


```

InterTable: .DO W_10MB
            .DB 2, 1, 7, 10, 8, 13, 3
            .ELSE
InterTable: .DB 2, 1, 26, 10, 8, 13, 22
            .FIN

FormatTrack:
            Ld !r2,#.HIBYTE. Zero_Fmt
            Ld !r3,#.LOWBYTE. Zero_Fmt
            Call Bank_Call
            Or DiskStat,#Wr_Op

FmtT_1:     Tm Port3,#IndexMark ;WHILE NOT( Index )
            Jr Z,FmtT_1

            And Port0,#$FF-AcEraseL ;turn on AC erase

FmtT_2:     Tm Port3,#IndexMark ;WHILE Index
            Jr Nz,FmtT_2

FmtT_3:     Tm Port3,#IndexMark ;WHILE NOT( Index )
            Jr Z,FmtT_3

            Or Port0,#AcEraseL ;turn off AC erase

            Or !r4,!r4 ;test for 0 offset
            Jr Z,Fmt_Begin

Fmt_Off1:   And !r4,#$FF-!r4_Sector ;clear old sector mark

Fmt_Off2:   Tm !r4,#!r4_Sector ;wait for mark to pass
            Jr Z,Fmt_Off2

            Djnz !r4,Fmt_Off1

Fmt_Begin:  Ld !r2,#.HIBYTE. InterTable
            Ld !r3,#.LOWBYTE. InterTable
            Add !r3,!r5 ;get offset into table
            Adc !r2,#0

            Ldc !r6,@!r2
            Clr Sector ;sector := 0

            Ld !r5,#NbrSctrs

FmtTrk_2:   Ld !r1,#Dmt_FmtTrk
            Call FormatBlock
            Jr Nz,FmtT_Until

            Ld !r9,!r0 ;pass reason for abort
            Call Abort

FmtT_Until: Add Sector,!r6 ;Sector := Sector + InterLeaveFactor
            Cp Sector,#NbrSctrs ;Sector := Sector MOD NbrSctrs
            Jr Lt,FmtT2_Until

            Sub Sector,#NbrSctrs ;do MOD by subtraction

FmtT2_Until: Djnz !r5,FmtTrk_2

            Jp Bank_Ret

.LSTOFF

```

```

            .DD    External
            .LSTON
            .Page
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;
;> Procedure: LocateSector
;
;> This procedure returns to the caller just after the sector
;> mark representing the sector JUST BEFORE the sector of the
;> last seek address.
;>
;> Inputs: { none }
;>
;> Outputs: { none }
;>
;> Global Variables Used:
;>     Sector
;>
;> Algorithm:
;>
;> BEGIN
;>     Offset := SpareTable.FmtOffset
;>     InterL := SpareTable.InterLeave
;>     SectorCount := 0
;>     Temp := 0
;>     WHILE NOT( Index ) DO BEGIN END
;>     WHILE ( Offset <> 0 ) DO
;>         Wait For End of Sector Mark
;>     WHILE ( Temp MOD NbrSctrs <> Sector ) DO
;>         Temp := Temp + InterL
;>         i := i + 2 { count sectors at 2:1 interleave }
;>     WHILE ( i <> 0 ) DO
;>         Wait For End Of Sector Mark
;>         i := i - 1
;>     END
;>
;>>>>>>>>>>>>>>>>>>>>
            .LSTOFF
            .FIN
            .DD    Internal
            .LSTON
            .Page
            .FIN

```

LocateSector:

```

    Ld    !r0,#Dmt_LctSctr
    Call  Set_Dmt

    Push  Rp ;save context
    Srp   #Wrk_Sys2

    Ld    !r2,!.HIBYTE. Fmt_Offset
    Ld    !r3,!.LOWBYTE. Fmt_Offset
    Lde   !r4,@!!r2 ;load offset value
    Or    Head,Head ;check for even/odd head
    Jr    Z,Lct_Offset
    Swap  !r4
Lct_Offset:
    And   !r4,#$0F

    Incw  !!r2 ;point to interleave value
    Lde   !r5,@!!r2

    Ld    !r2,!.HIBYTE. InterTable

```

```

Ld      !r3,*.LOWBYTE. InterTable
Add     !r3,!r5 ;add in interleave value
Acl     !r2,*0
Ldc     !r5,@!!r2 ;get real interleave value

LetSctr1:  Tm      Port3,*IndexMark ;WHILE Index DO BEGIN END
           Jr      Nz,LetSctr1

LetSctr2:  Tm      Port3,*IndexMark ;WHILE Not( Index ) DO BEGIN END
           Jr      Z,LetSctr2

           And     !r4,$FF-!r4_Sector ;clear any old sector marks

           Or      !r4,!r4 ;IF Offset = 0 THEN we're already here!
           Jr      Z,LetSctr4

           Ld      !r2,!r4
           Call    LetSctr3

LetSctr4:  Or      Sector,Sector ;IF Sector = 0 THEN We're already there
           Jr      Z,LetDone

           Clr     !r2
           Clr     !r4

LetSctr5:  Cp      !r4,Sector ;see if we've found the sector
           Jr      Z,LetSctr6
           Add     !r2,*2
           Add     !r4,!r5 ;temp := Temp + InterLeave Factor
           Cp      !r4,*NbrSctrs ;temp := temp MOD NbrSctrs
           Jr      Lt,LetSctr5
           Sub     !r4,*NbrSctrs
           Jr      LetSctr5

LetSctr6:  Call    LetSctr3

LetDone:   Call    Clr_Dmt
           Pop     !r4 ;restore original context
           Jp      Bank_Ret

LetSctr3:  Tm      !r4,*!r4_Sector ;wait for next sector
           Jr      Z,LetSctr3
           And     !r4,$FF-!r4_Sector ;mask old interrupt
           Djnz   !r2,LetSctr3 ;count the number of sector in offset
           Ret

```

.LSTOFF