET SELECT BYTE
EDE9'  D3 36                   OUT    (HDSEL),A               ; SELECT HARD DISK

EDEB'  C9              DSKRET: RET

                                ;
                                ;       RESTORE DISK DRIVE
                                ;          CARRY ON = ERROR RETURN
                                ;          CHANGED:     A
                                ;

EDEC'  E5              RSTORE: PUSH   HL
EDED'  3A F0AE'                LD     A,(SELBYT)      ; GET SELECT BYTE
EDF0'  CD EDD6'                CALL   DSKSEL          ; SELECT DISK DRIVE

EDF3'  3A F0B3'                LD     A,(DSKTYP)      ; GET DISK TYPE
EDF6'  CB 47                   BIT    DISKTY,A        ; CHECK IF HARD DISK
EDF8'  20 0A                   JR     NZ,RSTOR1       ; YES

EDFA'  3A F0B1'        RSTOR0: LD     A,(STEPS)       ; GET STEPPING RATE
EDFD'  F6 0C                   OR     HOMEFD          ; PUT INTO COMMAND
EDFF'  CD FFF9                 CALL   FDHCMD          ; SEND COMMAND
EE02'  18 08                   JR     RSTOR2

EE04'  3A F0B1'        RSTOR1: LD     A,(STEPS)       ; GET STEPPING RATE
EE07'  F6 0C                   OR     HOMEFD          ; PUT INTO COMMAND
EE09'  CD FFF0                 CALL   HDHCMD          ; SEND COMMAND
EE0C'  E6 90           RSTOR2: AND    SKERR           ; CHECK IF ERROR
EE0E'  32 F0AA'                LD     (ERFLAG),A      ; SAVE STATUS
EE11'  20 09                   JR     NZ,RSTERR       ; ERROR

EE13'  2A F0BC'        RSTNOR: LD     HL,(LSTADR)     ; GET LAST TRACK ADDRESS
EE16'  AF                      XOR    A
EE17'  77                      LD     (HL),A          ; SET LAST TRACK TO 0
EE18'  23                      INC    HL
EE19'  77                      LD     (HL),A
EE1A'  18 01                   JR     RSTRET

EE1C'  37              RSTERR: SCF

EE1D'  E1              RSTRET: POP    HL
```

```
EE1E'  C9                      RET

                               ;
                               ;       SEEK TRACK
                               ;          CARRY ON = ERROR RETURN
                               ;          CHANGED:     A
                               ;

EE1F'  C5              SEEK:   PUSH    BC
EE20'  D5                      PUSH    DE
EE21'  E5                      PUSH    HL
EE22'  CD EE92'                CALL    TRNTRK          ; TRANSLATE LOGICAL TRACK TO PHYSICAL TRACK

EE25'  3A F0B3'                LD      A,(DSKTYP)      ; GET DISK TYPE
EE28'  CB 47                   BIT     DISKTY,A        ; CHECK IF HARD DISK
EE2A'  20 52                   JR      NZ,SEEKNO       ; YES, NO SEEK

                               ;
                               ;       SELECT DISK AND SIDE
                               ;

EE2C'  3A F0AE'        SEEK0:  LD      A,(SELBYT)      ; GET SELECT BYTE
EE2F'  47                      LD      B,A
EE30'  3A F0A2'                LD      A,(RWSIDE)      ; GET SIDE NUMBER
EE33'  B7                      OR      A       ; CHECK IF SIDE 0
EE34'  78                      LD      A,B
EE35'  28 02                   JR      Z,SEEK1         ; YES
EE37'  CB D7                   SET     SELSID,A        ; NO, SET SIDE 1
EE39'  CD EDD6'        SEEK1:  CALL    DSKSEL          ; SELECT DRIVE

                               ;
                               ;       CHECK IF SAME PHYSICAL TRACK
                               ;

EE3C'  2A F0BC'                LD      HL,(LSTADR)     ; GET LAST TRACK ADDRESS
EE3F'  5E                      LD      E,(HL)          ; GET LAST PHYSICAL TRACK
EE40'  23                      INC     HL
EE41'  56                      LD      D,(HL)
EE42'  7B                      LD      A,E             ; SET UP TRACK REGISTER
EE43'  D3 11                   OUT     (FDTRK),A
EE45'  2A F09F'                LD      HL,(RWTRK)      ; GET CURRENT SEEK TRACK
EE48'  AF                      XOR     A       ; CHECK IF SAME TRACK
EE49'  ED 52                   SBC     HL,DE
EE4B'  28 31                   JR      Z,SEEKNO        ; YES, RETURN

                               ;
                               ;       SEEK TRACK
                               ;

EE4D'  06 02                   LD      B,SEKRTY        ; GET NUMBER OF SEEK RETRY
EE4F'  2A F09F'        LOOP4:  LD      HL,(RWTRK)      ; GET SEEK TRACK
EE52'  7D                      LD      A,L             ; SEND TO CONTROLLER
EE53'  D3 13                   OUT     (FDDATA),A
EE55'  2E 1C                   LD      L,SEEKFD        ; GET SEEK COMMAND
EE57'  3A F0B1'                LD      A,(STEPS)       ; GET STEPPING RATE
EE5A'  B5                      OR      L       ; PUT INTO COMMAND
```

```
EE5B'  CD FFF9                  CALL   FDHCMD          ; SEEK TRACK
EE5E'  E6 90                    AND    SKERR           ; CHECK IF ERROR
EE60'  32 F0AA'                 LD     (ERFLAG),A      ; SAVE STATUS
EE63'  28 0F                    JR     Z,SEEKOK        ; NO

EE65'  E5                       PUSH   HL
EE66'  CD EDFA'                 CALL   RSTOR0          ; ERROR, RESTORE DRIVE
EE69'  38 16                    JR     C,SEEKER        ; RESTORE ERROR
EE6B'  10 E2                    DJNZ   LOOP4           ; RETRY
EE6D'  3E FF                    LD     A,0FFH          ; SET ERROR FLAG
EE6F'  32 F0AA'                 LD     (ERFLAG),A
EE72'  18 0D                    JR     SEEKER          ; ERROR

EE74'  2A F0BC'          SEEKOK: LD     HL,(LSTADR)     ; GET LAST TRACK ADDRESS
EE77'  ED 5B F09F'              LD     DE,(RWTRK)      ; GET CURRENT TRACK
EE7B'  73                       LD     (HL),E          ; UPDATE LAST TRACK
EE7C'  23                       INC    HL
EE7D'  72                       LD     (HL),D
EE7E'  AF                SEEKNO: XOR    A
EE7F'  18 07                    JR     SEEKRT

EE81'  21 EF81'          SEEKER: LD     HL,ERMSG2       ; DISPLAY MESSAGE
EE84'  CD E3D1'                 CALL   PMSG
EE87'  37                       SCF
EE88'  E1                SEEKRT: POP    HL
EE89'  D1                       POP    DE
EE8A'  C1                       POP    BC
EE8B'  C9                       RET

                                ;
                                ;       TRANSLATE LOGICAL TRACK TO PHYSICAL TRACK
                                ;

EE8C'  C5                SEEK2:  PUSH   BC
EE8D'  D5                        PUSH   DE
EE8E'  E5                        PUSH   HL
EE8F'  C3 EE2C'                  JP     SEEK0

EE92'  F5                TRNTRK: PUSH   AF
EE93'  C5                        PUSH   BC
EE94'  D5                        PUSH   DE
EE95'  E5                        PUSH   HL
EE96'  3A F0B2'                  LD     A,(SIDES)       ; GET THE NUMBER OF SIDES
EE99'  FE 01                     CP     1       ; CHECK IF SINGLE SIDED
EE9B'  3E 00                     LD     A,0             ; SET TO SIDE 0
EE9D'  28 30                     JR     Z,TRNRET        ; RETRUN

EE9F'  47                        LD     B,A             ; CLEAR COUNTER
EEA0'  3A F0B2'                  LD     A,(SIDES)       ; GET THE NUMBER OF SIDES
EEA3'  37                        SCF
EEA4'  3F                        CCF
EEA5'  CB 1F             LOOP1:  RR     A       ; CALCULATE LOG2 OF SIDES
EEA7'  28 03                     JR     Z,TRN1          ; LOOP ENDS
EEA9'  04                        INC    B       ; INCREMENT COUNTER
EEAA'  18 F9                     JR     LOOP1
```

```
EEAC'  C5                TRN1:  PUSH   BC
EEAD'  2A F09F'                 LD     HL,(RWTRK)      ; GET LOGICAL TRACK
EEB0'  AF                LOOP2: XOR    A       ; DIVIDE BY NUMBER OF SIDES BY SHIFTING
                                        ; RIGHT LOG2 NUMBER OF OF SIDES
EEB1'  7C                       LD     A,H
EEB2'  1F                       RRA
EEB3'  67                       LD     H,A
EEB4'  7D                       LD     A,L
EEB5'  1F                       RRA
EEB6'  6F                       LD     L,A
EEB7'  10 F7                    DJNZ   LOOP2

EEB9'  ED 5B F09F'              LD     DE,(RWTRK)      ; GET LOGICAL TRACK NUMBER
EEBD'  22 F09F'                 LD     (RWTRK),HL      ; SAVE PYHSICAL TRACK NUMBER
EEC0'  C1                       POP    BC
EEC1'  AF                LOOP3: XOR    A
EEC2'  7C                       LD     A,H
EEC3'  17                       RLA
EEC4'  67                       LD     H,A
EEC5'  7D                       LD     A,L
EEC6'  17                       RLA
EEC7'  6F                       LD     L,A
EEC8'  10 F7                    DJNZ   LOOP3

EECA'  AF                       XOR    A       ; CALCULATE SIDE NUMBER
EECB'  EB                       EX     DE,HL
EECC'  ED 52                    SBC    HL,DE
EECE'  7D                       LD     A,L

EECF'  32 F0A2'          TRNRET: LD     (RWSIDE),A      ; SAVE SIDE NUMBER
EED2'  E1                       POP    HL
EED3'  D1                       POP    DE
EED4'  C1                       POP    BC
EED5'  F1                       POP    AF
EED6'  C9                       RET

                                ;
                                ;       READ FLOPPY DISK SECTOR
                                ;          CARRY ON = ERROR RETURN
                                ;          CHANGED:     A
                                ;

EED7'  3E 82             RDFDSK: LD     A,READFD        ; GET READ COMMAND
EED9'  18 02                    JR     FDSKRW

                                ;
                                ;       WRITE FLOPPY DISK SECTOR
                                ;          CARRY ON = ERROR RETURN
                                ;          CHANGED:     A
                                ;

EEDB'  3E A2             WRFDSK: LD     A,WRITFD        ; GET WRITE COMMAND

EEDD'  32 F0AD'          FDSKRW: LD     (DSKCMD),A
EEE0'  C5                       PUSH   BC
EEE1'  E5                       PUSH   HL
```

```
EEE2'  3A F0A1'            LD     A,(RWSEC)       ; GET SECTOR NUMBER
EEE5'  D3 12               OUT    (FDSEC),A             ; SET SECTOR REGISTER
EEE7'  06 0A               LD     B,RETRY         ; GET RETRY COUNT
EEE9'  3A F0A2'  LOOP5:    LD     A,(RWSIDE)      ; GET SIDE NUMBER
EEEC'  B7                  OR     A       ; CHECK IF SIDE 0
EEED'  3A F0AD'            LD     A,(DSKCMD)      ; GET COMMAND
EEF0'  28 02               JR     Z,FDSK1         ; YES
EEF2'  CB DF               SET    SIDE,A          ; SET SIDE 1 IN COMMAND

EEF4'  21 F376'  FDSK1:    LD     HL,HSTBUF       ; GET BUFFER ADDRESS
EEF7'  CD FFF3             CALL   FDRW            ; READ/WRITE SECTOR
EEFA'  32 F0AA'            LD     (ERFLAG),A      ; SAVE STATUS
EEFD'  B7                  OR     A       ; CHECK IF ERROR
EEFE'  28 27               JR     Z,FDSKRT        ; NORMAL RETURN
EF00'  10 02               DJNZ   FDSK3
EF02'  18 0F               JR     FDSK4

EF04'  CB 67     FDSK3:    BIT    RNFERR,A
EF06'  CA EEE9'            JP     Z,LOOP5
EF09'  06 01               LD     B,1
EF0B'  CD EDEC'            CALL   RSTORE
EF0E'  CD EE8C'            CALL   SEEK2
EF11'  18 D6               JR     LOOP5

EF13'  47        FDSK4:    LD     B,A
EF14'  3A F09E'            LD     A,(RDWROP)      ; GET OPERATION CODE
EF17'  FE 00               CP     READ            ; CHECK IF READ
EF19'  28 0B               JR     Z,FDSKER        ; YES, ERROR RETURN

EF1B'  78                  LD     A,B
EF1C'  CB 77               BIT    WRPROT,A        ; CHECK IF WRITE PROTECTED
EF1E'  28 06               JR     Z,FDSKER        ; NO

EF20'  21 EF62'            LD     HL,ERMSG1       ; DISPLAY ERROR MESSAGE
EF23'  CD E3D1'            CALL   PMSG

EF26'  37        FDSKER:   SCF

EF27'  E1        FDSKRT:   POP    HL
EF28'  C1                  POP    BC
EF29'  C9                  RET

                           ;
                           ;       READ HARD DISK SECTOR
                           ;          CARRY ON = ERROR RETURN
                           ;          CHANGED:     A
                           ;

EF2A'  3E 20     RDHDSK:   LD     A,READHD        ; GET READ COMMAND
EF2C'  18 02               JR     HDSKRW

                           ;
                           ;       WRITE HARD DISK SECTOR
                           ;          CARRY ON = ERROR RETURN
                           ;          CHANGED:     A
                           ;
```

```
EF2E'  3E 30              WRHDSK: LD    A,WRITHD        ; GET WRITE COMMAND

EF30'  32 F0AD'           HDSKRW: LD    (DSKCMD),A
EF33'  C5                       PUSH  BC
EF34'  E5                       PUSH  HL
EF35'  3A F0A2'                 LD    A,(RWSIDE)      ; GET SIDE NUMBER
EF38'  E6 07                    AND   HEAD            ; GET HEAD NUMBER
EF3A'  6F                       LD    L,A
EF3B'  3A F0AE'                 LD    A,(SELBYT)      ; GET SELECT BYTE
EF3E'  B5                       OR    L       ; PUT HEAD NUMBER INTO SELECT BYTE
EF3F'  D3 36                    OUT   (HDSEL),A             ; SELECT DRIVE AND HEAD
EF41'  3A F0A1'                 LD    A,(RWSEC)       ; GET SECTOR NUMBER
EF44'  D3 33                    OUT   (HDSEC),A             ; SET SECTOR REGISTER
EF46'  2A F09F'                 LD    HL,(RWTRK)      ; GET TRACK NUMBER
EF49'  7D                       LD    A,L             ; SELECT TRACK
EF4A'  D3 34                    OUT   (HDCYLL),A
EF4C'  7C                       LD    A,H
EF4D'  D3 35                    OUT   (HDCYLH),A
EF4F'  3A F0AD'                 LD    A,(DSKCMD)      ; GET COMMAND
EF52'  21 F376'                 LD    HL,HSTBUF       ; GET BUFER ADDRESS
EF55'  CD FFED                  CALL  HDRW            ; READ/WRITE SECTOR
EF58'  32 F0AA'                 LD    (ERFLAG),A      ; SAVE STATUS
EF5B'  B7                       OR    A       ; CHECK IF ERROR
EF5C'  28 01                    JR    Z,HDSKRT        ; NORMAL RETURN

EF5E'  37                       SCF

EF5F'  E1                 HDSKRT: POP   HL
EF60'  C1                       POP   BC
EF61'  C9                       RET

                                ;
                                ;       MESSAGES
                                ;

EF62'  0D 0A 44 69        ERMSG1: DB    CR,LF,'Diskette is write-protected.',EOL
EF66'  73 6B 65 74
EF6A'  74 65 20 69
EF6E'  73 20 77 72
EF72'  69 74 65 2D
EF76'  70 72 6F 74
EF7A'  65 63 74 65
EF7E'  64 2E 00
EF81'  0D 0A 44 69        ERMSG2: DB    CR,LF,'Diskette seek error',EOL
EF85'  73 6B 65 74
EF89'  74 65 20 73
EF8D'  65 65 6B 20
EF91'  65 72 72 6F
EF95'  72 00
EF97'  0D 0A 50 72        ERMSG3: DB    CR,LF,'Printer is not ready.'
EF9B'  69 6E 74 65
EF9F'  72 20 69 73
EFA3'  20 6E 6F 74
EFA7'  20 72 65 61
EFAB'  64 79 2E
```

```
EFAE' 0D 0A 50 72                DB      CR,LF,'Press RETURN to continue.',EOL
EFB2' 65 73 73 20
EFB6' 52 45 54 55
EFBA' 52 4E 20 74
EFBE' 6F 20 63 6F
EFC2' 6E 74 69 6E
EFC6' 75 65 2E 00
EFCA' 0D 0A 44 69          ERMSG4: DB      CR,LF,'Disk is not activated.',EOL
EFCE' 73 6B 20 69
EFD2' 73 20 6E 6F
EFD6' 74 20 61 63
EFDA' 74 69 76 61
EFDE' 74 65 64 2E
EFE2' 00
EFE3' 0D 0A 43 61          ERMSG5: DB      CR,LF,'Cannot read system,'
EFE7' 6E 6E 6F 74
EFEB' 20 72 65 61
EFEF' 64 20 73 79
EFF3' 73 74 65 6D
EFF7' 2E
EFF8' 0D 0A 49 6E                DB      CR,LF,'Insert system diskette,'
EFFC' 73 65 72 74
F000' 20 73 79 73
F004' 74 65 6D 20
F008' 64 69 73 6B
F00C' 65 74 74 65
F010' 2E
F011' 0D 0A 50 72                DB      CR,LF,'Press RESET button.',EOL
F015' 65 73 73 20
F019' 52 45 53 45
F01D' 54 20 62 75
F021' 74 74 6F 6E
F025' 2E 00

                                 ;
                                 ; PVTC DATA STORAGE
                                 ;

F027' 00                   SCRLFLG:      DB OFF  ; SCROLL FLAG
F028' 00                   BKFRFLG:      DB 0    ; BACK/FOREGROUND FLAG
F029' 00                   LEADFLG:      DB 0    ; LEAD-IN FLAG
F02A' 00                   LDFLG:        DB 0

F02B' 00                   SWANL:        DB OFF  ; AUTO NEWLINE SWITCH
                                                 ; ON  - AUTO WRAPAROUND, NO LINEFEED
                                                 ; OFF - LINE TRUNCATE, LINEFEED
F02C' 00                   SWALF:        DB OFF  ; AUTO LINEFEED SWITCH
                                                 ; ON  - AUTO LINEFEED WITH CR
                                                 ; OFF - NO LINEFEED WITH CR
F02D' 00                   SWLEAD:       DB OFF  ; LEAD-IN SWITCH
                                                 ; ON  - ESC
                                                 ; OFF - TILDA

F02E' 00                   LINETP:       DB 0    ; TEMPORARY LINE NUMBER
F02F' 00                   COLTP:        DB 0    ; TEMPORARY COLUMN NUMBER
F030' 00                   CURCNT:       DB 0    ; CURSOR COUNT NUMBER
```

```
F031'  FF                CURSW:    DB ON   ; CURSOR SWITCH
F032'  00                CHRCNT:   DB 0    ; CHARACTER COUNT
F033'  00                GRPHSW:   DB OFF  ; GRAPHIC SWITCH
F034'  00                TIMEOUT:  DB OFF  ; TIMEOUT ON LIST
F035'  00                KEYSW:    DB OFF  ; KEYBOARD LOCK OUT
F036'  F578'             BUFADR:   DW CHRBUF       ; CHARACTER BUFFER ADDRESS
F038'  0A                COUNTER:  DB 10

                                   ;
                                   ; OUTPUT PROCESSING ROUTINES
                                   ;

F039'  EB5A' E801'       CHRTAB: DW      NONE,ONC,OFFC,OND,OFFD,SNDADR,KUNLK,BELL
F03D'  E80B' E7F7'
F041'  E7FC' E7D5'
F045'  E7C2' E79F'
F049'  E6D5' EB5A'               DW      BKSPACE,NONE,LFEED,CURDWN,CURUP,CARRTN,NONE,CLREND
F04D'  E678' E884'
F051'  E841' E6B6'
F055'  EB5A' E9B3'
F059'  E70B' E8C8'               DW      CURIGT,DIRADR,CURHME,DELLNE,NONE,KLOCK,NONE,CLRBK
F05D'  E8B2' EA2A'
F061'  EB5A' E7BC'
F065'  EB5A' E9E0'
F069'  E9DC' E821'               DW      CLRFSC,SETBK,INSLNE,ESCAPE,CLRSCN,NONE,GRAPH,SETFR
F06D'  EAA5' E826'
F071'  E96E' EB5A'
F075'  E815' E81B'

                                   ;
                                   ; LEAD-IN TABLE
                                   ;

F079'  00 FF FF FF       LEADTB: DB      OFF,ON,ON,ON,ON,ON,ON,OFF
F07D'  FF FF FF 00
F081'  00 00 00 FF               DB      OFF,OFF,OFF,ON,ON,OFF,OFF,ON
F085'  FF 00 00 FF
F089'  00 FF FF FF               DB      OFF,ON,ON,ON,OFF,ON,OFF,ON
F08D'  00 FF 00 FF
F091'  FF FF FF 00               DB      ON,ON,ON,OFF,ON,OFF,ON,ON
F095'  FF 00 FF FF

                                   ;
                                   ;       UNITIALIZED DATA AREA
                                   ;

                                   ;
                                   ;       READ/WRITE OPERATION DATA
                                   ;

F099'                    SEKDSK: DS      1  ; SEEK DISK NUMBER
F09A'                    SEKTRK: DS      2  ; SEEK LOGICAL TRACK NUMBER
F09C'                    SEKSEC: DS      1  ; SEEK LOGICAL SECTOR NUMBER

                                   ;
                                   ;       BUFFER OPERATION DATA
```

```
                                ;
F09D'                   WRTYPE: DS      1  ; WRITE BUFFER TYPE
F09E'                   RDWROP: DS      1  ; READ/WRITE OPERATION CODE
F09F'                   RWTRK:  DS      2  ; READ/WRITE TRACK NUMBER
F0A1'                   RWSEC:  DS      1  ; READ/WRITE SECTOR NUMBER
F0A2'                   RWSIDE: DS      1  ; READ/WRITE SIDE NUMBER
F0A3'                   BUFDSK: DS      1  ; DISK NUMBER FOR HOST BUFFER
F0A4'                   BUFTRK: DS      2  ; TRACK NUMBER FOR HOST BUFFER
F0A6'                   BUFSEC: DS      1  ; SECTOR NUMBER FOR HOST BUFFER
F0A7'                   BUFACT: DS      1  ; HOST BUFFER ACTIVE INDICATOR
F0A8'                   HSTSEC: DS      1  ; PHYSICAL SECTOR NUMBER
F0A9'                   LOGOFF: DS      1  ; LOGICAL OFFSET IN PHYSICAL SECTOR
F0AA'                   ERFLAG: DS      1  ; ERROR FLAG
F0AB'                   DMAAD:  DS      2  ; DMA ADDRESS
F0AD'                   DSKCMD: DS      1  ; DISK COMMAND


                                ;
                                ;       SCRATCH DATA AREA
                                ;
F0AE'                   BEGDAT  EQU     $  ; BEGINNING OF DATA AREA

                                ;
                                ;       DISK TABLE SAVE AREA BEGIN
                                ;
F0AE'                   SELBYT: DS      1  ; SELECT BYTE
F0AF'                   SECSIZ: DS      2  ; SECTOR SIZE
F0B1'                   STEPS:  DS      1  ; STEPPING RATE
F0B2'                   SIDES:  DS      1  ; NUMBER OF SIDES
F0B3'                   DSKTYP: DS      1  ; DISK TYPE

                                ;
                                ;       DISK TABLE SAVE AREA END
                                ;

                                ;
                                ;       TAB SAVE AREA BEGIN
                                ;
F0B4'                   HSTSIZ: DS      2  ; PHYSICAL SECTOR SIZE
F0B6'                   BLKSIZ: DS      2  ; BLOCK SIZE
F0B8'                   HSTBLK: DS      1  ; NUMBER OF LOGICAL SECTOR PER PHYSICAL SECTOR
F0B9'                   CPMSPT: DS      1  ; NUMBER OF LOGICAL SECTORS PER LOGICAL TRACK
F0BA'                   SECMSK: DS      1  ; HSTBLK - 1
F0BB'                   SECSHF: DS      1  ; LOG2(HSTBLK)

                                ;
                                ;       TAB SAVE AREA END
                                ;

F0BC'                   LSTADR: DS      2  ; LAST TRACK ADDRESS
F0BE'                   DSKTAB: DS      2  ; CURRENT DISK TABLE ADDRESS
F0C0'                   DPADDR: DS      2  ; CURRENT DISK PARAMETER ADDRESS
```

```
                        ;
                        ;       DISK BUFFERS
                        ;

F0C2'                   DIRBF:  DS      128     ; SCRATCH DIRECTORY AREA
F142'                   ALL00:  DS      12 ; ALLOCATION VECTOR 0
F14E'                   ALL01:  DS      49 ; ALLOCATION VECTOR 1
F17F'                   ALL02:  DS      49 ; ALLOCATION VECTOR 2
F1B0'                   ALL03:  DS      77 ; ALLOCATION VECTOR 3
F1FD'                   ALL04:  DS      153     ; ALLOCATION VECTOR 4
F296'                   CHK00:  DS      16 ; CHECK VECTOR 0
F2A6'                   CHK01:  DS      64 ; CHECK VECTOR 1
F2E6'                   CHK02:  DS      64 ; CHECK VECTOR 2

F326'                   BUFTP:  DS      80 ; TEMPORARY BUFFER
F376'                   HSTBUF: DS      512     ; LARGEST PHYSICAL SECTOR SUPPORTED
F576'                   TIMER:  DS      2  ; LIST DEVICE TIMER
F578'                   CHRBUF: DS      10 ; INPUT CHARACTER
F582'                   DOSSTK: DS      2  ; STORAGE FOR BDOS STACK
                        ;
F584'                           DS      60
F5C0'                   LOCSTK  EQU     $  ; BIOS STACK

F5C0'                   ENDDAT  EQU     $       ; END OF DATA AREA
0512                    DATSIZ  EQU     $-BEGDAT        ; SIZE OF DATA AREA
                                END
```

Macros:

Symbols:

F142'	ALL00	F14E'	ALL01	F17F'	ALL02
F1B0'	ALL03	F1FD'	ALL04	D406	BDOS
0006	BDOSAD	0007	BEEP	F0AE'	BEGDAT
E79F'	BELL	9800	BIAS	E200	BIOS
F028'	BKFRFLG	E6D5'	BKSPACE	0020	BLANK
F0B6'	BLKSIZ	0006	BLOCK	E378'	BOOT
E703'	BSMOV1	E6EC'	BSMOVL	E704'	BSOUT
E70A'	BSRET	F0A7'	BUFACT	F036'	BUFADR
F0A3'	BUFDSK	F0A6'	BUFSEC	F326'	BUFTP
F0A4'	BUFTRK	0005	BWSCN	E6B6'	CARRTN
CC00	CCP	0004	CDISK	E8AB'	CDWN1
E8B1'	CDWNRT	E634'	CHAROUT	F296'	CHK00
F2A6'	CHK01	F2E6'	CHK02	E66F'	CHKLIN
E677'	CHORET	F578'	CHRBUF	F032'	CHRCNT
E65E'	CHROT	E668'	CHROT1	F039'	CHRTAB
E9E0'	CLRBK	E9B3'	CLREND	E9E2'	CLRF1
EA15'	CLRF2	EA22'	CLRFRT	E9DC'	CLRFSC
E9A0'	CLRSC1	E96E'	CLRSCN	0000	COL40
E556'	COLBCK	E567'	COLBK1	E56C'	COLBK2
E56F'	COLBRT	0037	COLBW	E541'	COLFOR
E552'	COLFRT	0038	COLFW	FFD4	COLNO
F02F'	COLTP	E5BB'	CON1	E5DC'	CON1A
E5D3'	CON1C	E5E1'	CON2	E5EA'	CON3
E5EF'	CON3A	E5F4'	CON4	E618'	CON5
E60D'	CON6	E61D'	CONEND	E4E4'	CONIN
E4EA'	CONIN2	FFEA	CONINP	E59D'	CONOUT
E628'	CONRT	E4A8'	CONST	E4D9'	CONST1
E4DB'	CONST2	E4BC'	CONST3	FFE7	CONSTA
0018	CONTRL	E96B'	CONVRT	F038'	COUNTER
F0B9'	CPMSPT	000D	CR	E883'	CUPRET
F030'	CURCNT	E884'	CURDWN	E73C'	CURET
E8B2'	CURHME	E71F'	CURIG1	E735'	CURIG2
E736'	CURIG3	E70B'	CURIGT	E8C7'	CURRET
F031'	CURSW	E841'	CURUP	E86F'	CURUP1
E87D'	CURUP2	0512	DATSIZ	EA45'	DEL1
EA66'	DEL2	EA69'	DEL3	EA72'	DEL4
FFFC	DELAY	EA8C'	DELCLR	EA2A'	DELLNE
E8CE'	DIADR1	E8FD'	DIADR2	E953'	DIADR3
E958'	DIADR4	E95E'	DIADR5	E93D'	DIADR6
E922'	DIADR7	E8F4'	DIADR8	E919'	DIADR9
E8E8'	DIADRA	E910'	DIADRB	E8E3'	DIADRC
E8F7'	DIADRD	E930'	DIADRE	ED02'	DIFBUF
E8C8'	DIRADR	F0C2'	DIRBF	E31F'	DISK0
E32A'	DISK1	E335'	DISK2	E340'	DISK3
E34B'	DISK4	0000	DISKTY	F0AB'	DMAAD
FFE3	DMAPTP	F582'	DOSSTK	F0C0'	DPADDR
E237'	DPBASE	E2C4'	DPBLK0	E2D3'	DPBLK1
E2E2'	DPBLK2	E2F1'	DPBLK3	E300'	DPBLK4
FFE5	DPBTP	F0AD'	DSKCMD	E233'	DSKMAP
EDEB'	DSKRET	EDD6'	DSKSEL	EDE4'	DSKSL1
F0BE'	DSKTAB	F0B3'	DSKTYP	07FF	ENDADR
F5C0'	ENDDAT	0000	EOL	F0AA'	ERFLAG

EF62'	ERMSG1	EF81'	ERMSG2	EF97'	ERMSG3
EFCA'	ERMSG4	EFE3'	ERMSG5	E826'	ESCAPE
0010	FDBASE	0010	FDCMD	0013	FDDATA
0007	FDERR	FFF9	FDHCMD	0002	FDISK1
0004	FDISK2	0008	FDISK3	FFF3	FDRW
0012	FDSEC	FFF6	FDSEL	EEF4'	FDSK1
EF04'	FDSK3	EF13'	FDSK4	EF26'	FDSKER
EF27'	FDSKRT	EEDD'	FDSKRW	0010	FDSTAT
0011	FDTRK	000C	FMFEED	0080	FORGND
EC39'	GETPARM	E454'	GOCPM	E815'	GRAPH
0003	GRAPHIC	F033'	GRPHSW	0030	HDBASE
0037	HDCMD	0035	HDCYLH	0034	HDCYLL
0030	HDDATA	0007	HDERRZ	FFF0	HDHCMD
0006	HDINT	0003	HDISK1	0005	HDISK2
0031	HDPRCM	FFED	HDRW	0033	HDSEC
0036	HDSEL	EF5F'	HDSKRT	EF30'	HDSKRW
0037	HDSTAT	0007	HEAD	0002	HGHINT
EBE8'	HOME	000C	HOMEFD	0010	HOMEHD
F0B8'	HSTBLK	F376'	HSTBUF	EDD3'	HSTERR
EDD0'	HSTNOR	EDD4'	HSTRET	F0A8'	HSTSEC
F0B4'	HSTSIZ	E515'	IN06	E51E'	IN07
E527'	IN08	E531'	IN09	E50A'	INFUNC
E53A'	INRET	E53C'	INRET1	E53D'	INRET2
EAC5'	INS1	EADF'	INS2	EAFC'	INS3
EAFF'	INS4	EB0A'	INS5	EAA5'	INSLNE
EB20'	INSRT	0004	INTLCE	0018	INTMSK
000E	INTREQ	0000	INTRPT	0003	IOBYTE
FFE2	IOREG	0044	IR0	0032	IR1
0097	IR10	000C	IR2	00AA	IR3
0017	IR4	004F	IR5	0009	IR6
0029	IR7	0080	IR8	0010	IR9
FFE1	IREG	00C3	JMPCOD	E62D'	JUMP
0004	KBDINT	0004	KCAPLK	0008	KEYBD
F035'	KEYSW	E7BC'	KLOCK	E7C2'	KUNLK
F02A'	LDFLG	F029'	LEADFLG	F079'	LEADTB
0039	LEFT	000A	LF	E678'	LFEED
E684'	LFEED1	E6A4'	LFEED2	E6AB'	LFEED3
E6B5'	LFRET	0003	LINE0	FFD6	LINEN0
F02E'	LINETP	EB5B'	LIST	EB73'	LIST1
EB85'	LISTOT	EB84'	LISTRT	EBCB'	LISTS1
EBCE'	LISTS2	EBD2'	LISTS3	EB94'	LISTST
F5C0'	LOCSTK	F0A9'	LOGOFF	E459'	LOOP0
EEA5'	LOOP1	EB44'	LOOP10	EB53'	LOOP11
EC19'	LOOP12	E7AB'	LOOP13	E7B4'	LOOP14
E7A2'	LOOP16	EBB5'	LOOP17	EEB0'	LOOP2
EB76'	LOOP20	E7CB'	LOOP21	E508'	LOOP22
EEC1'	LOOP3	EE4F'	LOOP4	EEE9'	LOOP5
E93A'	LOOP6	EB30'	LOOP9	F0BC'	LSTADR
EBE0'	LSTRET	0005	MARGIN	0050	MAXCOL
0003	MAXDSK	0017	MAXLIN	0028	MAXOFF
0028	MAXTVC	004C	MODE	FFE0	MODREG
003A	MSIZE	EB5A'	NONE	E4A0'	NXTSEC
0000	OFF	E80B'	OFFC	0030	OFFCUR
E7FC'	OFFD	0028	OFFDIS	FFD9	OFFSET
FFDB	OFFTMP	00FF	ON	E801'	ONC
0031	ONCUR	E7F7'	OND	003D	ONDICU

0029	ONDIS	ED1B'	OPREAD	000A	PADDING
FFE7	PBASE	E3D1'	PMSG	E3D5'	PMSG1
0020	PRECMP	0005	PRINT	0000	PROMPT
0005	PTRBSY	000D	PTRDAT	0003	PTRSTR
EBE2'	PUNCH	00AC	RDCUR	ECB1'	RDDISK
EED7'	RDFDSK	EF2A'	RDHDSK	ED78'	RDHST
ED9E'	RDHST1	00A4	RDPT	E48B'	RDSECS
E4A7'	RDSRET	E48D'	RDSTRY	E433'	RDTRK1
F09E'	RDWROP	0005	RDYFLG	0000	READ
EBE5'	READER	0082	READFD	0020	READHD
0001	REBOOT	0080	RECLEN	0000	RESET
000A	RETRY	0030	RIGHT	0004	RNFERR
EE1C'	RSTERR	0040	RSTFLG	EE13'	RSTNOR
EDFA'	RSTOR0	EE04'	RSTOR1	EE0C'	RSTOR2
EDEC'	RSTORE	EE1D'	RSTRET	0001	RVERSE
ECCA'	RW1	ECD2'	RW2	ED1E'	RWLAST
ECBD'	RWOPER	ED37'	RWRET	F0A1'	RWSEC
F0A2'	RWSIDE	F09F'	RWTRK	E59B'	SCNRET
E585'	SCNUP1	E598'	SCNUP2	E570'	SCNUPD
F027'	SCRLFLG	E778'	SCRLL1	E779'	SCRLL2
E789'	SCRLL3	E73D'	SCROLL	FFD7	SCSTART
F0BA'	SECMSK	F0BB'	SECSHF	F0AF'	SECSIZ
EC94'	SECTN1	EC87'	SECTRAN	EC97'	SECTRT
EE1F'	SEEK	EE2C'	SEEK0	EE39'	SEEK1
EE8C'	SEEK2	EE81'	SEEKER	001C	SEEKFD
EE7E'	SEEKNO	EE74'	SEEKOK	EE88'	SEEKRT
F099'	SEKDSK	0002	SEKRTY	F09C'	SEKSEC
F09A'	SEKTRK	EC2A'	SELBAD	F0AE'	SELBYT
EBF0'	SELDSK	0000	SELFA	0001	SELFB
EC1F'	SELHM1	EC06'	SELHME	EC24'	SELOK
EC38'	SELRET	EC30'	SELRST	0002	SELSID
E821'	SETBK	EC99'	SETDMA	E81B'	SETFR
EC82'	SETSEC	EBEB'	SETTRK	0003	SIDE
F0B2'	SIDES	E35A'	SIGNON	0090	SKERR
E7E7'	SNDAD1	E7F1'	SNDAD2	E7D5'	SNDADR
0002	SPLIT	0080	STADR	0018	STATUS
005C	STEPFD	F0B1'	STEPS	F02C'	SWALF
F02B'	SWANL	FFDF	SWITCH	F02D'	SWLEAD
0005	SYSCAL	E30F'	TAB0	E317'	TAB3
007E	TILDA	F034'	TIMEOUT	F576'	TIMER
FFFF	TIMING	E830'	TLDA	E83D'	TLDA1
ED75'	TRAN1	ED3F'	TRANS	E29B'	TRANS0
000A	TRKOEN	0002	TRKOST	0002	TRK1EN
0001	TRK1ST	EEAC'	TRN1	EECF'	TRNRET
EE92'	TRNTRK	0001	TV	000A	TVCNT
E423'	TWOSID	E78E'	UPDLIN	E79E'	UPDLRT
FFDD	VBFLAG	0004	VBLANK	0000	VTCBASE
0001	VTCCMD	0004	VTCCU1	0005	VTCCU2
001C	VTCDAT	0000	VTCINI	000D	VTCINP
001D	VTCMOD	0006	VTCPT1	0007	VTCPT2
0002	VTCSC1	0003	VTCSC2	0001	VTCSTA
EB2D'	WAIT	EB38'	WAIT1	EB28'	WAIT2
EB4D'	WAITNO	EB59'	WAITRT	0000	WARMST
E449'	WBERR	E3E2'	WBOOT	E3F5'	WBOOT1
E203'	WBOOTE	001D	WIDTH	0000	WRALL
00BB	WRCP	00AA	WRCUR	0001	WRDIR

EC9F'	WRDISK	EEDB'	WRFDSK	EF2E'	WRHDSK
EDA5'	WRHST	EDCB'	WRHST1	00AB	WRINC
0001	WRITE	00A2	WRITFD	0030	WRITHD
0006	WRPROT	F09D'	WRTYPE	0002	WRULA

No Fatal error(s)