

```

;
;
;
;*****
;*
;*   F L O P P Y   D I S K   F O R M A T T E R   *
;*   This program format floppy disk with      *
;*   10 sector with 256 byte/sector            *
;*   in single or double side                  *
;*                                              *
;*****
;
;
;
;   title      Floppy Disk Formatter For NE CP/M 2.2
;   subttl     Copyright 1986 by Costantino Haritakis - Last rev. 06/01/86 14:00
;
;
;
0000'      aseq
           .z80           ; this program is written in z80 mnemonics
;
0001      true   equ    1
0000      false  equ    0
;
0001      copyrig equ    true
;
           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      CompFlg  equ      rom+45     ; Version Number
C      ;
C      ;
C      Include ASCII.LIB
C      ;
C      ;
C      ;*****
C      ;*
C      ;*      ASCII EQUIVALENTS
C      ;*
C      ;*****
C      ;
0007   C      bell     equ      'G'-'@'    ; 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      pfx      equ      'S'-'@'    ; attributes pfx
0042   C      rever    equ      'B'        ; Reverse On (^SB)
0043   C      flash    equ      'C'        ; Flash On (^SC)
0040   C      norm     equ      '@'        ; Normal (^SE)
0020   C      space    equ      ' '        ; space char.
0024   C      endmsg   equ      '$'        ; end of print message
C      ;
C      ;
page
    
```

```

;
;*****
;*
;*          FD 1771 I/O port
;*
;*****
00D0 fddsts equ 0d0h ; fdd status port
00D1 fddtrk equ 0d1h ; fdd track port
00D2 fddsec equ 0d2h ; fdd sector port
00D6 fddlch equ 0d6h ; fdd lach port
00D7 fdddat equ 0d7h ; fdd data port
00D0 fddcmd equ fddsts ; fdd command port
;
;*****
;*
;*          FD 1771 Command Summary
;*
;*****
;
;      ; this command are without verify because disk don't are formatted
;
0002 fddrest equ 00000010b ; fdd restore command code
00D0 fddrst equ 11010000b ; fdd reset int. command code
0052 fddsin equ 01010010b ; fdd step in command code
00F4 fddwtrk 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
001B fdt1er equ 018h ; mask error for type I commands
001F fdt23er equ 01fh ; mask error for type II and type III commands
;
page
    
```

```

;
;*****
;*          Message cursor position          *
;*****
;
1719      ermsgadd equ    1719h          ; error msg position (X=25 Y=14)
0F20      msgladd equ    0f20h          ; form/ver msg position (X=36 Y=10)
111D      msg2add equ    111dh          ; sid/trk msg position (X=33 Y=12)
1122      snumadd equ    1122h          ; sid num. position (X=38 Y=12)
112B      tnumadd equ    112bh          ; trk num. position (X=48 Y=12)
;
;
;*****
;*          F D D F O R M                    *
;*          *                                *
;*          *                                *
;*****
;
0000      Aseg
          Org    100h
;
0100      31 015A      ld    sp,stack      ; set stack pointer
0103      C3 0792      jp     fddform      ; go to floppy format
;
;
;*****
;*          Ram data areas                    *
;*****
;
;
          if    copyrig
0106      20 43 4F 50      defb    ' COPYRIGHT (c) 1986 by Costantino Haritakis - ITALY '
010A      59 52 49 47
010E      48 54 20 28
0112      63 29 20 31
0116      39 38 36 20
011A      62 79 20 43
011E      6F 73 74 61
0122      6E 74 69 6E
0126      6F 20 48 61
012A      72 69 74 61
012E      6B 69 73 20
0132      2D 20 49 54
0136      41 4C 59 20
          endif
;
013A      defs    32          ; stack data areas
015A      stack equ    $
;
015A      dsksid:
          defs    1          ; byte to output lach
015B      fsideflag:
          defs    1          ; format side flagh
015B      vsideflag:
015C

```

```

015C                defs 1                ; verify side flag
015D                verflag:
015D                defs 1                ; verify flag (Y or N)
                                ;
015E                xlt:
015E    05 09 03 07                defb 5,9,3,7,2,6,10,4,8,0
0162    02 06 0A 04
0166    08 00

                                ;
0168                verbuff:
0168                defs 256            ; verify buffer (256 byte)
                                ;
                                ;
0268                vertab:                ;costas era 0 diventa 1
0268                verdsk: defs 1        ; verify disk and side
0269    0000                vertrk: defw 0    ; verify track number
026B    01                versec: defb 1    ; verify sector number
026C    0168                verdma: defw verbuff ; verify dma address
026E    00                verop: defb 0    ; verify = read operation
                                ;
                                page
    
```

```

;
;*****
;*          Error Messages          *
;*****
;
setnew:
    defb    cr,lf,'Set new system diskette in drive A,'

026F    0D 0A 53 65
0273    74 20 6E 65
0277    77 20 73 79
027B    73 74 65 6D
027F    20 64 69 73
0283    6B 65 74 74
0287    65 20 69 6E
028B    20 64 72 69
028F    76 65 20 41
0293    2C
0294    0D 0A 74 6B
0298    65 6E 20 70
029C    75 73 68 20
02A0    61 6E 79 20
02A4    6B 65 79 2E
02A8    0D 0A 24

;
oflmsg:
    defb    cr,lf,'Disk offline',endmsg

02AB    0D 0A 44 69
02AF    73 6B 20 6F
02B3    66 66 6C 69
02B7    6E 65 24

;
fntermmsg:
    defb    cr,lf,'Format error',endmsg

02BA    0D 0A 46 6F
02BE    72 6D 61 74
02C2    20 65 72 72
02C6    6F 72 24

;
rdrmsg:
    defb    'Verify read error',endmsg

02C9    56 65 72 69
02CD    66 79 20 72
02D1    65 61 64 20
02D5    65 72 72 6F
02D9    72 24

;
verrmsg:
    defb    'Verify data error',endmsg

02DB    56 65 72 69
02DF    66 79 20 64
02E3    61 74 61 20
02E7    65 72 72 6F
02EB    72 24

;
wterr:
    defb    ' - Hit any key ',endmsg

02ED    20 2D 20 4B
02F1    69 74 20 61
02F5    6E 79 20 6B
02F9    65 79 20 24

;
page
    
```

```

02FD                                ;
02FD 0C 20 20 20                    inmsg:
',pfx,norm,cr,lf,cr,lf,endmsg      defb  ffeed,'          ',pfx,rever,'  Floppy Disk formatter vers 3.0  C. Haritakis
0301 20 20 20 20
0305 20 20 20 20
0309 13 42 20 20
030D 20 46 6C 6F
0311 70 70 79 20
0315 44 69 73 6B
0319 20 66 6F 72
031D 6D 61 74 74
0321 65 72 20 76
0325 65 72 73 20
0329 33 2E 30 20
032D 20 20 20 20
0331 43 2E 20 48
0335 61 72 69 74
0339 61 6B 69 73
033D 20 20 20 13
0341 40 0D 0A 0D
0345 0A 24

0347                                ;
0347 D4                             quest1:
                                defb  0d4h
                                rept 76
                                .XLIST
                                0cch          ;graphic line
                                .LIST
                                endm

0394 D5 0D 0A                      defb  0d5h,cr,lf
0397 CB 20 20 5B                    defb  0cbh,' [1]. A  Single Format          [2]. A  Double Format          ',
0cbh,cr,lf
039B 31 5D 2E 20
039F 20 41 20 20
03A3 20 53 69 6E
03A7 67 6C 65 20
03AB 46 6F 72 6D
03AF 61 74 20 20
03B3 20 20 20 20
03B7 20 20 20 20
03BB 20 20 20 20
03BF 20 5B 32 5D
03C3 2E 20 20 41
03C7 20 20 20 44
03CB 6F 75 62 6C
03CF 65 20 46 6F
03D3 72 6D 61 74
03D7 20 20 20 20
03DB 20 20 20 20
03DF 20 20 20 20
03E3 20 CB 0D 0A
03E7 CB 20 20 5B                    defb  0cbh,' [3]. B  Single Format          [4]. B  Double Format          ',
0cbh,cr,lf
03EB 33 5D 2E 20
03EF 20 42 20 20
03F3 20 53 69 6E
03F7 67 6C 65 20
03FB 46 6F 72 6D

```

03FF 61 74 20 20  
0403 20 20 20 20  
0407 20 20 20 20  
040B 20 20 20 20  
040F 20 5B 34 5D  
0413 2E 20 20 42  
0417 20 20 20 44  
041B 6F 75 62 6C  
041F 65 20 46 6F  
0423 72 6D 61 74  
0427 20 20 20 20  
042B 20 20 20 20  
042F 20 20 20 20  
0433 20 CB 0D 0A  
0437 CB 20 20 5B

defb 0cbh,' [5]. A Single Format and Verify [6]. A Double Format and Verify ',

0cbh,cr,lf

043B 35 5D 2E 20  
043F 20 41 20 20  
0443 20 53 69 6E  
0447 67 6C 65 20  
044B 46 6F 72 6D  
044F 61 74 20 61  
0453 6E 64 20 56  
0457 65 72 69 66  
045B 79 20 20 20  
045F 20 5B 36 5D  
0463 2E 20 20 41  
0467 20 20 20 44  
046B 6F 75 62 6C  
046F 65 20 46 6F  
0473 72 6D 61 74  
0477 20 61 6E 64  
047B 20 56 65 72  
047F 69 66 79 20  
0483 20 CB 0D 0A  
0487 CB 20 20 5B

defb 0cbh,' [7]. B Single Format and Verify [8]. B Double Format and Verify ',

0cbh,cr,lf

048B 37 5D 2E 20  
048F 20 42 20 20  
0493 20 53 69 6E  
0497 67 6C 65 20  
049B 46 6F 72 6D  
049F 61 74 20 61  
04A3 6E 64 20 56  
04A7 65 72 69 66  
04AB 79 20 20 20  
04AF 20 5B 38 5D  
04B3 2E 20 20 42  
04B7 20 20 20 44  
04BB 6F 75 62 6C  
04BF 65 20 46 6F  
04C3 72 6D 61 74  
04C7 20 61 6E 64  
04CB 20 56 65 72  
04CF 69 66 79 20  
04D3 20 CB 0D 0A  
04D7 CB 20 20 20

defb 0cbh,'

CTR - C from EXIT

0cbh,cr,lf

04DB 20 20 20 20



04E7 20 20 20 20  
 04EB 20 20 20 20  
 04EF 20 20 20 20  
 04F3 20 20 43 54  
 04F7 52 20 20 20  
 04FB 43 20 66 72  
 04FF 6F 6D 20 45  
 0503 58 49 54 20  
 0507 20 20 20 20  
 050B 20 20 20 20  
 050F 20 20 20 20  
 0513 20 20 20 20  
 0517 20 20 20 20  
 051B 20 20 20 20  
 051F 20 20 20 20  
 0523 20 CB 0D 0A  
 0527 D2

```

      defb 0d2h
rept 76
.XLIST
      0cch          ;graphic line
.LIST
endm
      defb 0d3h,cr,lf,lf
      defb '        Choose from previous ones ... ?,08,20,'1',endmsg

```

0574 D3 0D 0A 0A  
 0578 20 20 20 20  
 057C 20 20 20 43  
 0580 6B 6F 6F 73  
 0584 65 20 66 72  
 0588 6F 6D 20 70  
 058C 72 65 76 69  
 0590 75 73 20 6F  
 0594 6E 65 73 20  
 0598 2E 2E 2E 20  
 059C 3F 08 14 31  
 05A0 24

```

;
quest3:
      defb cr,lf,lf,bell,'Are you sure for disk ',19,48h

```

05A1 0D 0A 0A 07  
 05A5 41 72 65 20  
 05A9 79 6F 75 20  
 05AD 73 75 72 65  
 05B1 20 66 6F 72  
 05B5 20 64 69 73  
 05B9 6B 20 13 48

```

disknum:
      defb 0
      defb 19,40h,' (Y or any key for No) ? ',endmsg

```

05BD 00  
 05BE 13 40 20 28  
 05C2 59 20 6F 72  
 05C6 20 61 6E 79  
 05CA 20 6B 65 79  
 05CE 20 66 6F 72  
 05D2 20 4E 6F 29  
 05D6 20 3F 20 24

```

;
fvisual:
      defb 20,'0',bell,pfx,rever,' FORMATTING ',pfx,norm,endmsg

```

05DA 14 30 07 13  
 05DE 42 20 46 4F  
 05E2 52 4D 41 54  
 05E6 54 49 4E 47

```

05EA 20 13 40 24
05EE                                vvisual:
05EE                                defb    bell,pfx,rever,' VERIFYING ',pfx,norm,endmsg
05F2 56 45 52 49
05F6 46 59 49 4E
05FA 47 20 20 13
05FE 40 24

;
0600                                vsidtrk:
0600                                defb    'SIDE    TRACK',endmsg
0604 20 20 20 20
0608 54 52 41 43
060C 4B 24

;
060E                                nomsg:
060E                                defb    19,48h,'NO',19,40h,endmsg
0612 13 40 24
0615                                yesmsg:
0615                                defb    19,48h,'YES',19,40h,endmsg
0619 53 13 40 24

;
;
;*****
;*          Input Verify Table          *
;*****
061D                                opt1:
061D                                defb    0,0,'N','A'
0621                                opt2:
0621                                defb    1,1,'N','A'
0625                                opt3:
0625                                defb    0,0,'N','B'
0629                                opt4:
0629                                defb    1,1,'N','B'
062D                                opt5:
062D                                defb    0,0,'Y','A'
0631                                opt6:
0631                                defb    1,1,'Y','A'
0635                                opt7:
0635                                defb    0,0,'Y','B'
0639                                opt8:
0639                                defb    1,1,'Y','B'

;
                                page
    
```

```

;
;*****
;*          Formatting Table          *
;*****
;
;
063D      fddtab:
063D      preamble:
                rept 40          ; track preamble (40 byte 'ff')
.XLIST      defb 255          ;
.LIST
                endm
;
0665      idfield:
                rept 6          ; GAP III (6 bytes '00')
                defb 0          ;
                endm
0665      00      +          defb 0          ;
0666      00      +          defb 0          ;
0667      00      +          defb 0          ;
0668      00      +          defb 0          ;
0669      00      +          defb 0          ;
066A      00      +          defb 0          ;
;
066B      FE          defb 0feh          ; ID Address Mark
066C      trknum:
066C      defs 1          ; track number (1 byte)
066D      sidnum:
066D      00          defb 0          ; side number (0 or 1)
066E      secnum:
066E      defs 1          ; sector number (1 byte)
;
066F      01          defb 1          ; sector length (256 byte)
;
0670      F7          defb 0f7h          ; 2 CRC's written
;
                rept 11          ; GAP II (11 bytes 'ff')
                defb 255          ;
                endm
0671      FF      +          defb 255          ;
0672      FF      +          defb 255          ;
0673      FF      +          defb 255          ;
0674      FF      +          defb 255          ;
0675      FF      +          defb 255          ;
0676      FF      +          defb 255          ;
0677      FF      +          defb 255          ;
0678      FF      +          defb 255          ;
0679      FF      +          defb 255          ;
067A      FF      +          defb 255          ;
067B      FF      +          defb 255          ;
;
                rept 6          ; GAP II (6 bytes '00')
                defb 0          ;
                endm
067C      00      +          defb 0          ;

```

```

067D  00      +      defb  0      ;
067E  00      +      defb  0      ;
067F  00      +      defb  0      ;
0680  00      +      defb  0      ;
0681  00      +      defb  0      ;

      ;
0682  FB      ;      defb  0fbh      ; Data Address Mark
      ;
      rept  256      ; User Data (256 bytes 'e5')
      .XLIST
      defb  0e5h      ;
      .LIST
      endm

      ;
0783  F7      ;      defb  0f7h      ; 2 CRC's written
      ;
      rept  14      ; GAP IV (14 bytes 'ff')
      defb  255      ;
      endm
0784  FF      +      defb  255      ;
0785  FF      +      defb  255      ;
0786  FF      +      defb  255      ;
0787  FF      +      defb  255      ;
0788  FF      +      defb  255      ;
0789  FF      +      defb  255      ;
078A  FF      +      defb  255      ;
078B  FF      +      defb  255      ;
078C  FF      +      defb  255      ;
078D  FF      +      defb  255      ;
078E  FF      +      defb  255      ;
078F  FF      +      defb  255      ;
0790  FF      +      defb  255      ;
0791  FF      +      defb  255      ;

      ;
      page
  
```

```

0792
0792 11 02FD      ld    de,inimsg    ; DE = initial message
0795 CD F01E      call   print        ; print it
0798 11 0347      ld    de,questi    ; display selection menu table
079B CD F01E      call   print        ;

;=====
;*****
;*
;*          FDDFORM entry point
;*
;*
;*****
;

fddform:
    ld    de,inimsg    ; DE = initial message
    call   print        ; print it
    ld    de,questi    ; display selection menu table
    call   print        ;

;=====
fddf00:
    call   cin          ; wait one char
    cp     3            ; is cntrl C
    push   af
    ld     c,a          ;
    call   cout         ; display chr
    pop    af
    jp     z,0          ; yes, then return to cp/m
    cp     '1'          ; is option 1 ?
    jr     z,sforA      ; yes goto format single A
    cp     '2'          ; is option 2 ?
    jr     z,dforA      ; yes goto format double A
    cp     '3'          ; is option 3 ?
    jr     z,sforB      ; yes goto format single B
    cp     '4'          ; is option 4 ?
    jr     z,dforB      ; yes goto format double B
    cp     '5'          ; is option 5 ?
    jr     z,sverA      ; yes goto format single verify A
    cp     '6'          ; is option 6 ?
    jr     z,dverA      ; yes goto format double verify A
    cp     '7'          ; is option 7 ?
    jr     z,sverB      ; yes goto format single verify B
    cp     '8'          ; is option 8 ?
    jr     nz,fddf00    ; no, then retry

dverB:
    ld     hl,opt8      ;load table double format verify B
    call   optload      ;load parameters
    jr     alls         ;now can format

sforA:
    ld     hl,opt1      ;load table single format A
    call   optload      ;load parameters
    jr     alls         ;now can format

dforA:
    ld     hl,opt2      ;load table double format A
    call   optload      ;load parameters
    jr     alls         ;now can format

sforB:
    ld     hl,opt3      ;load table double format B
    call   optload      ;load parameters
    jr     alls         ;now can format

dforB:
    ld     hl,opt4      ;load table double format B
    call   optload      ;load parameters

```

```

07F2 1B 16          jr      alls          ;now can format
07F4                      sverA:
07F4 21 062D        ld      hl,opt5      ;load table single format verify A
07F7 CD 0A09        call    optload      ;load parameters
07FA 1B 0E          jr      alls          ;now can format
07FC                      dverA:
07FC 21 0631        ld      hl,opt6      ;load table double format verify A
07FF CD 0A09        call    optload      ;load parameters
0802 1B 06          jr      alls          ;now can format
0804                      sverB:
0804 21 0635        ld      hl,opt7      ;load table single format verify B
0807 CD 0A09        call    optload      ;load parameters
;
080A 11 05A1        alls: ld      de,quest3  ; are you sure
080D CD F01E        call    print         ;
0810 CD F003        call    cin          ; wait one char
0813 CB AF          res     5,a          ; convert up-case
0815 FE 59          cp      'Y'         ; is yes ?
0817 2B 09          jr      z,form3      ; no, then retry
0819 11 060E        ld      de,nomsg     ; no message
081C CD F01E        call    print         ; print it
081F C3 0983        jp      endanyop     ; and retry
0822                      form3:
0822 11 0615        ld      de,yesmsg     ; yes message
0825 CD F01E        call    print         ; print it
; at this point all formatting parameter are entry
;
0828 21 0F20        ld      hl,msg1add    ; message 1 cursor address
082B 11 05DA        ld      de,fvisual    ;
082E CD F024        call    printat       ; print 'FORMATTING'
0831 21 111D        ld      hl,msg2add    ; SIDE TRACK cursor addr.
0834 11 0600        ld      de,vsidtrk    ; SIDE TRACK msg addrs.
0837 CD F024        call    printat       ; print it
083A 3A 05BD        ld      a,(disknum)    ; load disk number
083D D6 40          sub     '@'          ; A = 1 for disk A or 2 for disk B
083F 32 015A        ld      (dsksid),a    ; initialli, side is 0
0842 AF            xor      a            ; clear accumulator
0843 32 066D        ld      (sidnum),a    ; set side 0 in ID field
0846                      fdfok:
0846 AF            xor      a            ; clear accumulator
0847 32 066C        ld      (trknum),a    ; set track 0
084A 3C            inc     a            ; A = 1
084B 32 066E        ld      (secnum),a    ; set sector 1
084E 3A 015A        ld      a,(dsksid)    ; load disk and side para
0851 D3 D6          out     (fddlch),a    ; select drive and side
0853 3E 02          ld      a,fddrest     ; load fdd restore command code
0855 D3 D0          out     (fddcmd),a    ; send out to 1771
0857                      fdfor1:
0857 CD 09C8        call    waitfd        ; wait until end command
085A B7            or      a            ; zero in accumulator ?
085B C2 09F4        jp      nz,timeout    ; if no zero then disk offline
;
085E CD 0992        call    sidtrkvis     ; visualize side and track
0861 21 063D        ld      hl,preamble   ; M.L = preamble table (40 bytes 'ff')
0864 0E D7          ld      c,fdddat      ; C = fdd data register
0866 06 28          ld      b,40         ; bytes counter
0868 3E F4          ld      a,fddwtrk     ; load write track command code
086A D3 D0          out     (fddcmd),a    ; send out to 1771

```

```

086C  CD 09C3          call  fddelay          ; wait aproax 56 micros
086F                                wtpreamb:
086F  DB D0              in    a,(fddsts)      ; load fdd status
0871  CB 4F              bit    1,a            ; test DRQ bit
0873  2B FA              jr     z,wtpreamb     ; wait if no DRQ
0875  ED A3              outi                   ; write one byte
0877  20 F6              jr     nz,wtpreamb    ; repeat until all 40 bytes are output
0879                                fdfor2:
0879  21 0665            ld     hl,idfield     ; H.L = ID field data table
;
; total byte to output are 301.
;
087C  06 00              ld     b,0            ; 256 bytes to 1771
087E                                wtidfield:
087E  DB D0              in    a,(fddsts)      ; load fdd status
0880  CB 4F              bit    1,a            ; test DRQ bit
0882  2B FA              jr     z,wtidfield    ; wait if no DRQ
0884  ED A3              outi                   ; write one byte
0886  20 F6              jr     nz,wtidfield    ; repeat until all 256 bytes are output
;
0888  06 2D              ld     b,301-256      ; load rimanents bytes to 1771
088A                                wtliidfield:
088A  DB D0              in    a,(fddsts)      ; load fdd status
088C  CB 4F              bit    1,a            ; test DRQ bit
088E  2B FA              jr     z,wtliidfield  ; wait if no DRQ
0890  ED A3              outi                   ; write one byte
0892  20 F6              jr     nz,wtliidfield ; repeat until all 45 bytes are output
;
; now, all 301 bytes are output to 1771
;
0894  3A 066E            ld     a,(secnum)      ; load actual sector number
0897  3C                  inc     a            ; inc. sector number
0898  32 066E            ld     (secnum),a      ; set next sector number
0898  FE 0B              cp     11            ; last sector has been written ?
089D  C2 0879            jp     nz,fdfor2      ; no, then write next sector
08A0                                endtrk:
08A0  DB D0              in    a,(fddsts)      ; load fdd status
08A2  CB 47              bit    0,a            ; end track ?
08A4  2B 07              jr     z,nexttrk     ; yes, then go to next track
08A6  3E FF              ld     a,255          ; load byte 'ff'
08AB  D3 D7              out    (fdddat),a     ; write to 1771 data register
08AA  C3 08A0            jp     endtrk         ; repeat until end track
08AD                                nexttrk:
08AD  DB D0              in    a,(fddsts)      ; load fdd status
08AF  E6 E7              and    11100111b     ; write track error?
08B1  C2 09E8            jp     nz,fdforerr    ; yes, then goto error
08B4  3A 066C            ld     a,(trknum)     ; load track number
08B7  3C                  inc     a            ; point to next track
08BB  FE 28              cp     40            ; end side ?
08BA  2B 0F              jr     z,endside     ; yes, then go to end
08BC  32 066C            ld     (trknum),a     ; set next track
08BF  3E 01              ld     a,1            ; load first sector number
08C1  32 066E            ld     (secnum),a     ; set first sector number
;
08C4  3E 52              ld     a,fddsin      ; load fdd step in
08C6  D3 D0              out    (fddcmd),a     ; send out to 1771
08CB  C3 0857            jp     fdfor1        ; count for next track
;

```

```

08CB      endside:
08CB      3A 015B      ld      a,(fsideflag) ; single or double side ?
08CE      B7          or      a ;
08CF      CA 08E7      jp      z,endsdk ; a = 0 then s.s. or end side two then jmp to format another disk
08D2      AF          xor      a ; clear accumulator
08D3      32 015B      ld      (fsideflag),a ; reset side flag
08D6      3C          inc      a ; A = 1
08D7      32 066D      ld      (sidnum),a ; set side one in ID field
08DA      3A 05BD      ld      a,(disknum) ; load diak number
08DD      D6 40        sub      '0' ; 1 for A, 2 for B
08DF      F6 20        or      00100000b ; set side one
08E1      32 015A      ld      (dsksid),a ;
08E4      C3 0846      jp      fdfok ; format side 1

;
08E7      endsdk:
08E7      AF          xor      a ; clear accumulator
08E8      D3 D6        out      (fddlch),a ; deselect any disk

;
; at this point, disk has been formatted
;
08EA      3A 015D      ld      a,(verflag) ; load verify flag
08ED      FE 59        cp      'Y' ; is yes ?
08EF      C2 0983      jp      nz,endanyop ; no, then goto end disk

;
08F2      dskverify:
08F2      21 0F20      ld      hl,msgladd ; message 1 cursor address
08F5      11 05EE      ld      de,vvisual ; DE -> verify message
08F8      CD F024      call     printat ; print it at HL
08FB      3A 015A      ld      a,(dsksid) ; load disk and side
08FE      E6 03        and      00000011b ; mask disk number
0900      3D          dec      a ; disk A = 0, B = 1

;
0901      dskve10:
0901      32 0268      ld      (verdisk),a ; start disk and side
0904      AF          xor      a ; clear accumulator
0905      32 0269      ld      (vertrk),a ; start track 0
0908      dskve00:
0908      3C          inc      a ;
0909      32 0268      ld      (versec),a ; start sector 1
090C      11 015E      ld      de,xlt ; DE -> translate table

;
090F      vervis:
090F      3A 0268      ld      a,(verdisk) ; load verify disk and side
0912      E6 10        and      00010000b ; mask side
0914      28 02        jr      z,verv00 ; jmp if side 0
0916      3E 01        ld      a,1 ; set side 1
0918      verv00:
0918      D5          push     de ; save xlt pointer
0919      32 066D      ld      (sidnum),a ; set ID side for visualization
091C      3A 0269      ld      a,(vertrk) ; load verify track
091F      32 066C      ld      (trknum),a ; set ID side for visualization
0922      CD 0992      call     sidtrkvis ; visualize side and track

;
0925      21 0268      ld      hl,vertab ; verify tab para
0928      CD F015      call     fdiod ; read 256 bytes
092B      B7          or      a ; read error ?
092C      28 16        jr      z,readok ; no, then goto verify bytes
092E      readerr:
    
```



```

092E 11 02C9          ld     de,rdermsg      ; DE -> read error message
0931                      rder00:
0931          21 1719          ld     hl,errmsgadd    ; error message address
0934          CD F024          call    printat        ; print it
0937          11 02ED          ld     de,wterr      ; print 'Hit any key'
093A          CD F01E          call    print
093D          CD F003          call    cin
0940          D1              pop     de            ; restore xlt pointer
0941          C3 0792          jp      fddform       ; and retry
0944                      readok:
0944          21 0168          ld     hl,verbuff     ; HL -> verify buffer
0947          01 0100          ld     bc,256        ; num. byte to verify
094A                      verif00:
094A          7E              ld     a,(hl)         ; load one byte
094B          FE E5          cp      0e5h          ; IBM standard data ?
094D          2B 05          jr      z,verok        ; yes, then count for next data
094F                      vererr:
094F          11 02DB          ld     de,verrmmsg    ; DE -> verify error message
0952          18 DD          jr      rder00        ; print it and retry
0954                      verok:
0954          23              inc     hl            ; point to next data
0955          0B              dec     bc            ; dec. data counter
0956          7B              ld     a,b
0957          B1              or      c
0958          20 F0          jr      nz,verif00     ; all 256 bytes are verify ?
095A                      endverify:
095A          D1              pop     de            ; DE -> xlt pointer
095B          1A              ld     a,(de)         ; load verify sector number
095C          13              inc     de            ; point to next sector
095D          32 026B          ld     (versec),a    ; set next sector
0960          FE 00          cp      0             ; end track ?
0962          20 AB          jr      nz,vervis      ; no, then count
0964          3A 0269          ld     a,(vertrk)    ; load verify track
0967          3C              inc     a
0968          32 0269          ld     (vertrk),a    ; set next track
096B          FE 2B          cp      40            ; end side ?
096D          3E 00          ld     a,0            ; A = 0 for sector set
096F          20 97          jr      nz,dskve00    ; no end side, count to next track
0971                      vendside:
0971          3A 015C          ld     a,(vsideflag)  ; load verify side flag
0974          B7              or      a
0975          2B 0C          jr      z,endanyop     ; single side or end disk ?
0977          AF              xor     a
0978          32 015C          ld     (vsideflag),a ; clear verify side flag
097B          3A 0268          ld     a,(verdisk)   ; load verify disk
097E          F6 10          or      00010000b    ; set side 1
0980          C3 0901          jp      dskve10     ; and go to verify side one
0983                      ;
0983                      endanyop:
0983                      ; entry point to go to format another disk
0983          06 02          ld     b,2            ; set soft timer 2
0985                      endan00:
0985          11 0000          ld     de,0          ; set soft timer 1
0988                      endan11:
0988          1B              dec     de            ; timer 1 down
0989          7A              ld     a,d
098A          B3              or      e
098B          20 FB          jr      nz,endan11    ; zero for timer 1 ?
098B          20 FB          jr      nz,endan11    ; no, then count

```

```

098D 10 F6          djnz  endan00      ; timer 2 down and repeat until timer 2 = 0
098F C3 0792        jp    fddform     ; go to format another disk
;
;
0992          sidtrkvis:
; visualize side and track
0992 21 1122        ld    hl,snumadd   ; side video addr.
0995 CD F027        call  movcurs     ; move cursor
0998 3A 066D        ld    a,(sidnum)   ; load ID side number
099B C6 30          add    a,'0'       ; convert ASCII
099D 4F            ld    c,a          ;
099E CD F006        call  cout        ; print side 0 or 1
09A1 21 112B        ld    hl,tnumadd   ; track video addr.
09A4 CD F027        call  movcurs     ; move cursor
09A7 3A 066C        ld    a,(trknum)   ; load ID track number
09AA 06 FF          ld    b,255       ; set decimal counter
09AC          sidtr00:
09AC 04            inc    b            ; inc. decimal digit
09AD D6 0A          sub    10          ; A=A-10
09AF 30 FB          jr    nc,sidtr00   ; count if A >= 0
09B1 C6 0A          add    a,10        ; at this point A = lsd, B = msd (BCD)
09B3 F5            push   af          ; save lsd
09B4 78            ld    a,b          ; load msd
09B5 C6 30          add    a,'0'       ; convert ascii
09B7 4F            ld    c,a          ;
09BB CD F006        call  cout        ; print msd
09BB F1            pop    af          ;
09BC C6 30          add    a,'0'       ; convert ascii
09BE 4F            ld    c,a          ;
09BF CD F006        call  cout        ; print lsd
09C2 C9            ret              ; and ret
;
;
09C3          fddelay:
;
;
09C3          rept 4
09C3 E3            ex     (sp),hl      ; delay beetwen write command reg.
09C4 E3            ex     (sp),hl      ; to read status reg.
09C5 E3            ex     (sp),hl      ; delay beetwen write command reg.
09C6 E3            ex     (sp),hl      ; delay beetwen write command reg.
09C7 C9            ret
;
;
09C8          waitfd:
; wait until fdd busy is reset.
09C8 CD 09C3        call  fddelay     ; wait aproax 56 microS
09CB 06 02          ld    b,2         ; set soft timer
09CD          wait00:
09CD 11 0000        ld    de,0        ; for aproax five seconds
09D0          wait01:
09D0 DB D0          in     a,(fddsts)  ; input to fdd status
09D2 CB 47          bit    0,a         ; test busy bit
09D4 2B 0F          jr    z,wait02    ; jump if no command is in progress
09D6 1B            dec    de          ;
09D7 7A            ld    a,d          ; timer down
09D8 B3            or     e           ;
09D9 20 F5          jr    nz,wait01   ;

```

```

09DB 05          dec    b          ;
09DC 20 EF      jr     nz,wait00   ; time out
09DE          offline:
09DE 3E D0      ld     a,fddrst     ; reset fdd controller
09E0 D3 D0      out    (fddcmd),a  ; exec. command
09E2 3E 01      ld     a,00000001b ; set time-out bit error
09E4 C9         ret                ; and ret
09E5          wait02:
09E5 47         ld     b,a          ; save fdd status in B register
09E6 AF         xor     a           ; clear accumulator for
09E7 C9         ret                ; normal return

;
09E8          fdforerr:
09E8 21 1719    ld     hl,errmsgadd ; error message address
09EB 11 02BA    ld     de,fmtermg   ; DE = format error message
09EE CD F024    call   printat      ; print it
09F1 C3 09FD    jp     retry        ; and go to cp/m

;
09F4          timeout:
09F4 21 1719    ld     hl,errmsgadd ; error message address
09F7 11 02AB    ld     de,oflmsg    ; DE = offline message
09FA CD F024    call   printat      ; print it
09FD          retry:
09FD 11 026F    ld     de,setnew     ; DE = set new disk message
0A00 CD F01E    call   print        ; print it
0A03 CD F003    call   cin          ; wait one char.
0A06 C3 0000    jp     0           ; and return to cp/m
0A09          optload:
0A09 7E         ld     a,(hl)        ;load first byte
0A0A 32 015B    ld     (fsideflag),a ;load with that format side flag
0A0D 23         inc    hl           ;another one
0A0E 7E         ld     a,(hl)        ;load second byte
0A0F 32 015C    ld     (vsideflag),a ;load with that verify side flag
0A12 23         inc    hl           ;another one
0A13 7E         ld     a,(hl)        ;load thirth byte
0A14 32 015D    ld     (verflag),a   ;load with that verify flag
0A17 23         inc    hl           ;anoter one
0A18 7E         ld     a,(hl)        ;load fourth byte
0A19 32 05BD    ld     (disknum),a   ;load with that disk number buffer
0A1C C9         ret                ;no more bytes

;
;
;
;
end 100h          ; end of floppy disk formatter
    
```

Macros:

Symbols:

ALLS	080A	BACKSP	0008	BELL	0007	BOOTRD	F021
CIN	F003	COMPFL	F02D	COPYRI	0001	COU	F006
CR	000D	CSTS	F009	DFOR	07DC	DFORB	07EC
DISKNU	05BD	DSKSID	015A	DSKVE0	0908	DSKVE1	0901
DSKVER	08F2	DVERA	07FC	DVERB	07CC	ENDANO	0985
ENDAN1	0988	ENDANY	0983	ENDDSK	08E7	ENDMSG	0024
ENDSID	08CB	ENDTRK	08A0	ENDVER	095A	ERMSGA	1719
FALSE	0000	FDBUSY	0001	FDDCMD	00D0	FDDDAT	00D7
FDDELA	09C3	FDDF00	079E	FDDFOR	0792	FDDLCH	00D6
FDDRES	0002	FDDRQ	0002	FDDRST	00D0	FDDSEC	00D2
FDDSIN	0052	FDDSTS	00D0	FDDTAB	063D	FDDTRK	00D1
FDDWTR	00F4	FDFOK	0846	FDFOR1	0857	FDFOR2	0879
FDFORE	09E8	FDIOD	F015	FDIOS	F012	FDNRDY	0080
FDT1ER	0018	FDT23E	001F	FDWPR	0040	FFED	000C
FLASH	0043	FMTERM	02BA	FORM3	0822	FSIDEF	015B
FVISUA	05DA	IDFIEL	0665	INIMSG	02FD	LF	000A
LOUT	F00C	LSTS	F00F	MOVCUR	F027	MSG1AD	0F20
MSG2AD	111D	NEXTTR	08AD	NOMSG	060E	NORM	0040
OFFLINE	09DE	OFLMSG	02AB	OPT1	061D	OPT2	0621
OPT3	0625	OPT4	0629	OPT5	062D	OPT6	0631
OPT7	0635	OPT8	0639	OPTLOA	0A09	PFX	0013
PREAMB	063D	PRINT	F01E	PRINTA	F024	QUEST1	0347
QUEST3	05A1	RDER00	0931	RDERMS	02C9	READER	092E
READOK	0944	RETRY	09FD	REVER	0042	ROM	F000
SECNUM	066E	SETNEW	026F	SFOR	07D4	SFORB	07E4
SIDNUM	066D	SIDTRO	09AC	SIDTRK	0992	SNUMAD	1122
SPACE	0020	STACK	015A	STROUT	F01E	SVERA	07F4
SVERB	0804	TAB	0009	TIMOU	09F4	TNUMAD	112B
TRKNUM	066C	TRUE	0001	VENDSI	0971	VERBUF	0168
VERDMA	026C	VERDSK	0268	VERERR	094F	VERFLA	015D
VERIFO	094A	VEROK	0954	VEROP	026E	VERRMS	02DB
VERSEC	026B	VERTAB	0268	VERTRK	0269	VERV00	0918
VERVIS	090F	VIDINI	F02A	VSIDEF	015C	VSIDTR	0600
VVISUA	05EE	WAIT00	09CD	WAIT01	09D0	WAIT02	09E5
WAITFD	09CB	WDINI	F01B	WDIO	F01B	WT1IDF	088A
WTERR	02ED	WTIDFI	087E	WTPREA	086F	XLT	015E
YESMSG	0615						

No Fatal error(s)