```
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*     F L O P P Y   D I S K   F O R M A T T E R       *
                        ;*      This program format floopy disk with           *
                        ;*         10 sector with 256 byte/sector              *
                        ;*            in single or double side                 *
                        ;*                                                     *
                        ;*******************************************************
                        ;
                              title   Floopy Disk Formatter for NE CP/M 2.2
                              subttl  Copyright 1984 Studio Lg. Genova - Last rev 15/08/84 09:38
                        ;     Programmer: Martino Stefano
                        ;     Modified by Gallerani Paolo
                        ;
                        ;
0001                    true    equ     1
0000                    false   equ     0
                        ;
0000                    copyrig equ     false
                        ;
                    C         Include ASCII.LIB
                    C   ;
                    C   ;
                    C   ;*******************************************************
                    C   ;*                                                     *
                    C   ;*             ASCII EQUIVALENTS                       *
                    C   ;*                                                     *
                    C   ;*******************************************************
                    C   ;
0007                C   bell    equ     '@'-'@'         ; ring beeper
0008                C   backsp  equ     'H'-'@'         ; back space char.
0009                C   tab     equ     'I'-'@'         ; tabulation char.
000A                C   lf      equ     'J'-'@'         ; line-feed char.
000C                C   ffeed   equ     'L'-'@'         ; form feed char.
000D                C   cr      equ     'M'-'@'         ; carriage-return char.
0013                C   ofx     equ     'S'-'@'         ; attributes ofx
0042                C   rever   equ     'B'             ; Reverse On    (^SB)
0043                C   flash   equ     'C'             ; Flash On      (^SC)
0040                C   norm    equ     '@'             ; Normal        (^S@)
0020                C   space   equ     ' '             ; space char.
0024                C   endmsg  equ     '$'             ; end of print message
                    C   ;
                        ;
                              page    62
```

```
                        ;
                C               Include ROMENTRY.LIB
                C       ;
                C       ;
                C       ;*******************************************************
                C       ;*                                                     *
                C       ;*              Rom routines address                   *
                C       ;*                                                     *
                C       ;*******************************************************
                C       ;
F000            C       rom     equ     0F000h          ; <--- rom starting address
F003            C       cin     equ     rom+3           ; console input
F006            C       cout    equ     rom+6           ; console output
F009            C       csts    equ     rom+9           ; console status
F00C            C       lout    equ     rom+12          ; printer output
F00F            C       lsts    equ     rom+15          ; printer status
F012            C       fdios   equ     rom+18          ; fdd I/O 128 byte
F015            C       fdiod   equ     rom+21          ; fdd I/O 256 byte
F018            C       wdini   equ     rom+24          ; wdd initialization
F01B            C       wdio    equ     rom+27          ; wdd I/O 256 byte
F01E            C       strout  equ     rom+30          ; print string .DE until $
F01E            C       print   equ     strout          ; sinonime
F021            C       bootrom equ     rom+33          ; load BIOS and go to wboote
F024            C       printat equ     rom+36          ; print str. -> DE at -> HL cursor
F027            C       movcurs equ     rom+39          ; move cursor at -> HL
F02A            C       vidinit equ     rom+42          ; initialize video
F02D            C       ComoFlo equ     rom+45          ; Version Number
                C       ;
                        ;
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*              Message cursor position                *
                        ;*                                                     *
                        ;*******************************************************
                        ;
0700                    Confadd equ     0700h           ; Confirm msg position (X=01 Y=09)
1019                    ermsgadd equ    1019h           ; error msg position (X=25 Y=14)
0C23                    msg1add equ     0c23h           ; form/ver msg position (X=36 Y=10)
0E1F                    msg2add equ     0e1fh           ; trk/sid msg position (X=32 Y=15)
0E25                    tnumadd equ     0e25h           ; trk num. position (X=38 Y=15)
0E2F                    snumadd equ     0e2fh           ; sid num. position (X=48 Y=15)
                                ;
                                page
```

```
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*              FD 1771 I/O port                       *
                        ;*                                                     *
                        ;*******************************************************
00D0                    fdsts   equ     0d0h            ; fdd status port
00D1                    fdtrk   equ     0d1h            ; fdd track port
00D2                    fdsec   equ     0d2h            ; fdd sector port
00D6                    fdlch   equ     0d6h            ; fdd lach port
00D7                    fddat   equ     0d7h            ; fdd data port
00D0                    fdcmd   equ     fdsts           ; fdd command port
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*              FD 1771 Command Summary                *
                        ;*                                                     *
                        ;*******************************************************
                        ;
                                ; this command are without verify because disk isn't formatted
                        ;
0002                    fdrest  equ     00000010b       ; fdd restore command code
00D0                    fdrst   equ     11010000b       ; fdd reset int. command code
0052                    fdsin   equ     01010010b       ; fdd step in command code
00F4                    fdwtrk  equ     11110100b       ; fdd write track command code
                        ;
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*              FD 1771 Flag Mask                      *
                        ;*                                                     *
                        ;*******************************************************
                        ;
0001                    fdbusy  equ     001h            ; busy flag is bit 0
0002                    fddrq   equ     002h            ; drq flag is bit 1
0040                    fdwprt  equ     040h            ; write protect flag is bit 6
0080                    fdnrdy  equ     080h            ; not ready flag is bit 7
0018                    fdtier  equ     018h            ; mask error for type I commands
001F                    fdt23er equ     01fh            ; mask error for type II and type III commands
                        ;
                                page
```

```
                        ;
                        ;*********************************************************
                        ;* S T A R T                                             *
                        ;*          Set stack and go to floppy format            *
                        ;*********************************************************
                        ;
0000'                           aseg
                                org     100h
                        ;
                        ;
0100    31 0126                 ld      sp,stack        ; set stack pointer
0103    C3 056E                 jp      fddform         ; go to floppy format
                        ;
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*                  Ram data areas                       *
                        ;*                                                       *
                        ;*********************************************************
                        ;
                                if      copyrig
                                defb    ' COPYRIGHT (c) 1983 by STUDIO Lg. Genova. ITALY '
                                endif
                        ;
0106                            defs    32              ; stack data areas
0126                    stack   equ     $
                        ;
                        ;
                        ;
0126                    DskSid: defs    1               ; byte to output latch
                        ;
0127                    FMode:  defs    1               ; Single or Multiple mode
0128                    FSides: defs    1               ; Single or Double Side
0129                    FTrks:  defs    1               ; 40 or 80 Tracks
                        ;
                        ;
012A                    xlt:
012A    05 09 03 07             defb    5,9,3,7,2,6,10,4,8,0
012E    02 06 0A 04
0132    08 00
                        ;
                        ;
0134                    verbuff:
0134                            defs    256             ; verify buffer (256 byte)
                        ;
                        ;
                        ;*********************************************************
                        ;* V E R T A B                                           *
                        ;*          Param table for disk verify                  *
                        ;*********************************************************
                        ;
0234                    VerTab:                         ; Verify Parameters
0234    00              VerDsk: defb    0               ; disk and side
0235    0000            VerTrk: defw    0000            ; track number
0237    01              VerSec: defb    1               ; sector number
0238    0134            VerDma: defw    verbuff         ; dma address
```

```
023A  00                  VerOp:  defb    0               ; verify = read operation
                          ;
                          ;
                          ;
                          ;*********************************************************
                          ;*            E r r o r   M e s s a g e s                *
                          ;*********************************************************
                          ;
023B                      oflmsg:
023B  0D 0A 44 69                 defb    cr,lf,'Disk offline',endmsg
023F  73 6B 20 6F
0243  66 66 6C 69
0247  6E 65 24
                          ;
024A                      fmterrmsg:
024A  0D 0A 46 6F                 defb    cr,lf,'Format error',endmsg
024E  72 6D 61 74
0252  20 65 72 72
0256  6F 72 24
                          ;
0259                      rderrmsg:
0259  56 65 72 69                 defb    'Verify read error',endmsg
025D  66 79 20 72
0261  65 61 64 20
0265  65 72 72 6F
0269  72 24
                          ;
026B                      verrmsg:
026B  56 65 72 69                 defb    'Verify data error',endmsg
026F  66 79 20 64
0273  61 74 61 20
0277  65 72 72 6F
027B  72 24
                          ;
027D                      wterr:
027D  20 2D 20 48                 defb    ' - Hit any key ',endmsg
0281  69 74 20 61
0285  6E 79 20 6B
0289  65 79 20 24
                          ;
                          ;
                          ;*********************************************************
                          ;*     F o r m a t t i n g   O p t i o n                 *
                          ;*********************************************************
                          ;
028D                      IniMsg:
028D  0C 0D 0A 13                 defb    ffeed,cr,lf,19,'H'
0291  48
0292  20 20 20 46                 defb    '   Floppy Disk formatter vers 4.22',cr,lf
0296  6C 6F 70 70
029A  79 20 44 69
029E  73 6B 20 66
02A2  6F 72 6D 61
02A6  74 74 65 72
02AA  20 76 65 72
02AE  73 20 34 2E
02B2  32 32 0D 0A
```

```
02B6  13 40 0D 0A       CrLFOF: 19,'@',cr,lf,endmsg
02BA  24
                                :
02BB  28 53 29 69       Quest0: defb    '(S)ingle or (M)ultiple Mode (or ^C to exit) . M',backsp,endmsg
02BF  6E 67 6C 65
02C3  20 6F 72 20
02C7  28 4D 29 75
02CB  6C 74 69 70
02CF  6C 65 20 4D
02D3  6F 64 65 20
02D7  28 6F 72 20
02DB  5E 43 20 74
02DF  6F 20 65 78
02E3  69 74 29 20
02E7  2E 20 4D 08
02EB  24
02EC  28 53 29 69       Quest1: defb    '(S)ingle or (D)ouble Side ..... S',backsp,endmsg
02F0  6E 67 6C 65
02F4  20 6F 72 20
02F8  28 44 29 6F
02FC  75 62 6C 65
0300  20 53 69 64
0304  65 20 2E 2E
0308  2E 2E 2E 2E
030C  2E 2E 2E 2E
0310  2E 2E 2E 2E
0314  2E 2E 2E 2E
0318  2E 20 53 08
031C  24
031D  28 34 29 30       Quest2: defb    '(4)0 or (8)0 Tracks ..... 40',backsp,backsp,endmsg
0321  20 6F 72 20
0325  28 38 29 30
0329  20 54 72 61
032D  63 6B 73 20
0331  2E 2E 2E 2E
0335  2E 2E 2E 2E
0339  2E 2E 2E 2E
033D  2E 2E 2E 2E
0341  2E 2E 2E 2E
0345  2E 2E 2E 2E
0349  2E 20 34 30
034D  08 08 24
0350  44 69 73 6B       Quest3: defb    'Disk to be format (A) o (B) ..... B',backsp,endmsg
0354  20 74 6F 20
0358  62 65 20 66
035C  6F 72 6D 61
0360  74 20 28 41
0364  29 20 6F 20
0368  28 42 29 20
036C  2E 2E 2E 2E
0370  2E 2E 2E 2E
0374  2E 2E 2E 2E
0378  2E 2E 2E 2E
037C  2E 20 42 08
0380  24
0381  0D 0A 07 43       Quest4: defb    cr,lf,bell,'Confirm Formatting of Disk ',19,'H'
0385  6F 6E 66 69
0389  72 6D 20 46
```

```
038D  6F 72 6D 61
0391  74 74 69 6E
0395  67 20 6F 66
0399  20 44 69 73
039D  6B 20 13 48
03A1                    disknum:
03A1  00                        defb    0
03A2  13 40 20 28               defb    19,'@ (^C to exit) ... ',endmsg
03A6  5E 43 20 74
03AA  6F 20 65 78
03AE  69 74 29 20
03B2  2E 2E 2E 20
03B6  24

                        ;
                        ;
03B7  13 48 53 49       Msg.SM: defb    19,'HSINGLE',endmsg
03BB  4E 47 4C 45
03BF  24
03C0  13 48 4D 55       Msg.MM: defb    19,'HMULTIPLE',endmsg
03C4  4C 54 49 50
03C8  4C 45 24
03B7                    Msg.SS  equ     Msg.SM
03CB  13 48 44 4F       Msg.DS: defb    19,'HDOUBLE',endmsg
03CF  55 42 4C 45
03D3  24
03D4  13 48 34 30       Msg.40: defb    19,'H40',endmsg
03D8  24
03D9  13 48 38 30       Msg.80: defb    19,'H80',endmsg
03DD  24
03DE  13 48 41 24       Msg.A:  defb    19,'HA',endmsg
03E2  13 48 42 24       Msg.B:  defb    19,'HB',endmsg
03E6  59 45 53          MsgYES: defb    'YES'
                        ;
                        ;
                        ;
03E9                    fvisual:
03E9  07 13 48 46               defb    bell,19,'H','FORMATTING',19,'@',endmsg
03ED  4F 52 4D 41
03F1  54 54 49 4E
03F5  47 13 40 24
                        ;
03F9                    vvisual:
03F9  07 13 48 56               defb    bell,19,'H','VERIFYING ',19,'@',endmsg
03FD  45 52 49 46
0401  59 49 4E 47
0405  20 13 40 24
                        ;
0409                    vsidtrk:
0409  54 52 41 43               defb    'TRACK      SIDE',endmsg
040D  4B 20 20 20
0411  20 20 20 53
0415  49 44 45 24
                                page
```

```
                        ;
                        ;*********************************************************
                        ;* F O R M T A B                                         *
                        ;*            Table for track format                     *
                        ;*********************************************************
                        ;
0419                    fddtab:
                        ;
                        ; *** PREAMBLE ***
                        ;
0419                    preamble:
                                rept    40              ; track preamble (40 byte 'ff')
                                defb    255             ;
                                endm
0419    FF          +           defb    255             ;
041A    FF          +           defb    255             ;
041B    FF          +           defb    255             ;
041C    FF          +           defb    255             ;
041D    FF          +           defb    255             ;
041E    FF          +           defb    255             ;
041F    FF          +           defb    255             ;
0420    FF          +           defb    255             ;
0421    FF          +           defb    255             ;
0422    FF          +           defb    255             ;
0423    FF          +           defb    255             ;
0424    FF          +           defb    255             ;
0425    FF          +           defb    255             ;
0426    FF          +           defb    255             ;
0427    FF          +           defb    255             ;
0428    FF          +           defb    255             ;
0429    FF          +           defb    255             ;
042A    FF          +           defb    255             ;
042B    FF          +           defb    255             ;
042C    FF          +           defb    255             ;
042D    FF          +           defb    255             ;
042E    FF          +           defb    255             ;
042F    FF          +           defb    255             ;
0430    FF          +           defb    255             ;
0431    FF          +           defb    255             ;
0432    FF          +           defb    255             ;
0433    FF          +           defb    255             ;
0434    FF          +           defb    255             ;
0435    FF          +           defb    255             ;
0436    FF          +           defb    255             ;
0437    FF          +           defb    255             ;
0438    FF          +           defb    255             ;
0439    FF          +           defb    255             ;
043A    FF          +           defb    255             ;
043B    FF          +           defb    255             ;
043C    FF          +           defb    255             ;
043D    FF          +           defb    255             ;
043E    FF          +           defb    255             ;
043F    FF          +           defb    255             ;
0440    FF          +           defb    255             ;
                        ;
                        ; *** GAP III ***
```

```
                        ;
0441                    idfield:
                                reot    6               ; (6 bytes '00')
                                defb    0               ;
                                endm
0441    00              +       defb    0               ;
0442    00              +       defb    0               ;
0443    00              +       defb    0               ;
0444    00              +       defb    0               ;
0445    00              +       defb    0               ;
0446    00              +       defb    0               ;
                        ;
                        ; ***** ID fields *****
                        ;
0447    FE                      defb    0feh            ; ID Address Mark
0448                    ID.Trk: defs    1               ; track number  (1 byte)
0449    00              ID.Sid: defb    0               ; side number   (0 or 1)
044A                    ID.Sec: defs    1               ; sector number (1 byte)
044B    01                      defb    1               ; sector lenght (256 byte)
                        ;
044C    F7                      defb    0f7h            ; 2 CRC's written
                        ;
                        ; *** GAP II ***
                        ;
                                reot    11              ; GAP II (11 bytes 'ff')
                                defb    255             ;
                                endm
044D    FF              +       defb    255             ;
044E    FF              +       defb    255             ;
044F    FF              +       defb    255             ;
0450    FF              +       defb    255             ;
0451    FF              +       defb    255             ;
0452    FF              +       defb    255             ;
0453    FF              +       defb    255             ;
0454    FF              +       defb    255             ;
0455    FF              +       defb    255             ;
0456    FF              +       defb    255             ;
0457    FF              +       defb    255             ;
                        ;
                                reot    6               ; GAP II (6 bytes '00)
                                defb    0               ;
                                endm
0458    00              +       defb    0               ;
0459    00              +       defb    0               ;
045A    00              +       defb    0               ;
045B    00              +       defb    0               ;
045C    00              +       defb    0               ;
045D    00              +       defb    0               ;
                        ;
045E    FB                      defb    0fbh            ; Data Address Mark
                        ;
                                reot    256             ; User Data (256 bytes 'E5')
                                defb    0e5h            ;
                                endm
045F    E5              +       defb    0e5h            ;
0460    E5              +       defb    0e5h            ;
0461    E5              +       defb    0e5h            ;
0462    E5              +       defb    0e5h            ;
```

```
0463  E5          +          defb  0e5h      ;
0464  E5          +          defb  0e5h      ;
0465  E5          +          defb  0e5h      ;
0466  E5          +          defb  0e5h      ;
0467  E5          +          defb  0e5h      ;
0468  E5          +          defb  0e5h      ;
0469  E5          +          defb  0e5h      ;
046A  E5          +          defb  0e5h      ;
046B  E5          +          defb  0e5h      ;
046C  E5          +          defb  0e5h      ;
046D  E5          +          defb  0e5h      ;
046E  E5          +          defb  0e5h      ;
046F  E5          +          defb  0e5h      ;
0470  E5          +          defb  0e5h      ;
0471  E5          +          defb  0e5h      ;
0472  E5          +          defb  0e5h      ;
0473  E5          +          defb  0e5h      ;
0474  E5          +          defb  0e5h      ;
0475  E5          +          defb  0e5h      ;
0476  E5          +          defb  0e5h      ;
0477  E5          +          defb  0e5h      ;
0478  E5          +          defb  0e5h      ;
0479  E5          +          defb  0e5h      ;
047A  E5          +          defb  0e5h      ;
047B  E5          +          defb  0e5h      ;
047C  E5          +          defb  0e5h      ;
047D  E5          +          defb  0e5h      ;
047E  E5          +          defb  0e5h      ;
047F  E5          +          defb  0e5h      ;
0480  E5          +          defb  0e5h      ;
0481  E5          +          defb  0e5h      ;
0482  E5          +          defb  0e5h      ;
0483  E5          +          defb  0e5h      ;
0484  E5          +          defb  0e5h      ;
0485  E5          +          defb  0e5h      ;
0486  E5          +          defb  0e5h      ;
0487  E5          +          defb  0e5h      ;
0488  E5          +          defb  0e5h      ;
0489  E5          +          defb  0e5h      ;
048A  E5          +          defb  0e5h      ;
048B  E5          +          defb  0e5h      ;
048C  E5          +          defb  0e5h      ;
048D  E5          +          defb  0e5h      ;
048E  E5          +          defb  0e5h      ;
048F  E5          +          defb  0e5h      ;
0490  E5          +          defb  0e5h      ;
0491  E5          +          defb  0e5h      ;
0492  E5          +          defb  0e5h      ;
0493  E5          +          defb  0e5h      ;
0494  E5          +          defb  0e5h      ;
0495  E5          +          defb  0e5h      ;
0496  E5          +          defb  0e5h      ;
0497  E5          +          defb  0e5h      ;
0498  E5          +          defb  0e5h      ;
0499  E5          +          defb  0e5h      ;
049A  E5          +          defb  0e5h      ;
049B  E5          +          defb  0e5h      ;
049C  E5          +          defb  0e5h      ;
```

```
049D  E5          +         defb  0e5h      ;
049E  E5          +         defb  0e5h      ;
049F  E5          +         defb  0e5h      ;
04A0  E5          +         defb  0e5h      ;
04A1  E5          +         defb  0e5h      ;
04A2  E5          +         defb  0e5h      ;
04A3  E5          +         defb  0e5h      ;
04A4  E5          +         defb  0e5h      ;
04A5  E5          +         defb  0e5h      ;
04A6  E5          +         defb  0e5h      ;
04A7  E5          +         defb  0e5h      ;
04A8  E5          +         defb  0e5h      ;
04A9  E5          +         defb  0e5h      ;
04AA  E5          +         defb  0e5h      ;
04AB  E5          +         defb  0e5h      ;
04AC  E5          +         defb  0e5h      ;
04AD  E5          +         defb  0e5h      ;
04AE  E5          +         defb  0e5h      ;
04AF  E5          +         defb  0e5h      ;
04B0  E5          +         defb  0e5h      ;
04B1  E5          +         defb  0e5h      ;
04B2  E5          +         defb  0e5h      ;
04B3  E5          +         defb  0e5h      ;
04B4  E5          +         defb  0e5h      ;
04B5  E5          +         defb  0e5h      ;
04B6  E5          +         defb  0e5h      ;
04B7  E5          +         defb  0e5h      ;
04B8  E5          +         defb  0e5h      ;
04B9  E5          +         defb  0e5h      ;
04BA  E5          +         defb  0e5h      ;
04BB  E5          +         defb  0e5h      ;
04BC  E5          +         defb  0e5h      ;
04BD  E5          +         defb  0e5h      ;
04BE  E5          +         defb  0e5h      ;
04BF  E5          +         defb  0e5h      ;
04C0  E5          +         defb  0e5h      ;
04C1  E5          +         defb  0e5h      ;
04C2  E5          +         defb  0e5h      ;
04C3  E5          +         defb  0e5h      ;
04C4  E5          +         defb  0e5h      ;
04C5  E5          +         defb  0e5h      ;
04C6  E5          +         defb  0e5h      ;
04C7  E5          +         defb  0e5h      ;
04C8  E5          +         defb  0e5h      ;
04C9  E5          +         defb  0e5h      ;
04CA  E5          +         defb  0e5h      ;
04CB  E5          +         defb  0e5h      ;
04CC  E5          +         defb  0e5h      ;
04CD  E5          +         defb  0e5h      ;
04CE  E5          +         defb  0e5h      ;
04CF  E5          +         defb  0e5h      ;
04D0  E5          +         defb  0e5h      ;
04D1  E5          +         defb  0e5h      ;
04D2  E5          +         defb  0e5h      ;
04D3  E5          +         defb  0e5h      ;
04D4  E5          +         defb  0e5h      ;
04D5  E5          +         defb  0e5h      ;
04D6  E5          +         defb  0e5h      ;
```

```
04D7  E5          +         defb  0e5h      ;
04D8  E5          +         defb  0e5h      ;
04D9  E5          +         defb  0e5h      ;
04DA  E5          +         defb  0e5h      ;
04DB  E5          +         defb  0e5h      ;
04DC  E5          +         defb  0e5h      ;
04DD  E5          +         defb  0e5h      ;
04DE  E5          +         defb  0e5h      ;
04DF  E5          +         defb  0e5h      ;
04E0  E5          +         defb  0e5h      ;
04E1  E5          +         defb  0e5h      ;
04E2  E5          +         defb  0e5h      ;
04E3  E5          +         defb  0e5h      ;
04E4  E5          +         defb  0e5h      ;
04E5  E5          +         defb  0e5h      ;
04E6  E5          +         defb  0e5h      ;
04E7  E5          +         defb  0e5h      ;
04E8  E5          +         defb  0e5h      ;
04E9  E5          +         defb  0e5h      ;
04EA  E5          +         defb  0e5h      ;
04EB  E5          +         defb  0e5h      ;
04EC  E5          +         defb  0e5h      ;
04ED  E5          +         defb  0e5h      ;
04EE  E5          +         defb  0e5h      ;
04EF  E5          +         defb  0e5h      ;
04F0  E5          +         defb  0e5h      ;
04F1  E5          +         defb  0e5h      ;
04F2  E5          +         defb  0e5h      ;
04F3  E5          +         defb  0e5h      ;
04F4  E5          +         defb  0e5h      ;
04F5  E5          +         defb  0e5h      ;
04F6  E5          +         defb  0e5h      ;
04F7  E5          +         defb  0e5h      ;
04F8  E5          +         defb  0e5h      ;
04F9  E5          +         defb  0e5h      ;
04FA  E5          +         defb  0e5h      ;
04FB  E5          +         defb  0e5h      ;
04FC  E5          +         defb  0e5h      ;
04FD  E5          +         defb  0e5h      ;
04FE  E5          +         defb  0e5h      ;
04FF  E5          +         defb  0e5h      ;
0500  E5          +         defb  0e5h      ;
0501  E5          +         defb  0e5h      ;
0502  E5          +         defb  0e5h      ;
0503  E5          +         defb  0e5h      ;
0504  E5          +         defb  0e5h      ;
0505  E5          +         defb  0e5h      ;
0506  E5          +         defb  0e5h      ;
0507  E5          +         defb  0e5h      ;
0508  E5          +         defb  0e5h      ;
0509  E5          +         defb  0e5h      ;
050A  E5          +         defb  0e5h      ;
050B  E5          +         defb  0e5h      ;
050C  E5          +         defb  0e5h      ;
050D  E5          +         defb  0e5h      ;
050E  E5          +         defb  0e5h      ;
050F  E5          +         defb  0e5h      ;
0510  E5          +         defb  0e5h      ;
```

```
0511  E5          +          defb   0e5h          ;
0512  E5          +          defb   0e5h          ;
0513  E5          +          defb   0e5h          ;
0514  E5          +          defb   0e5h          ;
0515  E5          +          defb   0e5h          ;
0516  E5          +          defb   0e5h          ;
0517  E5          +          defb   0e5h          ;
0518  E5          +          defb   0e5h          ;
0519  E5          +          defb   0e5h          ;
051A  E5          +          defb   0e5h          ;
051B  E5          +          defb   0e5h          ;
051C  E5          +          defb   0e5h          ;
051D  E5          +          defb   0e5h          ;
051E  E5          +          defb   0e5h          ;
051F  E5          +          defb   0e5h          ;
0520  E5          +          defb   0e5h          ;
0521  E5          +          defb   0e5h          ;
0522  E5          +          defb   0e5h          ;
0523  E5          +          defb   0e5h          ;
0524  E5          +          defb   0e5h          ;
0525  E5          +          defb   0e5h          ;
0526  E5          +          defb   0e5h          ;
0527  E5          +          defb   0e5h          ;
0528  E5          +          defb   0e5h          ;
0529  E5          +          defb   0e5h          ;
052A  E5          +          defb   0e5h          ;
052B  E5          +          defb   0e5h          ;
052C  E5          +          defb   0e5h          ;
052D  E5          +          defb   0e5h          ;
052E  E5          +          defb   0e5h          ;
052F  E5          +          defb   0e5h          ;
0530  E5          +          defb   0e5h          ;
0531  E5          +          defb   0e5h          ;
0532  E5          +          defb   0e5h          ;
0533  E5          +          defb   0e5h          ;
0534  E5          +          defb   0e5h          ;
0535  E5          +          defb   0e5h          ;
0536  E5          +          defb   0e5h          ;
0537  E5          +          defb   0e5h          ;
0538  E5          +          defb   0e5h          ;
0539  E5          +          defb   0e5h          ;
053A  E5          +          defb   0e5h          ;
053B  E5          +          defb   0e5h          ;
053C  E5          +          defb   0e5h          ;
053D  E5          +          defb   0e5h          ;
053E  E5          +          defb   0e5h          ;
053F  E5          +          defb   0e5h          ;
0540  E5          +          defb   0e5h          ;
0541  E5          +          defb   0e5h          ;
0542  E5          +          defb   0e5h          ;
0543  E5          +          defb   0e5h          ;
0544  E5          +          defb   0e5h          ;
0545  E5          +          defb   0e5h          ;
0546  E5          +          defb   0e5h          ;
0547  E5          +          defb   0e5h          ;
0548  E5          +          defb   0e5h          ;
0549  E5          +          defb   0e5h          ;
0549  E5          +          defb   0e5h          ;
```

```
054B  E5          +         defb  0e5h        ;
054C  E5          +         defb  0e5h        ;
054D  E5          +         defb  0e5h        ;
054E  E5          +         defb  0e5h        ;
054F  E5          +         defb  0e5h        ;
0550  E5          +         defb  0e5h        ;
0551  E5          +         defb  0e5h        ;
0552  E5          +         defb  0e5h        ;
0553  E5          +         defb  0e5h        ;
0554  E5          +         defb  0e5h        ;
0555  E5          +         defb  0e5h        ;
0556  E5          +         defb  0e5h        ;
0557  E5          +         defb  0e5h        ;
0558  E5          +         defb  0e5h        ;
0559  E5          +         defb  0e5h        ;
055A  E5          +         defb  0e5h        ;
055B  E5          +         defb  0e5h        ;
055C  E5          +         defb  0e5h        ;
055D  E5          +         defb  0e5h        ;
055E  E5          +         defb  0e5h        ;
                        ;
055F  F7                    defb  0f7h        ; 2 CRC's written
                        ;
                            rept  14          ; GAP IV (14 bytes 'ff')
                            defb  255         ;
                            endm
0560  FF          +         defb  255         ;
0561  FF          +         defb  255         ;
0562  FF          +         defb  255         ;
0563  FF          +         defb  255         ;
0564  FF          +         defb  255         ;
0565  FF          +         defb  255         ;
0566  FF          +         defb  255         ;
0567  FF          +         defb  255         ;
0568  FF          +         defb  255         ;
0569  FF          +         defb  255         ;
056A  FF          +         defb  255         ;
056B  FF          +         defb  255         ;
056C  FF          +         defb  255         ;
056D  FF          +         defb  255         ;
                            ;
                            page
```

```
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*              FDDFORM entry point                    *
                        ;*                                                     *
                        ;*******************************************************
                        ;
056E                    fddform:
056E  11 028D                   ld      de,IniMsg       ; DE = Sign On and Mode message
0571  CD F01E                   call    print           ; print it
0574                    SelS.M:
0574  11 02BB                   ld      de,Quest0       ; print Msg & repeat
0577  CD F01E                   call    print           ; print it
057A  CD 0814                   call    RdiCBf          ; read a char
057D  3E 4D                     ld      a,'M'           ; default multiple
057F  28 13                     jr      z,Sel.MM        ; yes, set it
0581  1A                        ld      a,(de)          ; get answer
0582  CB AF                     res     5,a             ; convert to-case
0584  FE 4D                     cp      'M'             ; Multiple ?
0586  28 0C                     jr      z,Sel.MM        ; yes
0588  FE 53                     cp      'S'             ; Single ?
058A  11 03B7                   ld      de,Msg.SM       ; set Msg ptr
058D  28 08                     jr      z,Sel.SM        ; yes
058F  CD 087D                   call    errmp           ; beep & backspace
0592  18 E0                     jr      SelS.M          ; repeat
                        ;
0594                    Sel.MM:
0594  11 03C0                   ld      de,Msg.MM       ; Multiple Msg
0597                    Sel.SM:
0597  32 0127                   ld      (FMode),a       ; save mode
059A  CD F01E                   call    print           ; print mode
059D  CD 080E                   call    CrLF            ; and cr, lf
                        ;
                        ;
                        ;
05A0                    SelS.D:
05A0  11 02EC                   ld      de,Quest1       ; Side Msg
05A3  CD F01E                   call    print           ; print it
05A6  CD 0814                   call    RdiCBf          ; read a char
05A9  3E 53                     ld      a,'S'           ; default single
05AB  28 13                     jr      z,Sel.SS        ; yes, set it
05AD  1A                        ld      a,(de)          ; get answer
05AE  CB AF                     res     5,a             ; convert to-case
05B0  FE 53                     cp      'S'             ; Single ?
05B2  28 0C                     jr      z,Sel.SS        ; yes
05B4  FE 44                     cp      'D'             ; Double ?
05B6  11 03CB                   ld      de,Msg.DS       ; set Msg ptr
05B9  28 08                     jr      z,Sel.DS        ; yes
05BB  CD 087D                   call    errmp           ; beep & backspace
05BE  18 E0                     jr      SelS.D          ; repeat
                        ;
05C0                    Sel.SS:
05C0  11 03B7                   ld      de,Msg.SS       ; Single Side
05C3                    Sel.DS:
05C3  32 0128                   ld      (FSides),a      ; save Sides flag
05C6  CD F01E                   call    print           ; print single or double side
```

```
05C9  CD 080E                 call    CrLF            ; and cr, lf
                              ;
                              ;
                              ;
05CC                          Sel4.8:
05CC  11 031D                 ld      de,Quest2       ; 80 or 40 Msg
05CF  CD F01E                 call    print           ; print it
05D2  CD 0814                 call    RdlCBf          ; read a char
05D5  3E 34                   ld      a,'4'           ; default 40
05D7  28 13                   jr      z,Sel.40        ; yes, set it
05D9  1A                      ld      a,(de)          ; get answer
05DA  FE 34                   cp      '4'             ; 40 ?
05DC  28 0E                   jr      z,Sel.40        ; yes
05DE  FE 38                   cp      '8'             ; 80 ?
05E0  11 03D9                 ld      de,Msg.80       ; set Msg ptr
05E3  3E 50                   ld      a,80            ; 80 tracks
05E5  28 0A                   jr      z,Sel.80        ; yes
05E7  CD 087D                 call    errinp          ; beep & backspace
05EA  18 E0                   jr      Sel4.8          ; repeat
                              ;
05EC                          Sel.40:
05EC  11 03D4                 ld      de,Msg.40       ; 40 tracks
05EF  3E 28                   ld      a,40            ;
05F1                          Sel.80:
05F1  32 0129                 ld      (FTrks),a       ; save # trks
05F4  CD F01E                 call    print
05F7  CD 080E                 call    CrLF            ; and cr, lf
                              ;
                              ;
                              ;
05FA                          SelA.B:
05FA  11 0350                 ld      de,Quest3       ; A or B Msg
05FD  CD F01E                 call    print           ; print it
0600  CD 0814                 call    RdlCBf          ; read a char
0603  3E 42                   ld      a,'B'           ; default B
0605  28 13                   jr      z,Sel.B         ; yes, set it
0607  1A                      ld      a,(de)          ; get answer
0608  CB AF                   res     5,a             ; convert up-case
060A  FE 42                   cp      'B'             ; B ?
060C  28 0C                   jr      z,Sel.B         ; yes
060E  FE 41                   cp      'A'             ; A ?
0610  11 03DE                 ld      de,Msg.A        ; set Msg ptr
0613  28 08                   jr      z,Sel.A         ; yes
0615  CD 087D                 call    errinp          ; beep & backspace
0618  18 E0                   jr      SelA.B          ; repeat
                              ;
061A                          Sel.B:
061A  11 03E2                 ld      de,Msg.B        ; Drive B:
061D                          Sel.A:
061D  32 03A1                 ld      (disknum),a     ; set disk code
0620  CD F01E                 call    print
0623  CD 080E                 call    CrLF            ; and cr, lf
                              ;
0626                          SelYES:
0626  11 0381                 ld      de,Quest4       ; Confirm Msg
0629  CD F01E                 call    print           ; print it
062C  3E 03                   ld      a,3             ; 3 chars
062E  CD 0816                 call    RdCBuf          ; read answer
```

```
0631  FE 03                    cp      3               ; len('YES') ?
0633  C2 0772                  jp      nz,EndFmt       ; no, abort
0636  47                       ld      b,a             ; 3 char
0637  21 03E6                  ld      hl,MsgYES       ; check Msk
063A  1A                 CkYES:  ld      a,(de)          ; get a char
063B  CB AF                      res     5,a             ; up-case
063D  BE                         cp      (hl)            ; equal
063E  C2 0772                    jp      nz,EndFmt       ; no, abort
0641  10 F7                      djnz    CkYES           ; all char
                                 ;
                                 page
```

```
                        ;
                        ;*******************************************************
                        ;* F O R M                                             *
                        ;*     at this point all parameters are entried        *
                        ;*******************************************************
                        ;
0643                    FORM:
0643  21 0C23                 ld      hl,msg1add      ; message 1 cursor address
0646  11 03E9                 ld      de,fvisual      ;
0649  CD F024                 call    printat         ; print 'FORMATTING'
064C  21 0E1F                 ld      hl,msg2add      ; @msg2add
064F  11 0409                 ld      de,vsidtrk      ; print SIDE  TRACK msg addrs.
0652  CD F024                 call    printat         ;
0655  AF                      xor     a               ; clear accumulator
0656  32 0448                 ld      (ID.Trk),a      ; set track 0 in ID fields
0659  3A 03A1                 ld      a,(disknum)     ; load disk code
065C  D6 40                   sub     '@'             ; Disk A=1, disk B=2 and side 0
065E  32 0126                 ld      (DskSid),a      ; load disk and side para
0661  D3 D6                   out     (fddlch),a      ; select drive and side 0
0663  3E 02                   ld      a,fddrest       ; load fdd restore command code
0665  D3 D0                   out     (fddcmd),a      ; send out to 1771
                        ;
0667                    FmmNxt:
0667  AF                      xor     a               ; clear accumulator
0668  32 0449                 ld      (ID.Sid),a      ; set side 0 in ID fields
066B  3A 0126                 ld      a,(DskSid)      ; get drive code
066E  D3 D6             FmSd:  out     (fddlch),a      ; set side 0
0670                    FNxtSd:
0670  3E 01                   ld      a,1             ; A = 1
0672  32 044A                 ld      (ID.Sec),a      ; set sector 1
0675  CD 07EE                 call    waitfd          ; wait until end command
0678  B7                      or      a               ; zero in accumulator ?
0679  C2 07B3                 jp      nz,timeout      ; if no zero then disk offline
                        ;
067C  CD 07B8                 call    sidtrkvis       ; visualize Track and Side
067F  21 0419                 ld      hl,preamble     ; H.L = preamble table (40 bytes 'ff')
0682  01 28D7                 ld      bc,0+(40 * 256) + fdddat; B= 40 bytes counter
                                                      ; C = fdd data register
                        ;
0685  3E F4                   ld      a,fdwtrk        ; load write track command code
0687  D3 D0                   out     (fddcmd),a      ; send out to 1771
0689  CD 07E9                 call    fddelay         ; wait aproax 56 micros
                        ;
068C                    wtpreamb:
068C  DB D0                   in      a,(fdsts)       ; load fdd status
068E  CB 4F                   bit     1,a             ; test DRQ bit
0690  28 FA                   jr      z,wtpreamb      ; wait if no DRQ
0692  ED A3                   outi                    ; write one byte
0694  20 F6                   jr      nz,wtpreamb     ; repeat until all 40 bytes are output
                        ;
0696                    FmSect:
0696  21 0441                 ld      hl,idfield      ; H.L = ID field data table
                        ;
                              ; total byte to output are 301.
                        ;
0699  06 00                   ld      b,0             ; 256 bytes to 1771
```

```
069B                    wtidfield:
069B  DB D0                     in      a,(fddsts)      ; load fdd status
069D  CB 4F                     bit     1,a             ; test DRQ bit
069F  28 FA                     jr      z,wtidfield     ; wait if no DRQ
06A1  ED A3                     outi                    ; write one byte
06A3  20 F6                     jr      nz,wtidfield    ; repeat until all 256 bytes are outout
                        ;
06A5  06 2D                     ld      b,301-256       ; load rimanents  bytes to 1771
06A7                    wtlidfield:
06A7  DB D0                     in      a,(fddsts)      ; load fdd status
06A9  CB 4F                     bit     1,a             ; test DRQ bit
06AB  28 FA                     jr      z,wtlidfield    ; wait if no DRQ
06AD  ED A3                     outi                    ; write one byte
06AF  20 F6                     jr      nz,wtlidfield   ; repeat until all 45 bytes are outout
                        ;
                                ; now, all 301 bytes are output to 1771
                        ;
06B1  3A 044A                   ld      a,(ID.Sec)      ; load actual sector number
06B4  3C                        inc     a               ; inc. sector number
06B5  32 044A                   ld      (ID.Sec),a      ; set next sector number
06B8  FE 08                     cp      11              ; last sector has been written ?
06BA  C2 0696                   jp      nz,FmSect       ; no, then write next sector
06BD                    endtrk:
06BD  DB D0                     in      a,(fddsts)      ; load fdd status
06BF  CB 47                     bit     0,a             ; end track ?
06C1  28 07                     jr      z,NextSid       ; yes, then go to next track
06C3  3E FF                     ld      a,255           ; load byte 'ff'
06C5  D3 D7                     out     (fddat),a       ; write to 1771 data register
06C7  C3 06BD                   jp      endtrk          ; repeat until end track
06CA                    NextSid:
06CA  DB D0                     in      a,(fddsts)      ; load fdd status
06CC  E6 E7                     and     11100111b       ; write track error?
06CE  C2 07AE                   jp      nz,fdforerr     ; yes, then goto error
06D1  21 0449                   ld      hl,ID.Sid       ; point current side
06D4  3A 0128                   ld      a,(Fsides)      ;
06D7  FE 53                     cp      'S'             ; Single Side ?
06D9  CA 06E8                   jp      z,NextTrk       ; yes, next track
06DC  AF                        xor     a               ; zero on a
06DD  B6                        or      (hl)            ; check with current side
06DE  20 08                     jr      nz,NextTrk      ; side 1, then next trk
06E0  34                        inc     (hl)            ; else set side 1
06E1  3A 0126                   ld      a,(DskSid)      ; load disk number
06E4  F6 20                     or      00100000b       ; set side one
06E6  18 B6                     jr      FmSd            ; format side 1
                        ;
06E8                    NextTrk:
06E8  21 0129                   ld      hl,FTrks        ; point to # of tracks
06EB  3A 0448                   ld      a,(ID.Trk)      ; load current track #
06EE  3C                        inc     a               ; point to next track
06EF  BE                        cp      (hl)            ; end track ?
06F0  28 0A                     jr      z,dskverify     ; yes, then go to end
06F2  32 0448                   ld      (ID.Trk),a      ; set next track
                        ;
06F5  3E 52                     ld      a,fddsin        ; load fdd step in
06F7  D3 D0                     out     (fddcmd),a      ; send out to 1771
06F9  C3 0667                   jp      FmmNxt          ; count for next track
                        ;
                        ;
```

```
                        ;*******************************************************
                        ;* V E R I F Y                                         *
                        ;*     Check all sector on disk                        *
                        ;*******************************************************
                        ;
06FC                    dskverify:
06FC  21 0C23                   ld      hl,msg1add      ; message 1 cursor address
06FF  11 03F9                   ld      de,vvisual      ; DE -) verify message
0702  CD F024                   call    printat         ; print it at HL
0705  AF                        xor     a               ; start with
0706  32 0235                   ld      (VerTrk),a      ;       track 00
0709                    VNxTrk:
0709  3A 0126                   ld      a,(DskSid)      ; load disk and side
070C                    VNxSd:
070C  3D                        dec     a               ; disk A = 0, B = 1
070D  32 0234                   ld      (VerDsk),a      ; start disk and side
                                ;
0710  3E 01                     ld      a,1             ;
0712  32 0237                   ld      (VerSec),a      ; start sector 1
0715  11 012A                   ld      de,xlt          ; DE -) translate table
                        ;
0718                    vervis:
0718  3A 0234                   ld      a,(VerDsk)      ; load verify disk and side
071B  E6 10                     and     00010000b       ; mask side
071D  28 02                     jr      z,verv00        ; jmp if side 0
071F  3E 01                     ld      a,1             ; set side 1
0721                    verv00:
0721  D5                        push    de              ; save xlt pointer
0722  32 0449                   ld      (ID.Sid),a      ; set ID side for visualization
0725  3A 0235                   ld      a,(VerTrk)      ; load verify track (only low byte)
0728  32 0448                   ld      (ID.Trk),a      ; set ID side for visualization
072B  CD 07B8                   call    sidtrkvis       ; visualize side and track
                        ;
072E  21 0234                   ld      hl,vertab       ; verify tab para
0731  CD F015                   call    fdiod           ; read 256 bytes
0734  B7                        or      a               ; read error ?
0735  20 5D                     jr      nz,readerr      ; yes, abort
                                ;
0737  21 0134                   ld      hl,verbuff      ; HL -) verify buffer
073A  06 00                     ld      b,0             ; num. byte to verify (256)
073C                    verif00:
073C  7E                        ld      a,(hl)          ; load one byte
073D  FE E5                     cp      0e5h            ; IBM standard data ?
073F  20 68                     jr      nz,vererr       ; yes, then count for next data
0741  23                        inc     hl              ; point to next data
0742  10 F8                     djnz    verif00         ; loop until finished
                        ;
0744                    endverify:
0744  D1                        pop     de              ; DE -) xlt pointer
0745  1A                        ld      a,(de)          ; load verify sector number
0746  13                        inc     de              ; point to next sector
0747  32 0237                   ld      (VerSec),a      ; set next sector
074A  B7                        or      a               ; end track ?
074B  20 CB                     jr      nz,vervis       ; no, then count
074D  3A 0128                   ld      a,(Fsides)      ; get fmt mode
0750  FE 53                     cp      'S'             ; Single Side ?
0752  CA 0764                   jp      z,VerNTrk       ; yes, next track
0755  3A 0234                   ld      a,(VerDsk)      ; get side & disk
```

```
0758  E6 10                     and     00010000b       ; mask side
075A  20 08                     jr      nz,VerNTrk      ; side 1, then next trk
075C  3A 0126                   ld      a,(DskSid)      ; get disk code
075F  F6 10                     or      00010000b       ; set side 1
0761  C3 070C                   jp      VNxSd           ; verify side 1
0764                    VerNTrk:
0764  21 0129                   ld      hl,FTrks        ; point to # of tracks
0767  3A 0235                   ld      a,(VerTrk)      ; load current track #
076A  3C                        inc     a               ; point to next track
076B  BE                        cp      (hl)            ; end track ?
076C  32 0235                   ld      (VerTrk),a      ; set next track
076F  C2 0709                   jp      nz,VNxTrk       ; repeat if not last track
                                ;
                                ; End of Formatting
                                ; short delay and go to format another
0772                    EndFmt:
0772  06 02                     ld      b,2             ; set soft timer 2
0774                    endan00:
0774  11 0000                   ld      de,0            ; set soft timer 1
0777                    endan11:
0777  1B                        dec     de              ; timer 1 down
0778  7A                        ld      a,d             ;
0779  B3                        or      e               ; zero for timer 1 ?
077A  20 FB                     jr      nz,endan11      ; no, then count
077C  10 F6                     djnz    endan00         ; timer 2 down and repeat until timer 2 = 0
077E                    GoFmt:
077E  3A 0127                   ld      a,(FMode)       ; check for Single or Multiple Mode
0781  FE 53                     cp      'S'             ; Single
0783  CA 056E                   jp      z,fdform        ;       start from beggining
0786  21 0700                   ld      hl,Confadd      ; Cursor @ 8,1
0789  CD F027                   call    movcurs         ; move
078C  0E 06                     ld      c,6             ; clear to end of screen
078E  CD F006                   call    cout            ; do it
0791  C3 0626                   jp      SelYES          ; else Multiple
                        ;
                        ;
0794                    readerr:
0794  11 0259                   ld      de,rdermsg      ; DE -) read error message
0797                    rder00:
0797  21 1019                   ld      hl,ermsgadd     ; error message address
079A  CD F024                   call    printat         ; print it
079D  11 027D                   ld      de,wterr        ; print 'Hit any key'
07A0  CD F01E                   call    print
07A3  CD F003                   call    cin
07A6  D1                        pop     de              ; restore xlt pointer
07A7  18 D5                     jr      GoFmt           ; go to format
                        ;
                        ;
07A9                    vererr:
07A9  11 026B                   ld      de,verrmsg      ; DE -) verify error message
07AC  18 E9                     jr      rder00          ; print it and retry
                        ;
                        ;
07AE                    fdforerr:
07AE  11 024A                   ld      de,fmtermsg     ; DE = format error message
07B1  18 E4                     jr      rder00          ; print it
                        ;
07B3                    timeout:
```

```
07B3  11 023B                   ld      de,oflmsg       ; DE = offline message
07B6  18 DF                     jr      rder00          ; print it
                        ;
                        ;
                        ; Print current Track # and Side #
                        ;
                        ;
07B8                    sidtrkvis:
                                ; visualize track and side
07B8  21 0E25                   ld      hl,tnumadd      ; track video adrr.
07BB  CD F027                   call    movcurs         ; move cursor
07BE  3A 0448                   ld      a,(ID.Trk)      ; load ID track number
07C1  06 FF                     ld      b,255           ; set decimal counter
07C3                    sidtr00:
07C3  04                        inc     b               ; inc. decimal digit
07C4  D6 0A                     sub     10              ; A=A-10
07C6  30 FB                     jr      nc,sidtr00      ; count if A )= 0
07C8  C6 0A                     add     a,10            ; at this point A = lsd, B = msd (BCD)
07CA  F5                        push    af              ; save lsd
07CB  78                        ld      a,b             ; load msd
07CC  C6 30                     add     a,'0'           ; convert ascii
07CE  4F                        ld      c,a             ;
07CF  CD F006                   call    cout            ; print msd
07D2  F1                        pop     af              ;
07D3  C6 30                     add     a,'0'           ; convert ascii
07D5  4F                        ld      c,a             ;
07D6  CD F006                   call    cout            ; print lsd
07D9  21 0E2F                   ld      hl,snumadd      ; side video addr.
07DC  CD F027                   call    movcurs         ; move cursor
07DF  3A 0449                   ld      a,(ID.Sid)      ; load ID side number
07E2  C6 30                     add     a,'0'           ; convert ASCII
07E4  4F                        ld      c,a             ;
07E5  CD F006                   call    cout            ; print side 0 or 1
07E8  C9                        ret                     ; and ret
                        ;
                        ;
07E9                    fddelay:
                                rept    4
                                ex      (sp),hl         ; delay beetwen write command reg.
                                endm                    ; to read status reg.
07E9  E3            +           ex      (sp),hl         ; delay beetwen write command reg.
07EA  E3            +           ex      (sp),hl         ; delay beetwen write command reg.
07EB  E3            +           ex      (sp),hl         ; delay beetwen write command reg.
07EC  E3            +           ex      (sp),hl         ; delay beetwen write command reg.
07ED  C9                        ret
                        ;
                        ;
07EE                    waitfd:
                                ; wait until fdd busy is reset.
07EE  CD 07E9                   call    fddelay         ; wait aproax 56 microS
07F1  06 02                     ld      b,2             ; set soft timer
07F3                    wait00:
07F3  11 0000                   ld      de,0            ; for aproax five seconds
07F6                    wait01:
07F6  DB D0                     in      a,(fdcsts)      ; input to fdd status
07F8  CB 47                     bit     0,a             ; test busy bit
07FA  28 0F                     jr      z,wait02        ; jump if no command is in progress
07FC  1B                        dec     de              ;
```

```
07FD  7A                    ld      a,d             ; timer down
07FE  B3                    or      e               ;
07FF  20 F5                 jr      nz,wait01       ;
0801  05                    dec     b               ;
0802  20 EF                 jr      nz,wait00       ; time out
0804                  ofline:
0804  3E D0                 ld      a,fddrst        ; reset fdd controller
0806  D3 D0                 out     (fddcmd),a      ; exec. command
0808  3E 01                 ld      a,00000001b     ; set time-out bit error
080A  C9                    ret                     ; and ret
080B                  wait02:
080B  47                    ld      b,a             ; save fdd status in B register
080C  AF                    xor     a               ; clear accumulator for
080D  C9                    ret                     ; normal return
                      ;
                      ;
                      ; ************************************
                      ; * Print CR, LF & Attr Off          *
                      ; ************************************
                      ;
080E                  CrLf:
080E  11 02B6               ld      de,CrLfOF       ; print
0811  C3 F01E               jp      print           ; & ret
                      ;
                      ;
                      ; ************************************
                      ; * Console input function           *
                      ; ************************************
                      ; entry with a=maxchar
                      ;
0814  3E 01           RdlCBf: ld      a,1
                      ;
0816                  RdCBuf:
0816  21 0887               ld      hl,CBuf         ; point to console buffer
0819  77                    ld      (hl),a          ; # of char
081A  23                    inc     hl
081B  36 00                 ld      (hl),0          ; set 0 char readed
081D  23                    inc     hl
081E  E5                    push    hl              ; save
081F                  WaitKey:
081F  CD F003               call    cin             ; wait a char
0822  FE 03                 cp      'C'-'@'         ; Is it a ^C ?
0824  CA 0000               jp      z,0             ; yes, abort
0827  FE 0D                 cp      cr              ; Is it a CR ?
0829  28 3A                 jr      z,endino        ;
082B  FE 08                 cp      backsp          ; Is it a BS ?
082D  20 1A                 jr      nz,cont
082F  3E 00                 ld      a,0
0831  21 0888               ld      hl,CBuf+1       ; point to chars
0834  B6                    or      (hl)            ; there is a char ?
0835  28 27                 jr      z,Cbeo          ; no, error
0837  35              Bs1:  dec     (hl)            ; dec # of char
0838  D1                    pop     de              ; restore pointer
0839  1B                    dec     de              ; back step
083A  D5                    push    de              ; save
083B  0E 08                 ld      c,backsp        ; print backspace
083D  CD F006               call    cout            ;
0840  0E 20                 ld      c,space         ; print space
```

```
0842  CD F006                call    cout            ;
0845  0E 08                   ld      c,backsp        ; print backspace and
0847  18 17                   jr      CCout           ; wait another key
                              ;
0849  FE 20           cont:   cp      ' '             ; (' '
084B  38 D2                   jr      c,WaitKey       ;
084D  4F                      ld      c,a             ; save char
084E  21 0888                 ld      hl,Cbuf+1       ; point to #char
0851  7E                      ld      a,(hl)          ; get it
0852  2B                      dec     hl              ; point to maxchar
0853  BE                      cp      (hl)            ;
0854  30 08                   jr      nc,Cbeep        ; error if )=max
0856  23                      inc     hl              ; point to #char
0857  34                      inc     (hl)            ; increment
0858  E1                      pop     hl              ; restore buffer pointer
0859  71                      ld      (hl),c          ; save char
085A  23                      inc     hl
085B  E5                      push    hl              ; resave
085C  18 02                   jr      CCout           ; print it
                              ;
085E  0E 07           Cbeep:  ld      c,bell          ; print bell
                              ;
0860                  CCout:                          ; Print char
0860  CD F006                 call    cout            ; & return
0863  18 BA                   jr      WaitKey         ; to wait another key
                              ;
0865                  EndImp:
0865  E1                      pop     hl              ; buff ptr
0866  11 0889                 ld      de,Cbuf+2       ; point to char
0869  3A 0888                 ld      a,(Cbuf+1)      ; get # of char
086C  B7                      or      a               ; set flag
086D  C8                      ret     z               ; no char ret
086E  F5                      push    af              ; save # char
086F  D5                      push    de              ;
0870  47                      ld      b,a             ; on b
0871  0E 08                   ld      c,backsp        ; back step
0873  C5              Bstep:  push    bc              ; save
0874  CD F006                 call    cout            ; print backspace
0877  C1                      pop     bc              ; restore
0878  10 F9                   djnz    Bstep           ; loop
087A  D1                      pop     de              ;
087B  F1                      pop     af              ; restore
087C  C9                      ret                     ; done
                      ;
                      ;
                      ; ***************************************
                      ; * error on input                      *
                      ; ***************************************
                      ;
087D                  errimp:
087D  0E 07                   ld      c,bell          ; beep
087F  CD F006                 call    cout            ;       on console
0882  0E 0D                   ld      c,cr            ; start of line
0884  C3 F006                 jp      cout            ;
                      ;
                      ; Input Buffer
                      ;
                      ;
```

```
0887  01              CBuf:   defb    1               ; only 1 char
0888  00                      defb    0               ; char read
0889                          defs    5               ; chars space
                              :
                              :
                              end     100h            ; end of floppy disk formatter
```

Macros:

Symbols:

BACKSP	0008	BELL	0007	BOOTRD	F021	BS1	0837
BSTEP	0873	CBEP	085E	CBUF	0887	CCOUT	0860
CIN	F003	CKYES	063A	COMPFL	F02D	CONFAD	0700
CONT	0849	COPYRI	0000	COUT	F006	CR	000D
CRLF	080E	CRLFOF	02B6	CSTS	F009	DISKNU	03A1
DSKSID	0126	DSKVER	06FC	ENDAN0	0774	ENDAN1	0777
ENDFMT	0772	ENDINP	0865	ENDMSG	0024	ENDTRK	06BD
ENDVER	0744	ERMSGA	1019	ERRINP	087D	FALSE	0000
FDBUSY	0001	FDDCMD	00D0	FDDDAT	00D7	FDDELA	07E9
FDDFOR	056E	FDDLCH	00D6	FDDRES	0002	FDDR0	0002
FDDRST	00D0	FDDSEC	00D2	FDDSIN	0052	FDDSTS	00D0
FDDTAB	0419	FDDTRK	00D1	FDDWTR	00F4	FDFORE	07AE
FDIOD	F015	FDIOS	F012	FDNRDY	0080	FDT1ER	0018
FDT23E	001F	FDWPRT	0040	FFEED	000C	FLASH	0043
FMMNXT	0667	FMODE	0127	FMSD	066E	FMSECT	0696
FMTERM	024A	FNXTSD	0670	FORM	0643	FSIDES	0128
FTRKS	0129	FVISUA	03E9	GOFMT	077E	ID.SEC	044A
ID.SID	0449	ID.TRK	0448	IDFIEL	0441	INIMSG	028D
LF	000A	LOUT	F00C	LSTS	F00F	MOVCUR	F027
MSG.40	03D4	MSG.80	03D9	MSG.A	03DE	MSG.B	03E2
MSG.DS	03CB	MSG.MM	03C0	MSG.SM	03B7	MSG.SS	03B7
MSG1AD	0C23	MSG2AD	0E1F	MSGYES	03E6	NEXTSI	06CA
NEXTTR	06E8	NORM	0040	OFLINE	0804	OFLMSG	023B
PFX	0013	PREAMB	0419	PRINT	F01E	PRINTA	F024
QUESTO	02BB	QUEST1	02EC	QUEST2	031D	QUEST3	0350
QUEST4	0381	RD1CBF	0814	RDCBUF	0816	RDER00	0797
RDERMS	0259	READER	0794	REVER	0042	ROM	F000
SEL.40	05EC	SEL.80	05F1	SEL.A	061D	SEL.B	061A
SEL.DS	05C3	SEL.MM	0594	SEL.SM	0597	SEL.SS	05C0
SEL4.8	05CC	SELA.B	05FA	SELS.D	05A0	SELS.M	0574
SELYES	0626	SIDTR0	07C3	SIDTRK	07B8	SNUMAD	0E2F
SPACE	0020	STACK	0126	STROUT	F01E	TAB	0009
TIMEDU	07B3	TNUMAD	0E25	TRUE	0001	VERBUF	0134
VERDMA	0238	VERDSK	0234	VERERR	07A9	VERIFO	073C
VERNTR	0764	VEROP	023A	VERRMS	026B	VERSEC	0237
VERTAB	0234	VERTRK	0235	VERV00	0721	VERVIS	0718
VIDINI	F02A	VNXSD	070C	VNXTRK	0709	VSIDTR	0409
VVISUA	03F9	WAIT00	07F3	WAIT01	07F6	WAIT02	080B
WAITFD	07EE	WAITKE	081F	WDINI	F018	WDIO	F018
WT1IDF	06A7	WTERR	027D	WTIDFI	0698	WTPREA	068C
XLT	012A						

No Fatal error(s)