```
                        ;
                        ;
                        ;
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*     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 1984 by Studio Lg, Genova-Last rev.19/08/84 14:17
                        ; Programmer: Martino Stefano
                        ; Modified by Gallerani Paolo
                        ; Last Revision 19/08/1984 14:17
                        ;
0000'                           aseg
                                .z80            ; this program is written in z80 mnemonics
                        ;
0001                    true    equ     1
0000                    false   equ     0
                        ;
0000                    copyrig equ     false
                                ;
                                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               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
0048            C       rever   equ     'H'             ; Reverse On    (^SH)
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
```

```
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*            FD 1771 I/O port                           *
                        ;*                                                       *
                        ;*********************************************************
00D0                    fddsts  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     fddsts          ; fdd command port
                        ;
                        ;*********************************************************
                        ;*                                                       *
                        ;*            FD 1771 Command Summary                    *
                        ;*                                                       *
                        ;*********************************************************
                        ;
                               ; this command are without verify because disk don't are 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                    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                  *
                        ;*******************************************************
                        ;
0E19                    ermsgadd equ    0e19h           ; error msg position (X=25 Y=14)
0A24                    msg1add equ     0a24h           ; form/ver msg position (X=36 Y=10)
0C21                    msg2add equ     0c21h           ; sid/trk msg position (X=33 Y=12)
0C26                    snumadd equ     0c26h           ; sid num. position (X=38 Y=12)
0C30                    tnumadd equ     0c30h           ; trk num. position (X=48 Y=12)
                        ;
                        ;
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*                F D D F O R M                        *
                        ;*                                                     *
                        ;*******************************************************
                        ;
0000                            Aseg
                                Org     100h
                        ;
0100    31 0126                 ld      sp,stack        ; set stack pointer
0103    C3 054B                 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:
0126                            defs    1               ; byte to output lach
0127                    fsideflag:
0127                            defs    1               ; format side flagh
0128                    vsideflag:
0128                            defs    1               ; verify side flag
                        ;
0129                    xlt:
0129    05 09 03 07             defb    5,9,3,7,2,6,10,4,8,0
012D    02 06 0A 04
0131    08 00
                        ;
0133                    verflag:
0133                            defs    1               ; verify flag (Y or N)
                        ;
0134                    verbuff:
0134                            defs    256             ; verify buffer (256 byte)
```

```
                        ;
                        ;
0234                    vertab:
0234                    verdsk: defs    0               ; verify disk and side
0234    0000            vertrk: defw    0               ; verify track number
0236    01              versec: defb    1               ; verify sector number
0237    0134            verdma: defw    verbuff         ; verify dma address
0239    00              verop:  defb    0               ; verify = read operation
                        ;
                                page
```

```
                        ;
                        ;*******************************************************
                        ;*           E r r o r   M e s s a g e s               *
                        ;*******************************************************
                        ;
023A                    setnew:
023A  0D 0A 53 65               defb    cr,lf,'Set new system diskette in drive A,'
023E  74 20 6E 65
0242  77 20 73 79
0246  73 74 65 6D
024A  20 64 69 73
024E  6B 65 74 74
0252  65 20 69 6E
0256  20 64 72 69
025A  76 65 20 41
025E  2C
025F  0D 0A 74 68               defb    cr,lf,'then push any key.',cr,lf,endmsg
0263  65 6E 20 70
0267  75 73 68 20
026B  61 6E 79 20
026F  6B 65 79 2E
0273  0D 0A 24
                        ;
0276                    oflmsg:
0276  0D 0A 44 69               defb    cr,lf,'Disk offline',endmsg
027A  73 6B 20 6F
027E  66 66 6C 69
0282  6E 65 24
                        ;
0285                    fmtermsg:
0285  0D 0A 46 6F               defb    cr,lf,'Format error',endmsg
0289  72 6D 61 74
028D  20 65 72 72
0291  6F 72 24
                        ;
0294                    rdermsg:
0294  56 65 72 69       ->      defb    'Verify read error',endmsg
0298  66 79 20 72
029C  65 61 64 20
02A0  65 72 72 6F
02A4  72 24
02A6                    verrmsg:
02A6  56 65 72 69               defb    'Verify data error',endmsg
02AA  66 79 20 64
02AE  61 74 61 20
02B2  65 72 72 6F
02B6  72 24
02B8                    wterr:
02B8  20 2D 20 48               defb    ' - Hit any key ',endmsg
02BC  69 74 20 61
02C0  6E 79 20 6B
02C4  65 79 20 24
                                ;
                                page
```

```
                                ;
02C8                    inimsg:
02C8  0C 0D 0A 13               defb    ffeed,cr,lf,19,48h,'   Floppy Disk formatter vers 4.0',19,40h,cr,lf,cr,lf,endmsg
02CC  48 20 20 20
02D0  46 6C 6F 70
02D4  70 79 20 44
02D8  69 73 6B 20
02DC  66 6F 72 6D
02E0  61 74 74 65
02E4  72 20 76 65
02E8  72 73 20 34
02EC  2E 30 13 40
02F0  0D 0A 0D 0A
02F4  24
                        ;
02F5                    quest1:
02F5  53 69 6E 67               defb    'Single or Double Side (S/D or ^C to exit) ? ',endmsg
02F9  6C 65 20 6F
02FD  72 20 44 6F
0301  75 62 6C 65
0305  20 53 69 64
0309  65 20 28 53
030D  2F 44 20 6F
0311  72 20 5E 43
0315  20 74 6F 20
0319  65 78 69 74
031D  29 20 3F 20
0321  24
                        ;
0322                    quest2:
0322  0D 0A 54 79               defb    cr,lf,'Type disk to be format (A/B) ',endmsg
0326  70 65 20 64
032A  69 73 6B 20
032E  74 6F 20 62
0332  65 20 66 6F
0336  72 6D 61 74
033A  20 28 41 2F
033E  42 29 20 24
                        ;
0342                    quest3:
0342  0D 0A 07 41               defb    cr,lf,bell,'Are you sure for disk ',19,48h
0346  72 65 20 79
034A  6F 75 20 73
034E  75 72 65 20
0352  66 6F 72 20
0356  64 69 73 6B
035A  20 13 48
035D                    disknum:
035D  00                        defb    0
035E  13 40 20 28               defb    19,40h,' (Y for Yes, any key for no) ? ',endmsg
0362  59 20 66 6F
0366  72 20 59 65
036A  73 2C 20 61
036E  6E 79 20 6B
0372  65 79 20 66
0376  6F 72 20 6E
```

```
037A  6F 29 20 3F
037E  20 24

                        ;
0380                    quest4:
0380  0D 0A 44 6F               defb    cr,lf,'Do you want disk verify (Y/N) ? ',endmsg
0384  20 79 6F 75
0388  20 77 61 6E
038C  74 20 64 69
0390  73 6B 20 76
0394  65 72 69 66
0398  79 20 28 59
039C  2F 4E 29 20
03A0  3F 20 24

                        ;
03A3                    fvisual:
03A3  07 13 48 46               defb    bell,19,48h,'FORMATTING',19,40h,endmsg
03A7  4F 52 4D 41
03AB  54 54 49 4E
03AF  47 13 40 24
03B3                    vvisual:
03B3  07 13 48 56               defb    bell,19,48h,'VERIFYING ',19,48h,endmsg
03B7  45 52 49 46
03BB  59 49 4E 47
03BF  20 13 48 24

                        ;
03C3                    vsidtrk:
03C3  53 49 44 45               defb    'SIDE    TRACK',endmsg
03C7  20 20 20 20
03CB  54 52 41 43
03CF  4B 24

                        ;
03D1                    nomsg:
03D1  13 48 4E 4F               defb    19,48h,'NO',19,40h,endmsg
03D5  13 40 24
03D8                    yesmsg:
03D8  13 48 59 45               defb    19,48h,'YES',19,40h,endmsg
03DC  53 13 40 24

                        ;
03E0                    sngmsg:
03E0  13 48 53 49               defb    19,48h,'SINGLE',19,40h,endmsg
03E4  4E 47 4C 45
03E8  13 40 24
03EB                    dblmsg:
03EB  13 48 44 4F               defb    19,48h,'DOUBLE',19,40h,endmsg
03EF  55 42 4C 45
03F3  13 40 24

                        ;
                                page
```

```
                        ;
                        ;*********************************************************
                        ;*              Formatting Table                         *
                        ;*********************************************************
                        ;
                        ;
03F6                    fdctab:
03F6                    preamble:
                                rept    40              ; track preamble (40 byte 'ff')
                                defb    255             ;
                                endm
03F6    FF              +       defb    255             ;
03F7    FF              +       defb    255             ;
03F8    FF              +       defb    255             ;
03F9    FF              +       defb    255             ;
03FA    FF              +       defb    255             ;
03FB    FF              +       defb    255             ;
03FC    FF              +       defb    255             ;
03FD    FF              +       defb    255             ;
03FE    FF              +       defb    255             ;
03FF    FF              +       defb    255             ;
0400    FF              +       defb    255             ;
0401    FF              +       defb    255             ;
0402    FF              +       defb    255             ;
0403    FF              +       defb    255             ;
0404    FF              +       defb    255             ;
0405    FF              +       defb    255             ;
0406    FF              +       defb    255             ;
0407    FF              +       defb    255             ;
0408    FF              +       defb    255             ;
0409    FF              +       defb    255             ;
040A    FF              +       defb    255             ;
040B    FF              +       defb    255             ;
040C    FF              +       defb    255             ;
040D    FF              +       defb    255             ;
040E    FF              +       defb    255             ;
040F    FF              +       defb    255             ;
0410    FF              +       defb    255             ;
0411    FF              +       defb    255             ;
0412    FF              +       defb    255             ;
0413    FF              +       defb    255             ;
0414    FF              +       defb    255             ;
0415    FF              +       defb    255             ;
0416    FF              +       defb    255             ;
0417    FF              +       defb    255             ;
0418    FF              +       defb    255             ;
0419    FF              +       defb    255             ;
041A    FF              +       defb    255             ;
041B    FF              +       defb    255             ;
041C    FF              +       defb    255             ;
041D    FF              +       defb    255             ;
                        ;
041E                    idfield:
                                rept    6               ; GAP III (6 bytes '00')
                                defb    0               ;
                                endm
```

```
041E  00        +           defb  0          ;
041F  00        +           defb  0          ;
0420  00        +           defb  0          ;
0421  00        +           defb  0          ;
0422  00        +           defb  0          ;
0423  00        +           defb  0          ;
                        ;
0424  FE                    defb  0feh       ; ID Address Mark
0425                    trknum:
0425                        defs  1          ; track number (1 byte)
0426                    sidnum:
0426  00                    defb  0          ; side number (0 or 1)
0427                    secnum:
0427                        defs  1          ; sector number (1 byte)
                        ;
0428  01                    defb  1          ; sector lenght (256 byte)
                        ;
0429  F7                    defb  0f7h       ; 2 CRC's written
                        ;
                            rept  11         ; GAP II (11 bytes 'ff')
                            defb  255        ;
                            endm
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        ;
                        ;
                            rept  6          ; GAP II (6 bytes '00)
                            defb  0          ;
                            endm
0435  00        +           defb  0          ;
0436  00        +           defb  0          ;
0437  00        +           defb  0          ;
0438  00        +           defb  0          ;
0439  00        +           defb  0          ;
043A  00        +           defb  0          ;
                        ;
043B  FB                    defb  0fbh       ; Data Address Mark
                        ;
                            rept  256        ; User Data (256 bytes 'e5')
                            defb  0e5h       ;
                            endm
043C  E5        +           defb  0e5h       ;
043D  E5        +           defb  0e5h       ;
043E  E5        +           defb  0e5h       ;
043F  E5        +           defb  0e5h       ;
0440  E5        +           defb  0e5h       ;
0441  E5        +           defb  0e5h       ;
0442  E5        +           defb  0e5h       ;
0443  E5        +           defb  0e5h       ;
0444  E5        +           defb  0e5h       ;
```

```
0445  E5          +          defb   0e5h          ;
0446  E5          +          defb   0e5h          ;
0447  E5          +          defb   0e5h          ;
0448  E5          +          defb   0e5h          ;
0449  E5          +          defb   0e5h          ;
044A  E5          +          defb   0e5h          ;
044B  E5          +          defb   0e5h          ;
044C  E5          +          defb   0e5h          ;
044D  E5          +          defb   0e5h          ;
044E  E5          +          defb   0e5h          ;
044F  E5          +          defb   0e5h          ;
0450  E5          +          defb   0e5h          ;
0451  E5          +          defb   0e5h          ;
0452  E5          +          defb   0e5h          ;
0453  E5          +          defb   0e5h          ;
0454  E5          +          defb   0e5h          ;
0455  E5          +          defb   0e5h          ;
0456  E5          +          defb   0e5h          ;
0457  E5          +          defb   0e5h          ;
0458  E5          +          defb   0e5h          ;
0459  E5          +          defb   0e5h          ;
045A  E5          +          defb   0e5h          ;
045B  E5          +          defb   0e5h          ;
045C  E5          +          defb   0e5h          ;
045D  E5          +          defb   0e5h          ;
045E  E5          +          defb   0e5h          ;
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  F7                defb    0f7h        ; 2 CRC's written
                ;
                          rept    14          ; GAP IV (14 bytes 'ff')
                          defb    255         ;
                          endm
053D  FF      +         defb    255         ;
053E  FF      +         defb    255         ;
053F  FF      +         defb    255         ;
0540  FF      +         defb    255         ;
0541  FF      +         defb    255         ;
0542  FF      +         defb    255         ;
0543  FF      +         defb    255         ;
0544  FF      +         defb    255         ;
0545  FF      +         defb    255         ;
0546  FF      +         defb    255         ;
0547  FF      +         defb    255         ;
0548  FF      +         defb    255         ;
0549  FF      +         defb    255         ;
054A  FF      +         defb    255         ;
                ;
                          page
```

```
                        ;
                        ;*******************************************************
                        ;*                                                     *
                        ;*              FDDFORM entry point                    *
                        ;*                                                     *
                        ;*******************************************************
                        ;
054B                    fddform:
054B  11 02C8                   ld      de,inimsg       ; DE = initial message
054E  CD F01E                   call    print           ; print it
0551  11 02F5                   ld      de,quest1       ; single or double side ?
0554  CD F01E                   call    print           ;
0557                    fddf00:
0557  CD F003                   call    cin             ; wait one char
055A  FE 03                     cp      3               ; is cntrl C
055C  CA 0000                   jp      z,0             ; yes, then return to cp/m
055F  CB AF                     res     5,a             ; convert up-case
0561  FE 44                     cp      'D'             ; is double ?
0563  28 0A                     jr      z,fordbl        ; yes then goto format double
0565  FE 53                     cp      'S'             ; is single ?
0567  20 EE                     jr      nz,fddf00       ; no, then retry
0569                    forsng:
0569  11 03E0                   ld      de,sngmsg       ; single message
056C  AF                        xor     a               ; clear accumulator for single side
056D  18 05                     jr      formsd          ; goto format common routine
056F                    fordbl:
056F  11 03EB                   ld      de,dblmsg       ; double message
0572  3E 01                     ld      a,1             ; load 1 for double side
0574                    formsd:
0574  32 0127                   ld      (fsideflag),a   ; set side flag for single or double side
0577  32 0128                   ld      (vsideflag),a   ; set side flag for 'S' or 'D' in verify
057A  CD F01E                   call    print           ; print single or double side
                        ;
057D  11 0380                   ld      de,quest4       ; do you want verify ?
0580  CD F01E                   call    print           ;
0583                    form7:
0583  CD F003                   call    cin             ; wait one char.
0586  CB AF                     res     5,a             ; conver up-case
0588  FE 59                     cp      'Y'             ; is yes ?
058A  11 03D8                   ld      de,yesmsg       ; yes message
058D  28 07                     jr      z,form8         ; yes, then count
058F  FE 4E                     cp      'N'             ; is no ?
0591  20 F0                     jr      nz,form7        ; no, then retry
0593  11 03D1                   ld      de,nomsg        ; no message
0596                    form8:
0596  32 0133                   ld      (verflag),a     ; set verify flag
0599  CD F01E                   call    print           ; print yes or no
                        ;
059C  11 0322                   ld      de,quest2       ; disk number
059F  CD F01E                   call    print           ;
05A2                    form2:
05A2  CD F003                   call    cin             ; wait one char
05A5  CB AF                     res     5,a             ; convert up-case
05A7  4F                        ld      c,a             ; save on c for print
05A8  FE 41                     cp      'A'             ; disk is less then 'A' ?
05AA  38 F6                     jr      c,form2         ; yes, then retry
```

```
05AC  FE 43              cp    'C'           ; disk is great then 'B' ?
05AE  30 F2              jr    nc,form2      ; yes, then retry
05B0  32 035D            ld    (disknum),a   ; disk is A or B then set disk number
05B3  CD F006            call  cout          ; print 'A' or 'B' for dissk number
                         ;
05B6  11 0342            ld    de,quest3     ; are you sure
05B9  CD F01E            call  print         ;
05BC  CD F003            call  cin           ; wait one char
05BF  CB AF              res   5,a           ; convert up-case
05C1  FE 59              cp    'Y'           ; is yes ?
05C3  28 09              jr    z,form3       ; no, then retry
05C5  11 03D1            ld    de,nomsg      ; no message
05C8  CD F01E            call  print         ; print it
05CB  C3 072F            jp    endanyop      ; and retry
05CE                     form3:
05CE  11 03D8            ld    de,yesmsg     ; yes message
05D1  CD F01E            call  print         ; print it
                         ;
                         ; at this point all formatting parameter are entry
                         ;
05D4  21 0A24            ld    hl,msg1add    ; message 1 cursor address
05D7  11 03A3            ld    de,fvisual    ;
05DA  CD F024  JP FS62   call  printat       ; print 'FORMATTING'
05DD  21 0C21            ld    hl,msg2add    ; SIDE  TRACK cursor addr.
05E0  11 03C3            ld    de,vsidtrk    ; SIDE  TRACK msg addrs.
05E3  CD F024            call  printat       ; print it
05E6  3A 035D            ld    a,(disknum)   ; load disk number
05E9  D6 40              sub   '@'           ; A = 1 for disk A or 2 for disk B
05EB  32 0126            ld    (dsksid),a    ; initialli, side is 0
05EE  AF                 xor   a             ; clear accumulator
05EF  32 0426            ld    (sidnum),a    ; set side 0 in ID field
05F2                     fdfok:
05F2  AF                 xor   a             ; clear accumulator
05F3  32 0425            ld    (trknum),a    ; set track 0
05F6  3C                 inc   a             ; A = 1
05F7  32 0427            ld    (secnum),a    ; set sector 1
05FA  3A 0126            ld    a,(dsksid)    ; load disk and side para
05FD  D3 D6              out   (fddlch),a    ; select drive and side
05FF  3E 02              ld    a,fdrest      ; load fdd restore command code
0601  D3 D0              out   (fddcmd),a    ; send out to 1771
0603                     fdfor1:
0603  CD 0774            call  waitfd        ; wait until end command
0606  B7                 or    a             ; zero in accumulator ?
0607  C2 07A0            jp    nz,timeout    ; if no zero then disk offline
                         ;
060A  CD 073E            call  sidtrkvis     ; visualize side and track
060D  21 03F6            ld    hl,preamble   ; H.L = preamble table (40 bytes 'ff')
0610  0E D7              ld    c,fdddat      ; C = fdd data register
0612  06 28              ld    b,40          ; bytes counter
0614  3E F4              ld    a,fddwtrk     ; load write track command code
0616  D3 D0              out   (fddcmd),a    ; send out to 1771
0618  CD 076F            call  fddelay       ; wait aproax 56 micro$
061B                     wtpreamb:
061B  DB D0              in    a,(fddsts)    ; load fdd status
061D  CB 4F              bit   1,a           ; test DRQ bit
061F  28 FA              jr    z,wtpreamb    ; wait if no DRQ
0621  ED A3              outi                ; write one byte
0623  20 F6              jr    nz,wtpreamb   ; repeat until all 40 bytes are output
```

```
0625                    fdfor2:
0625  21 041E                ld      hl,idfield      ; H.L = ID field data table
                        ;
                              ; total byte to output are 301.
                        ;
0628  06 00                 ld      b,0             ; 256 bytes to 1771
062A                    wtidfield:
062A  DB D0                 in      a,(fddsts)      ; load fdd status
062C  CB 4F                 bit     1,a             ; test DRQ bit
062E  28 FA                 jr      z,wtidfield     ; wait if no DRQ
0630  ED A3                 outi                    ; write one byte
0632  20 F6                 jr      nz,wtidfield    ; repeat until all 256 bytes are output
                        ;
0634  06 2D                 ld      b,301-256       ; load rimanents  bytes to 1771
0636                    wtiidfield:
0636  DB D0                 in      a,(fddsts)      ; load fdd status
0638  CB 4F                 bit     1,a             ; test DRQ bit
063A  28 FA                 jr      z,wtiidfield    ; wait if no DRQ
063C  ED A3                 outi                    ; write one byte
063E  20 F6                 jr      nz,wtiidfield   ; repeat until all 45 bytes are output
                        ;
                              ; now, all 301 bytes are output to 1771
                        ;
0640  3A 0427               ld      a,(secnum)      ; load actual sector number
0643  3C                    inc     a               ; inc. sector number
0644  32 0427               ld      (secnum),a      ; set next sector number
0647  FE 0B                 cp      11              ; last sector has been written ?
0649  C2 0625               jp      nz,fdfor2       ; no, then write next sector
064C                    endtrk:
064C  DB D0                 in      a,(fddsts)      ; load fdd status
064E  CB 47                 bit     0,a             ; end track ?
0650  28 07                 jr      z,nexttrk       ; yes, then go to next track
0652  3E FF                 ld      a,255           ; load byte 'ff'
0654  D3 D7                 out     (fdddat),a      ; write to 1771 data register
0656  C3 064C               jp      endtrk          ; repeat until end track
0659                    nexttrk:
0659  DB D0                 in      a,(fddsts)      ; load fdd status
065B  E6 E7                 and     11100111b       ; write track error?
065D  C2 0794               jp      nz,fdforerr     ; yes, then goto error
0660  3A 0425               ld      a,(trknum)      ; load track number
0663  3C                    inc     a               ; point to next track
0664  FE 28                 cp      40              ; end side ?
0666  28 0F                 jr      z,endside       ; yes, then go to end
0668  32 0425               ld      (trknum),a      ; set next track
066B  3E 01                 ld      a,1             ; load first sector number
066D  32 0427               ld      (secnum),a      ; set first sector number
                        ;
0670  3E 52                 ld      a,fddsin        ; load fdd step in
0672  D3 D0                 out     (fddcmd),a      ; send out to 1771
0674  C3 0603               jp      fdfor1          ; count for next track
                        ;
0677                    endside:
0677  3A 0127               ld      a,(fsideflag)   ; single or double side ?
067A  B7                    or      a               ;
067B  CA 0693               jp      z,enddsk        ; a = 0 then s.s. or end side two then jmp to format another dis
067E  AF                    xor     a               ; clear accumulator
067F  32 0127               ld      (fsideflag),a   ; reset side flag
0682  3C                    inc     a               ; A = 1
```

Sector

back

```
0683  32 0426            ld      (sidnum),a      ; set side one in ID field
0686  3A 035D            ld      a,(disknum)     ; load diak number
0689  D6 40              sub     '@'             ; 1 for A, 2 for B
068B  F6 20              or      00100000b       ; set side one
068D  32 0126            ld      (dsksid),a      ;
0690  C3 05F2            jp      fdfok           ; format side 1
                         ;
0693                     enddsk:
0693  AF                 xor     a               ; clear accumulator
0694  D3 D6              out     (fddlch),a      ; deselect any disk
                         ;
                         ; at this point, disk has been formatted
                         ;
0696  3A 0133            ld      a,(verflag)     ; load verify flag
0699  FE 59              cp      'Y'             ; is yes ?
069B  C2 072F            jp      nz,endanyop     ; no, then goto end disk
                         ;
069E                     dskverify:   con verify
069E  21 0A24            ld      hl,msg1add      ; message 1 cursor address
06A1  11 03B3            ld      de,vvisual      ; DE -) verify message
06A4  CD F024            call    printat         ; print it at HL
06A7  3A 0126            ld      a,(dsksid)      ; load disk and side
06AA  E6 03              and     00000011b       ; mask disk number
06AC  3D                 dec     a               ; disk A = 0, B = 1
                         ;
06AD                     dskve10:
06AD  32 0234            ld      (verdsk),a      ; start disk and side
06B0  AF                 xor     a               ; clear accumulator
06B1  32 0234            ld      (vertrk),a      ; start track 0
06B4                     dskve00:
06B4  3C                 inc     a               ;
06B5  32 0236            ld      (versec),a      ; start sector 1
06B8  11 0129            ld      de,xlt          ; DE -) translate table
                         ;
06BB                     vervis:
06BB  3A 0234            ld      a,(verdsk)      ; load verify disk and side
06BE  E6 10              and     00010000b       ; mask side
06C0  28 02              jr      z,verv00        ; jmp if side 0
06C2  3E 01              ld      a,1             ; set side 1
06C4                     verv00:
06C4  D5                 push    de              ; save xlt pointer
06C5  32 0426            ld      (sidnum),a      ; set ID side for visualization
06C8  3A 0234            ld      a,(vertrk)      ; load verify track
06CB  32 0425            ld      (trknum),a      ; set ID side for visualization
06CE  CD 073E            call    sidtrkvis       ; visualize side and track
                         ;
06D1  21 0234            ld      hl,vertab       ; verify tab para
06D4  CD F015            call    fdiod           ; read 256 bytes      F015
06D7  B7                 or      a               ; read error ?
06D8  28 16              jr      z,readok        ; no, then goto verify bytes
06DA                     readerr:
06DA  11 0294            ld      de,rdermsg      ; DE -) read error message <-- avi
06DD                     rder00:
06DD  21 0E19            ld      hl,ermsgadd     ; error message address
06E0  CD F024            call    printat         ; print it
06E3  11 02B8            ld      de,wterr        ; print 'Hit any key'
06E6  CD F01E            call    print
06E9  CD F003            call    cin
```

```
06EC  D1                    pop     de              ; restore xlt pointer
06ED  C3 054B               jp      fddform         ; and retry
06F0                  readok:
06F0  21 0134               ld      hl,verbuff      ; HL -) verify buffer
06F3  01 0100               ld      bc,256          ; num. byte to verify
06F6                  verif00:
06F6  7E                    ld      a,(hl)          ; load one byte
06F7  FE E5                 cp      0e5h            ; IBM standard data ?
06F9  28 05                 jr      z,verok         ; yes, then count for next data
06FB                  vererr:
06FB  11 02A6               ld      de,verrmsg      ; DE -) verify error message
06FE  18 DD                 jr      rder00          ; print it and retry
0700                  verok:
0700  23                    inc     hl              ; point to next data
0701  0B                    dec     bc              ; dec. data counter
0702  78                    ld      a,b             ;
0703  B1                    or      c               ; all 256 bytes are verify ?
0704  20 F0                 jr      nz,verif00      ; no, then count
0706                  endverify:
0706  D1                    pop     de              ; DE -) xlt pointer
0707  1A                    ld      a,(de)          ; load verify sector number
0708  13                    inc     de              ; point to next sector
0709  32 0236               ld      (versec),a      ; set next sector
070C  FE 00                 cp      0               ; end track ?
070E  20 AB                 jr      nz,vervis       ; no, then count
0710  3A 0234               ld      a,(vertrk)      ; load verify track
0713  3C                    inc     a               ; point to next track
0714  32 0234               ld      (vertrk),a      ; set next track
0717  FE 28                 cp      40              ; end side ?
0719  3E 00                 ld      a,0             ; A = 0 for sector set
071B  20 97                 jr      nz,dskve00      ; no end side, count to next track
071D                  vendside:
071D  3A 0128               ld      a,(vsideflag)   ; load verify side flag
0720  B7                    or      a               ; single side or end disk ?
0721  28 0C                 jr      z,endanyop      ; yes, then retry
0723  AF                    xor     a               ; clear accumulator
0724  32 0128               ld      (vsideflag),a   ; clear verify side flag
0727  3A 0234               ld      a,(verdsk)      ; load verify disk
072A  F6 10                 or      00010000b       ; set side 1
072C  C3 06AD               jp      dskve10         ; and go to verify side one
                      ;
072F                  endanyop:
                            ; entry point to go to format another disk
072F  06 02                 ld      b,2             ; set soft timer 2
0731                  endan00:
0731  11 0000               ld      de,0            ; set soft timer 1
0734                  endan11:
0734  1B                    dec     de              ; timer 1 down
0735  7A                    ld      a,d             ;
0736  B3                    or      e               ; zero for timer 1 ?
0737  20 FB                 jr      nz,endan11      ; no, then count
0739  10 F6                 djnz    endan00         ; timer 2 down and repeat until timer 2 = 0
073B  C3 054B               jp      fddform         ; go to format another disk
                      ;
                      ;
073E                  sidtrkvis:
                            ; visualize side and track
073E  21 0C26               ld      hl,snumadd      ; side video addr.
```

```
0741  CD F027                call    movcurs         ; move cursor
0744  3A 0426                ld      a,(sidnum)      ; load ID side number
0747  C6 30                  add     a,'0'           ; convert ASCII
0749  4F                     ld      c,a             ;
074A  CD F006                call    cout            ; print side 0 or 1
074D  21 0C30                ld      hl,tnumadd      ; track video adrr.
0750  CD F027                call    movcurs         ; move cursor
0753  3A 0425                ld      a,(trknum)      ; load ID track number
0756  06 FF                  ld      b,255           ; set decimal counter
0758                  sidtr00:
0758  04                     inc     b               ; inc. decimal digit
0759  D6 0A                  sub     10              ; A=A-10
075B  30 FB                  jr      nc,sidtr00      ; count if A )= 0
075D  C6 0A                  add     a,10            ; at this point A = lsd, B = msd (BCD)
075F  F5                     push    af              ; save lsd
0760  78                     ld      a,b             ; load msd
0761  C6 30                  add     a,'0'           ; convert ascii
0763  4F                     ld      c,a             ;
0764  CD F006                call    cout            ; print msd
0767  F1                     pop     af              ;
0768  C6 30                  add     a,'0'           ; convert ascii
076A  4F                     ld      c,a             ;
076B  CD F006                call    cout            ; print lsd
076E  C9                     ret                     ; and ret
                      ;
                      ;
076F                  fddelay:
                             rept    4
                             ex      (sp),hl         ; delay beetwen write command reg.
                             endm                    ; to read status reg.
076F  E3              +      ex      (sp),hl         ; delay beetwen write command reg.
0770  E3              +      ex      (sp),hl         ; delay beetwen write command reg.
0771  E3              +      ex      (sp),hl         ; delay beetwen write command reg.
0772  E3              +      ex      (sp),hl         ; delay beetwen write command reg.
0773  C9                     ret
                      ;
                      ;
0774                  waitfd:
                             ; wait until fdd busy is reset.
0774  CD 076F                call    fddelay         ; wait aproax 56 micro5
0777  06 02                  ld      b,2             ; set soft timer
0779                  wait00:
0779  11 0000                ld      de,0            ; for aproax five seconds
077C                  wait01:
077C  DB D0                  in      a,(fddsts)      ; input to fdd status
077E  CB 47                  bit     0,a             ; test busy bit
0780  28 0F                  jr      z,wait02        ; jump if no command is in progress
0782  1B                     dec     de              ;
0783  7A                     ld      a,d             ; timer down
0784  B3                     or      e               ;
0785  20 F5                  jr      nz,wait01       ;
0787  05                     dec     b               ;
0788  20 EF                  jr      nz,wait00       ; time out
078A                  ofline:
078A  3E D0                  ld      a,fddrst        ; reset fdd controller
078C  D3 D0                  out     (fddcmd),a      ; exec. command
078E  3E 01                  ld      a,00000001b     ; set time-out bit error
0790  C9                     ret                     ; and ret
```

```
0791                    wait02:
0791  47                        ld      b,a             ; save fdd status in B register
0792  AF                        xor     a               ; clear accumulator for
0793  C9                        ret                     ; normal return
                        ;
0794                    fdforerr:
0794  21 0E19                   ld      hl,ermsgadd     ; error message address
0797  11 0285                   ld      de,fmtermsg     ; DE = format error message
079A  CD F024                   call    printat         ; print it
079D  C3 07A9                   jp      retry           ; and go to cp/m
                        ;
07A0                    timeout:
07A0  21 0E19                   ld      hl,ermsgadd     ; error message address
07A3  11 0276                   ld      de,oflmsg       ; DE = offline message
07A6  CD F024                   call    printat         ; print it
07A9                    retry:
07A9  11 023A                   ld      de,setnew       ; DE = set new disk message
07AC  CD F01E                   call    print           ; print it
07AF  CD F003                   call    cin             ; wait one char.
07B2  C3 0000                   jp      0               ; and return to cp/m
                        ;
                        ;
                        ;
                        ;
                                end     100h            ; end of floppy disk formatter
```

Macros:

Symbols:

BACKSP	0008	BELL	0007	BOOTRO	F021	CIN	F003
COMPFL	F02D	COPYRI	0000	COUT	F006	CR	000D
CSTS	F009	DBLMSG	03EB	DISKNU	035D	DSKSID	0126
DSKVE0	06B4	DSKVE1	06AD	DSKVER	069E	ENDAN0	0731
ENDAN1	0734	ENDANY	072F	ENDDSK	0693	ENDMSG	0024
ENDSID	0677	ENDTRK	064C	ENDVER	0706	ERMSGA	0E19
FALSE	0000	FDBUSY	0001	FDDCMD	00D0	FDDDAT	00D7
FDDELA	076F	FDDF00	0557	FDDFOR	054B	FDDLCH	00D6
FDDRES	0002	FDDRQ	0002	FDDRST	00D0	FDDSEC	00D2
FDDSIN	0052	FDDSTS	00D0	FDDTAB	03F6	FDDTRK	00D1
FDDWTR	00F4	FDFOK	05F2	FDFOR1	0603	FDFOR2	0625
FDFORE	0794	FDIOD	F015	FDIOS	F012	FDNRDY	0080
FDT1ER	0018	FDT23E	001F	FDWPRT	0040	FFEED	000C
FLASH	0043	FMTERM	0285	FORDBL	056F	FORM2	05A2
FORM3	05CE	FORM7	0583	FORM8	0596	FORMSD	0574
FORSNG	0569	FSIDEF	0127	FVISUA	03A3	IDFIEL	041E
INIMSG	02C8	LF	000A	LOUT	F00C	LSTS	F00F
MOVCUR	F027	MSG1AD	0A24	MSG2AD	0C21	NEXTTR	0659
NOMSG	03D1	NORM	0040	OFLINE	078A	OFLMSG	0276
PFX	0013	PREAMB	03F6	PRINT	F01E	PRINTA	F024
QUEST1	02F5	QUEST2	0322	QUEST3	0342	QUEST4	0380
RDER00	06DD	RDERMS	0294	READER	06DA	READOK	06F0
RETRY	07A9	REVER	0048	ROM	F000	SECNUM	0427
SETNEW	023A	SIDNUM	0426	SIDTR0	0758	SIDTRK	073E
SNGMSG	03E0	SNUMAD	0C26	SPACE	0020	STACK	0126
STROUT	F01E	TAB	0009	TIMEOU	07A0	TNUMAD	0C30
TRKNUM	0425	TRUE	0001	VENDSI	071D	VERBUF	0134
VERDMA	0237	VERDSK	0234	VERERR	06FB	VERFLA	0133
VERIF0	06F6	VEROK	0700	VEROP	0239	VERRMS	02A6
VERSEC	0236	VERTAB	0234	VERTRK	0234	VERV00	06C4
VERVIS	06BB	VIDINI	F02A	VSIDEF	0128	VSIDTR	03C3
VVISUA	03B3	WAIT00	0779	WAIT01	077C	WAIT02	0791
WAITFD	0774	WDINI	F018	WDIO	F01B	WT1IDF	0636
WTERR	02B8	WTIDFI	062A	WTPREA	061B	XLT	0129
YESMSG	03D8						

No Fatal error(s)