```
c81.lnk
.report       3
.output       CBM1581
.header       c81.hdr.rel
.vlir

.psect $6000
c81.install.rel

.mod 1
.psect $9000
c81.drv.rel
c81.turbo.rel
c81_128.lnk
.report       3,2
.output       CBM1581_128
.header       c81.hdr.rel
.vlir

.psect $6000
c81.install.rel

.mod 1
.psect $9000
c81.drv.rel
c81.turbo.rel
c81.turbo.lnk
.report       2,3
.output       81-Turbo.bin
.cbm
.psect $0300
c81.turbo.rel
```

```
;c81.drv.s
.include c81.inc            ;9000 <=                    ;59
HDR_NOT_THERE = $20
TB_0300              = $0300       ;Load address in drive for turbo
AddDirBlock          == $9039
D_8867        == $8867
D_D07A        == $D07A      ;Fast SCPU
D_D07B        == $D07B      ;Slow SCPU
D_D0B8        == $D0B8

RNDX=$8B
.zsect RNDX                 ;RND function seed value location 5 bytes
       zBufPtr:      .block 2      ;BC
       zSerialData:  .block 1      ;D
       zLowPort:     .block 1      ;E
       zRcsSend00:   .block 1      ;F
       ;---   Redefined for use at a non conflicting time with prior
.zsect RNDX                 ;RND function seed value location 5 bytes
                     .block 2
                     .block 1
       zRcsSnd11:    .block 1
       t1L:          .block 1

.psect
;      DISK_BASE:
              .word  _InitForIO
              .word  _DoneWithIO
              .word  _ExitTurbo
              .word  _PurgeTurbo
              .word  _EnterTurbo
              .word  _ChangeDiskDev
              .word  _NewDisk
              .word  _ReadBlock
              .word  _WriteBlock
              .word  _VerWrtBlck
              .word  _OpenDisk
              .word  _GetBlock
              .word  _PutBlock
              .word  _GetDirHead
              .word  _PutDirHead
              .word  _GetFreeDirBlk
              .word  _CalcBlksFree
              .word  _FreeBlock
              .word  _SetNextFree
              .word  _FindBamBit
              .word  _NxtBlkAlloc
              .word  _BlkAlloc
              .word  _ChkDkGeos
              .word  _SetGEOSDisk
              jmp    _Get1stDirEntry
              jmp    _GetNxtDirEntry
              jmp    _GetOffPageTrSc
              jmp    _AddDirBlock
              jmp    _GetBufBlock
              jmp    _PutBufBlock
       CallDrvRout:
              nop
              nop
              rts
       CheckDrvStatus:
              jmp    _ChkStatus
```

```
               jmp    _AllocateBlock
               jmp    _ReadLink
        drvType:       .byte 3
        drvVer:        .byte $54
        OpenRoot:
               jmp    _OpenRoot
        OpenDir:
               jmp    _OpenRoot
        GetBAMBlk:
               nop
               nop
               rts
        PutBAMBlk:
               nop
               nop
               rts
        dirHeadTr:     .byte 40
        dirHeadSec:    .block 1
        curBamBlock:   .block 1
        lastBamByte:   .block 1
        lastBamSec:    .block 1
        bamAltered:    .block 1
        highestTr:     .byte 80
               jmp    _GetHeadTS
               jmp    _PutHeadTS
               jmp    _GetLink
               jmp    _GetSysDirBlk
        startBank:     .block 1
        startPage:     .block 1
        pagesUsed:     .block 3
        dirEnd:        .block 1
        blksFree:      .block 2                          ;9075
        lastAlocTS:    .byte 41,0
                       .block 1
        dskClosed:     .byte $FF
        SendTab:       .byte $00,$10,$20,$30,$00,$10,$20,$30          ;907B
                       .byte $00,$10,$20,$30,$00,$10,$20,$30
        Send2Tab:      .byte $00,$00,$00,$00,$10,$10,$10,$10          ;908B
                       .byte $20,$20,$20,$20,$30,$30,$30,$30
        _PutDirHead:
               lda    #[WriteBlock
clda   _GetDirHead,   #[ReadBlock
               sta    DoBlock+1
               jsr    EnterTurbo
               bne    96$     ;-> Error
               jsr    InitForIO
               jsr    SDirHead
               jsr    DoBlock
               bne    94$     ;-> Error
               inc    r1H
               LoadB  r4H,#]dir2Head
               jsr    DoBlock
               bne    94$     ;-> Error
               inc    r1H
               LoadW  r4,#dir3Head
               jsr    DoBlock
        94$    jsr    DoneWithIO
        96$    jsr    SDirHead
               txa
               rts
        _PutBufBlock:
```

```
              ldy    #[WriteBlock
cldy   _GetBufBlock, #[ReadBlock
              jsr    SDiskBlkBuf
cldy   _PutBlock,    #[WriteBlock
cldy   _GetBlock,    #[ReadBlock
              sty    DoBlock+1
              jsr    EnterTurbo
              bne    99$
              jsr    InitForIO
              jsr    DoBlock
              jsr    DoneWithIO
              txa
       99$    rts

       DoBlock:
              jmp    ReadBlock
       _GetLink:
              jsr    EnterTurbo
              jsr    InitForIO
              jsr    ReadLink
              jsr    DoneWithIO
              txa
              rts
       _GetHeadTS:
              LoadB  r2,#1
       _PutHeadTS:
              LoadB  r1,#40
              sta    dirHeadTr
              ldx    #0
              stx    r1H
              stx    dirHeadSec
              rts
       _OpenRoot:
              jsr    _PutHeadTS
       _OpenDisk:
              ldx    #r5
              jsr    GetPtrCurDkNm
              LoadB  lastAlocTS,#41
              LoadB  lastAlocTS+1,#0
              sta    isGEOS
              tay
              sta    (r5),y
              jsr    NewDisk
              bne    20$
              jsr    GetDirHead
              bne    20$
              ldx    #HDR_NOT_THERE
              CmpB   dir3Head+2,#$44
              bne    20$
              CmpB   dir3Head+3,#$BB
              bne    20$
              ldy    #17
       10$    MoveB  curDirHead+OFF_DISK_NAME,y ,(r5),y
              dey
              bpl    10$
              jsr    CalcFree
              jsr    ChkDkGEOS
              ldx    #0
       20$    stx    dskClosed
              jsr    SaveVars
              ldx    dskClosed
```

```
SDirHead:
       LoadW  r4,#curDirHead
       LoadB  r1,#40
       LoadB  r1H,#0
       rts
_BlkAlloc:
       bbne   lastAlocTS,10$
       lda    #41
10$    sta    r3
       MoveB  lastAlocTS+1,r3H
_NxtBlkAlloc:
       PushW  r9
       CmpIW  #32258,r2
10$    ldx    #3
       bcc    50$
       LoadW  r5,#254       ;Size of data in block
       ldx    #r2
       ldy    #r5
       jsr    Ddiv
       bbeq   r8,20$
       inc    r2
20$    bbne   blksFree+1,30$
       CmpB   blksFree,r2
       blt    10$
30$    MoveW  r6,r4
       MoveB  r2,r5
40$    jsr    SetNextFree
50$    bne    80$
       ldy    #0
       MoveB  r3,lastAlocTS
       sta    (r4),y
       iny
       MoveB  r3H,lastAlocTS+1
       sta    (r4),y
       AddVW  #2,r4
       dec    r5
       bne    40$
       dey
       tya
       tax
       sta    (r4),y
       iny
       bbne   r8,70$
       lda    #$FE
70$    add    #1
       sta    (r4),y
80$    PopW   r9
       rts
_Get1stDirEntry:
       LoadB  r1H,#3
       LoadB  r1,#40
       LoadB  dirEnd,#0
       jmp    GDirBlock
_GetNxtDirEntry:
       ldx    #0
       ldy    #0
       AddB   #32,r5
       bcc    90$
       dey
       jsr    GetLink
       bne    GDirBlock
```

```
        bbsf    7,dirEnd,90$
        sty     dirEnd
        jsr     _GetOffPageTrSc
        bne     99$
GDirBlock.                                      ;9232
        jsr     GetBufBlock
        ldy     #0
        LoadW   r5,#diskBlkBuf+2
90$
99$     rts
GetLink:                                        ;9240
        MoveW   diskBlkBuf,r1
        rts

SDiskBlkBuf:
        LoadW   r4,#diskBlkBuf
        rts
_GetOffPageTrSc:
        jsr     GetDirHead
        bne     99$
        jsr     ChkDkGEOS
        MoveW   curDirHead+OFF_OP_TR_SC,r1
        lda     isGEOS
        eor     #$FF
        tay
99$     rts
_ChkDkGeos:
        ldy     #10
10$     CmpB    curDirHead+OFF_GS_ID,y ,formatID,y
        bne     20$
        dey
        bpl     10$
        tya
clda    20$,            #0
        sta     isGEOS
        rts

formatID:       .byte "GEOS format"," V1.1"
_GetSysDirBlk:
        jsr     _GetOffPageTrSc
        beq     10$
        ldx     #FULL_DIRECTORY
        rts

_GetFreeDirBlk.
        MoveW   curDirHead,r1
        ldy     #0
cldy    10$,            #TRUE
        PushW   r6
        sty     r6H
        ldx     r10
        inx
        stx     r6
20$     jsr     GetBufBlock
30$     bxne    98$
        dec     r6
        beq     60$
40$     bbne    diskBlkBuf,50$
        jsr     AddDirBlock
        bra     30$
```

```
        50$     sta     r1
                MoveB   diskBlkBuf+1,r1H
                bra     20$
        60$     ldy     #2
                ldx     #0
        70$     bbeq    diskBlkBuf,y ,98$
                tya
                add     #32
                tay
                bcc     70$
                LoadB   r6,#1
                ldx     #FULL_DIRECTORY
                inc     r10
                ldy     r10
                beq     98$
                bbsf    7,r6H,40$
                cpy     #37
                blt     40$
        98$     PopW    r6
                rts
        _AddDirBlock:
                PushW   r6
                MoveW   r1,r3
                jsr     SetNextFree
                bne     98$
                MoveW   r3,diskBlkBuf
                jsr     PutBufBlock
                bne     98$
                jsr     ClearAndWrite
        98$     PopW    r6
                rts
        ClearAndWrite:                                  ;932C
                MoveW   r3,r1
                lda     #0
                tay
        10$     sta     diskBlkBuf,y
                iny
                bne     10$
                dey
                sty     diskBlkBuf+1
                jmp     PutBufBlock
        _SetNextFree:
                MoveB   r3,r6
                MoveB   r3H,r6H
                jsr     CheckTS
                beq     10$
                LoadB   r6,#41
                LoadB   r6H,#0
        10$     jsr     J_93A8
                bne     20$
        88$     txa
                rts

        20$     CmpB    r6,#40
                beq     88$
                jsr     J_9379
                beq     88$
                CmpB    D_93A7,#40
                bcc     30$
                lda     #39
clda    30$,            #41
```

```
              sta     r6
       J_9379:
              sta     D_93A7
       40$    LoadB   r6H,#19
              jsr     J_93A8
              beq     90$
              lda     #0
              ldx     D_93A7
              cpx     #41
              bge     50$
              lda     #$FF
       50$    adc     r6
              beq     60$
              cmp     #81
              bcc     75$
              lda     #41
clda   60$,           #39
       75$    sta     r6
              cmp     D_93A7
              bne     40$
              ldx     #INSUFF_SPACE
       90$    rts

       D_93A7:       .block 1
       J_93A8:
              jsr     J_93BB
              bne     10$
              rts

       10$    CmpB    r6H,#20
              blt     20$
              lda     #19
clda   20$,           #39
              sta     r6H
       J_93BB:
              ldx     r6H
              stx     startSector
       30$    inx
              cpx     #40
              bge     40$
              cpx     #20
              bne     50$
              ldx     #0
              CmpB    r6,#1
              bne     50$
              cpx     startSector
              beq     98$
              inx
cldx   40$,           #20
       50$    stx     r6H
              jsr     AllocateBlock
              bne     60$
              MoveW   r6,r3
              txa
              rts

       60$    ldx     r6H
              cpx     startSector
              bne     30$
       98$    ldx     #INSUFF_SPACE
              rts
```

```
startSector:  .block 1                          ;93F5
_FreeBlock:
       jsr    CheckTS
       bne    rts_0
       jsr    FindBAMBit
       bne    bad_bam
       php
       inc    blksFree
       bne    DoBlockOp
       inc    blksFree+1
       bne    DoBlockOp
       plp
bad_bam                                 =*
       ldx    #BAD_BAM
rts_0                                   =*
       rts
CheckTS:
       CmpB   r6H,#40
       bge    96$    ;-> Bad sector
       jsr_a  ValTrack,r6
       beq    98$    ;-> Good Track
96$    ldx    #INV_TRACKS
98$    rts
_AllocateBlock:
       jsr    CheckTS
       bne    rts_0
       jsr    FindBAMBit
       beq    bad_bam
       php
       DecW   blksFree
DoBlockOp:
       CmpB   r6,#41
       lda    r8H
       blt    20$
       eor    dir3Head+16,x
       sta    dir3Head+16,x
       ldx    r7H
       plp
       beq    10$
       dec    dir3Head+16,x
       ldx    #NO_ERROR
       rts

10$    inc    dir3Head+16,x
       ldx    #NO_ERROR
       rts

20$    eor    dir2Head+16,x
       sta    dir2Head+16,x
       ldx    r7H
       plp
       beq    80$
       dec    dir2Head+16,x
       ldx    #NO_ERROR
       rts

80$    inc    dir2Head+16,x
       ldx    #NO_ERROR
       rts
_VerWrtBlck:
```

```
       lda    r1
ValTrack:
       tax
       beq    98$
       cmp    #81
       bge    98$
       ldx    #NO_ERROR
       rts

98$    ldx    #INV_TRACKS
       rts
_FindBamBit:
       lda    r6H
       and    #%0111
       tax
       lda    bitMask,x
       sta    r8H
       CmpB   r6,#$29
       php
       bcc    10$
       sbc    #40
10$    sub    #1
       asl    a
       sta    r7H
       asl    a
       adc    r7H
       sta    r7H
       lda    r6H
       lsr    a
       lsr    a
       lsr    a
       sec
       adc    r7H
       tax
       lda    dir2Head+16,x
       plp
       bcc    12$
       lda    dir3Head+16,x
12$    and    r8H
       rts

bitMask:
       .byte  $01,$02,$04,$08,$10,$20,$40,$80
_CalcBlksFree:
       LoadW  r3,#3160
       MoveW  blksFree,r4
       rts
CalcFree:                                   ;94C8
       LoadB  r1,#0
       sta    r1H
       sta    zBufPtr
       LoadB  zBufPtr+1,#]dir2Head
       jsr    CalcBBFree
       LoadW  zBufPtr,#dir3Head
       jsr    CalcBBFree
       SubBWS2       dir2Head+250,r1,blksFree
       rts
CalcBBFree:                                 ;94F
       ldy    #16
10$    AddBW  (zBufPtr),y ,r1
       tya
```

```
              add     #6
              tay
              bne     10$
              rts
       _ChangeDiskDev:
              tax
              lda     #1
clda   _SetGEOSDisk, #2
              jmp     JmpIndX
       _InitForIO:
              php
              PopB    origFlags
              sei
.if !C128
              MoveB   CPU_DATA,origMMap
              LoadB   CPU_DATA,KRNL_IO_IN
.endif
              MoveB   grirqen,origGrirgen
              MoveB   clkreg,origClock
              MoveB   mobenble,origSpriteReg
              ldy     #0
              sty     clkreg
              sty     grirqen
              LoadB   grirq,#%01111111
              sta     cia1icr
              sta     cia2icr
              bit     cia1icr
              bit     cia2icr
.if C128
              LdWW2   irqvec,nmivec,#$9FF6
.else
              LoadW   irqvec,#$9FFA
              LoadW   nmivec,#$9FFF
.endif
              LoadB   cia2ddra,#%00111111
              sty     mobenble
              sty     cia2tahi
              iny
              sty     cia2talo
              LoadB   cia2icr,#%10000001
              setbit  cia2cra,#%10000000,#%1001
              ldy     #44
       10$    CmpB    rasreg,t1L
              beq     10$
              sta     t1L
              dey
              bne     10$
       InitSerial:                          ;      957E
              lda     cia2pra
              and     #%0111
              sta     zLowPort
              ora     #%00010000
              sta     zRcsSend00
              rts
       _DoneWithIO:
              sei
              MoveB   origClock,clkreg
              MoveB   origSpriteReg,mobenble
              LoadB   cia2icr,#$7F
              lda     cia2icr
              MoveB   origGrirgen,grirqen
```

```
.if !C128
              MoveB   origMMap,CPU_DATA
.endif
              PushB   origFlags
              plp
              rts

       origClock:     .block 1
       origFlags:     .block 1
.if !C128
       origMMap:      .block 1
.endif
       origGrirgen:   .block 1
       origSpriteReg:       .block 1
       blkSize:       .block 1
.if !C128
                      .block 1
.else
                      .block 3
.endif
       tbStatus:      .block 1                        ;95B3
       SendCommand:
              sty     cmdSize
              jsr     UNLSN
              LoadB   STATUS,#0
              jsr_a   LISTEN,curDrive
              jsr_a   SECOND,#%01100000 | 15
              bbsf    7,STATUS,98$
              ldy     #0
       10$    jsr_a   CIOUT,(zBufPtr),y
              iny
              cpy     cmdSize
              blt     10$
              bbsf    7,cmdResult,88$
              jsr_a   CIOUT,#CR
              jsr     UNLSN
       88$    ldx     #NO_ERROR
              stx     cmdResult
              rts

       98$    jsr     UNLSN
              ldx     #DEV_NOT_FOUND
              stx     cmdResult
              rts

       cmdResult:     .block 1
       cmdSize:       .block 1
       _EnterTurbo:
              jsr_a   SetDevice,curDrive
              ldx     curDrive
              bbmi    turboFlags-8,x ,10$
              jsr     InstallTurboCode
              bne     98$
              ldx     curDrive
              LoadB   turboFlags-8,x ,#%10000000
       10$    ldx     #0
              and     #%01000000
              bne     98$
              jsr     InitForIO
              jsr     SMemExe
              ldy     #5
```

```
              jsr    SendCommand
              sei
              jsr    SCPUSlow
              ldx    #33
       20$    dex
              bne    20$
              jsr    EndSerial
              ldx    #0
              ldy    curDrive
              LoadB  turboFlags-8,y ,#%11000000
              jsr    DoneWithIO
       98$    txa
              rts
       InstallTurboCode:
              jsr    InitForIO
              jsr    SMemWrite
              LoadW  zSerialData,#TurboCode
       10$    ldy    #6
              LoadB  cmdResult,#TRUE
              jsr    SendCommand
              bne    98$
              ldy    #0
       20$    jsr_a  CIOUT,(zSerialData),y
              iny
              cpy    #32
              blt    20$
              jsr_a  CIOUT,#CR
              jsr    UNLSN
              AddVW2 #32,MRWhere
              AddVW2 #32,zSerialData
              CmpWI  zSerialData,#TurboEnd  ;End of turbo code D_9B6A
              blt    10$
              ldx    #NO_ERROR
       98$    jsr    DoneWithIO
              txa
              rts
       SMemWrite:
              lda    #'W'
clda   SMemExe,       #'E'
              sta    memRdSyn+2
              LoadB  MRWhere,#[TB_0300
              LoadB  MRWhere+1,#]TB_0300
              LoadW  zBufPtr,#memRdSyn
              rts

       memRdSyn:
              .byte  "M-W"

       MRWhere:
              .word  TB_0300
              .byte  ' '                    ; End of cmd
       _ExitTurbo:
              PushX
              jsr    ExitTur_
              jsr    SaveVars
              PopX
              rts

       ExitTur_:
              ldx    curDrive
              lda    turboFlags-8,x
```

```
        and     #%01000000
        beq     90$
        jsr     InitForIO
        ldx     #0              ;InitDisk
        jsr     CallCmd
        jsr     WaitOnDiskRdy
        jsr     DoneWithIO
        ldx     curDrive
        rmbf2   6,turboFlags-8,x
90$     rts
_PurgeTurbo:
        jsr     ExitTurbo
        ldy     curDrive
        LoadB   turboFlags-8,y,#0
        tax
        rts
InitializeDisk:                         ;96F6
        jsr     InitForIO
        LoadW   zBufPtr,#cmd_I
        ldy     #1
        jsr     SendCommand
        jsr     DoneWithIO
        jmp     SaveVars

cmd_I: .byte 'I'                        ;970C
SaveVars:
        ldx     #0
10$     PushB   r0,x
        MoveB   T_9735,x ,r0,x
        inx
        cpx     #7
        bcc     10$
        ldx     curDrive
        MoveB   T_TSL-8,x,r1
        MoveB   T_TSH-8,x,r1H
        jsr     StashRAM
        ldx     #7
20$     PopB    r0-1,x
        dex
        bne     20$
        rts

T_9735:
        .word   $905C
        .word   $8300
        .word   31
        .byte   0

T_TSL:
        .byte   <$835C                  ;REU Addresses into driver cache
        .byte   <$90DC
        .byte   <$9E5C
        .byte   <$ABDC
T_TSH:
        .byte   >$835C
        .byte   >$90DC
        .byte   >$9E5C
        .byte   >$ABDC
Wait_DiskRdy:
        jsr     SCPUSlow
WaitOnDiskRdy:
```

```
              MoveB  zLowPort,cia2pra
       10$    bbrf   7,cia2pra,10$
              rts
       EndSerial:                          ;                                    9752
              ldx    zRcsSend00
              stx    cia2pra
       10$    bbsf   7,cia2pra,10$
              bbsf   7,D_8867,90$
.if !C128
              rmbf2  7,CPU_DDR            ;FIX why? bit 7 is not used.
.endif
              sta    D_D07B               ;Return to fast SCPU speed.
       90$    rts
       SCPUSlow:                           ;                                    9765
.if C128
              MoveB  D_D0B8,D_8867        ;SCPU Speed Flag 1=Normal/0=Turbo
              sta    D_D07A               ;Software Speed Select to Normal mode
              rts
.else
              CmpB   D_D0B8,#$FF          ;SCPU Speed Flag 1=Normal/0=Turbo
              bne    10$
              lda    CPU_DDR
       10$    sta    D_8867
              smbf2  7,CPU_DDR            ;Still not a bit in use
              sta    D_D07A               ;Software Speed Select to Normal mode
              rts
.endif
       _NewDisk:
              jsr    EnterTurbo
              LoadB  dirHeadTr,#40
              LoadB  dirHeadSec,#0
              jsr    LogDiskIn
              beq    90$
              jsr    ExitTurbo
              jsr    InitializeDisk       ;Send initialize command to drive to reload disk.
              jsr    EnterTurbo
              jsr    LogDiskIn
       90$    rts
       LogDiskIn:                          ;                                    978E
              jsr    InitForIO
              ldx    #5
              jsr    CallCmd
              jsr    DoneWithIO
              txa
              rts
       ReadFastSerial:                     ;                                    979B
              jsr    InitSerial
              jsr    Wait_DiskRdy
              ldx    zRcsSend00
       ReadLoop.
              sec
              nop                          ;wait 8us
              nop
              nop
              nop
       20$    lda    rasreg               ;get current raster position
              sbc    grcntrl1             ;Check for valid start time
              and    #%0111               ;Line - top of card row?
              cmp    #7
              bge    20$                  ;if so wait
              stx    cia2pra
```

```
        MoveB  zLowPort,cia2pra
        lda    cia2pra              ;get data
        lsr    a                    ;move it over
        lsr    a
        ora    cia2pra              ;get next 2 bits
        lsr    a                    ;move it over into the lower
        lsr    a
        eor    zLowPort             ;next 2
        eor    cia2pra
        lsr    a                    ;make room for another two bits
        lsr    a
        eor    zLowPort             ;last 2 bits
        eor    cia2pra
        dey
        sta    (zBufPtr),y          ;save byte
        bne    ReadLoop
70$     bit    cia2pra              ;wait for turbo code
        bpl    70$
        jmp    EndSerial
TcmdDat1:    .byte $00,$80,$80,$02,$00,$00        ;                          97DF
TcmdDat2:    .byte $80,$81,$01,$01,$01,$01        ;                          97E5

TcmdLAdr:    .byte [TExitTurbo     ;Turbo Code jump table locations          97EB
             .byte [TWriteSector
             .byte [TReadSector
             .byte [TReadLnk
             .byte [TGetStatus
             .byte [TLogDiskIn
TcmdHAdr:    .byte ]TExitTurbo     ;FIX these are ALL $03                    97F1
             .byte ]TWriteSector   ;useless high table
             .byte ]TReadSector
             .byte ]TReadLnk
             .byte ]TGetStatus
             .byte ]TLogDiskIn
CallBlkCmd:                        ;                                         97F7
        MoveW  r4,zBufPtr
cldx    _ChkStatus,   #4
CallCmd:                           ;                                         9802
        LoadB  tbStatus,#0
        MoveB  TcmdDat1,x,cmdDat1
        MoveB  TcmdDat2,x,cmdDat2
        MoveB  TcmdLAdr,x,cmdAddr
        MoveB  TcmdHAdr,x,cmdAddr+1
        MoveW  r1,cmdTS
        PushW  zBufPtr
        LoadW  zBufPtr,#cmdAddr
        ldy    #4
        jsr    SendFastSerial
        PopW   zBufPtr
        ldy    cmdDat1
        beq    30$
        bpl    10$
        ldy    #0
10$     bbsf   7,cmdDat2,20$
        jsr    ReadFastSerial
        bra    30$

20$     jsr    SendFastSerial
30$     lda    cmdDat2
        and    #1
        beq    40$
```

```
              tay
              LoadW  zBufPtr,#tbStatus
              jsr    ReadFastSerial
       40$    CmpB   tbStatus,#2
              blt    80$    ;-> Not an error
              add    #30                    ; Add base to DOS Error#
clda   80$,           #0
              tax
              stx    tbStatus
              rts

       cmdDat1:      .block 1               ;                                    987E
       cmdDat2:      .block 1               ;                                    987F
       cmdAddr:      .block 2               ;                                    9880
       cmdTS: .block 2
       SendFastSerial:                      ;                                    9884
              jsr    InitSerial
              jsr    Wait_DiskRdy
       SendLoop.
              dey
              sty    zSerialData
              MoveB  zLowPort,cia2pra
              lda    (zBufPtr),y
              tay
              and    #%1111
              tax
              tya
              lsr    a
              lsr    a
              lsr    a
              lsr    a
              tay
              sec
       20$    lda    rasreg
              sbc    grcntrl1
              and    #%0111
              cmp    #6
              bge    20$
              MoveB  zRcsSend00,cia2pra
              lda    SendTab,x
              ora    zLowPort
              sta    cia2pra
              lda    Send2Tab,x
              ora    zLowPort
              sta    cia2pra
              lda    SendTab,y
              ora    zLowPort
              sta    cia2pra
              lda    Send2Tab,y
              ora    zLowPort
              sta    cia2pra
              ldy    zSerialData
              bne    SendLoop
              jmp    EndSerial
       _ReadLink:
              lda    #2
clda   _ReadBlock,    #0
              sta    blkSize
              jsr_a  ValTrack,r1
              bne    99$
              CmpB   r1H,#40
```

```
               bcs     98$
               ldx     #2                      ;WriteSector
               bbeq    blkSize,10$
               inx                             ;ReadSector
        10$    jsr     CallBlkCmd
               bbne    blkSize,20$
               jsr     FixHdr
        20$    ldx     tbStatus
        99$    rts

        98$    ldx     #INV_TRACKS
               rts
        FixHdr:                                ;                                       9906
               CmpB    r4H,#]curDirHead
               bne     90$
               ldy     #25
        10$    ldx     curDirHead+4,y
               MoveB   curDirHead+OFF_DISK_NAME,y,curDirHead+4,y
               txa
               sta     curDirHead+OFF_DISK_NAME,y
               dey
               bpl     10$
        90$    rts
        _WriteBlock:
               jsr     _VerWrtBlck
               bne     99$
               stx     blkSize
               CmpB    r1H,#40
               bge     98$     ;-> Invalid Sector
               jsr     FixHdr
               ldx     #1
               jsr     CallBlkCmd
               jsr     FixHdr
               ldx     tbStatus
        99$    rts

        98$    ldx     #INV_TRACKS
               rts
;---------------------------------------------------------------------
; 1581 Turbo Code
;---------------------------------------------------------------------
;c81.turbo.s
.include c81.inc            ;0300-              ;21
BIT1           = $02
BIT2           = $04
BIT3           = $08
BIT4           = $10
BIT5           = $20
BIT6           = $40
BIT7           = $80
MASKBIT6       = %10111111

CLOCKIN        = $04
ATNRESP        = $10
DATAOUT        = $02
WPROTECT       = 8

;Jobs
JOB4           = $04
SEEKHD_DV      = $B0  ;Read first header block
READ_DV        = $80  ;Read buffer(X)
```

```
WRTSD_DV      = $90  ;Write buffer(X)

Dummy  == $1000      ; Value of URoutine prior to first use. Inluded here to be Bit Perfect...
wpstatus      == $01FA
buffer4       == $0700
ciapa         == $4000
ciapb         == $4001
ciaddra       == $4002
TrackCache    == $0C00
CONTROLLER    == $FF54
JTRKWRT       == $FF6C

.zsect $02
       zJ0Status:    .block 1
       zJ1Status:    .block 1
       zJ2Status:    .block 1
       zJ3Status:    .block 1
       zJ4Status:    .block 1      ; Our job code for all read/write transactions with disk
       zJ5Status:    .block 1
       zJ6Status:    .block 1
       zJ7Status:    .block 1      ;For BAM Jobs only
       zJ8Status:    .block 1      ;For BAM Jobs only
       zHdrs3:       .block 2      ;Job 0 TS
                     .block 6      ;Job 1-3 TS
       zJ4Track:     .block 1      ;Job 4 TS
       zJ4Sector:    .block 1
                     .block 6      ;Job 6-8 TS
                     .block 2      ;Disk ID
.zsect $25
       zFloppyChg:   .block 1      ;Flag set when disk change detected

;41 byte used by GEOS 2.0 turbo so would be safe to use here
;7E word used by GEOS 2.0 turbo so would be safe to use here

.zsect $8B
       zCache:       .block 2      ; pointer to TrackCache

.zsect $FB
       zRdYNdx:      .block 1      ; tmp storage for y register
       zBufPtr:      .block 2      ; pointer to buffer4
       zYNdx: .block 1             ; tmp storage for y register

.psect $300
       TurboCode:@                 ; Reference handle for loader in driver
       TInitDisk:    jmp InitDisk
       TExitTurbo:   jmp _TExitTurbo
       TWriteSector: jmp _WriteSector
       TReadSector:  jmp _ReadSector
       TReadLnk:     jmp _ReadLnk
       TGetStatus:   jmp _ReadStatus
              nop
              nop
              rts
              nop
              nop
              rts
              nop
              nop
              rts
       TLogDiskIn:   jmp _LogDiskIn
       drvSendTab:
```

```
        .byte   $1F,$17,$1D,$15,$1B,$13,$19,$11
        .byte   $1E,$16,$1C,$14,$1A,$12,$18,$10
drvReadTab:
        .byte   $00,$08,$02,$0A,$04,$0C,$06,$0E
        .byte   $01,$09,$03,$0B,$05,$0D,$07,$0F
drvRead2Tab:
        .byte   $00,$80,$20,$A0,$40,$C0,$60,$E0
        .byte   $10,$90,$30,$B0,$50,$D0,$70,$F0
SendFastSerial:
        jsr     WaitOnHostRdy
SendLoop:
        dey
        sty     zYNdx
        lda     (zBufPtr),y
SendOneFast:
        tay
        and     #%1111
        tax
        LoadB   ciapb,#ATNRESP
        lda     #CLOCKIN
10$     bit     ciapb
        beq     10$
        MoveB   drvSendTab,x, ciapb
        nop
        nop
        nop
        asl     a
        and     #%1111
        ora     #ATNRESP
        sta     ciapb
        tya
        lsr     a
        lsr     a
        lsr     a
        lsr     a
        tay
        nop
        MoveB   drvSendTab,y,ciapb
        jsr     Wait6us
        asl     a
        and     #%1111
        ora     #ATNRESP
        sta     ciapb
        ldy     zYNdx
        bne     SendLoop
        jsr     Wait6us
        jsr     SATNResp
        jmp     EndSerial
RSerialBlock:
        ldy     #0
ReadFastSerial:
        jsr     WaitOnHostRdy
ReadLoop:
        sty     zRdYNdx
        jsr     Wait6us
        lda     #CLOCKIN
10$     bit     ciapb
        beq     10$
        jsr     Wait16us
        lda     ciapb
        asl     a
```

```
        jsr     Wait10us
        ora     ciapb
        and     #%1111
        tax
        jsr     Wait8us
        lda     ciapb
        asl     a
        jsr     Wait10us
        ora     ciapb
        and     #%1111
        tay
        lda     drvReadTab,x
        ora     drvRead2Tab,y
        ldy     zRdYNdx
        dey
        sta     (zBufPtr),y
        bne     ReadLoop
EndSerial:
        lda     #CLOCKIN
10$     bit     ciapb
        beq     10$
        jmp     SATNDatOut
Wait16us:
        nop
        nop
        nop
Wait10us:
        nop
Wait8us:
        nop
Wait6us:
        rts
_TExitTurbo:
        bbeq    cacheFlag,10$
        jsr     DoFlushCache
10$     jsr     LED_Off
        jsr     WaitOnHostRdy
        pla
        pla
        cli
        rts
InitDisk:
        sei
        LoadB   cacheFlag,#0
        ldx     #2
        ldy     #0
StillWaiting:
        dey
        bne     StillWaiting
        dex
        bne     StillWaiting
10$     jsr     BClockIn
        beq     10$
        jsr     SATNDatOut
Loop.
        jsr     BClockIn
        beq     20$
        jsr     FlushCacheIfDirty
        bcs     20$
        jsr     LED_Off
        cli
```

```
        jsr     WClockIn
20$     sei
        LoadW   zBufPtr,#routineAddr+1
        ldy     #4
        jsr     ReadFastSerial
        jsr     LED_On
        jsr     URoutine
        jmp     Loop
WClockIn:
        jsr     BClockIn
        bne     WClockIn
        rts
SATNDatOut:
        LoadB   ciapb,#ATNRESP | DATAOUT
        rts
WaitOnHostRdy:
        jsr     WClockIn
SATNResp:
        LoadB   ciapb,#ATNRESP
        rts
BClockIn:
        lda     #CLOCKIN
        bit     ciapb
        rts
URoutine:
routineAddr:
        jmp     Dummy

blkTrack:       .block 1
blkSector:      .block 1
FlushCacheIfDirty:
        bbeq    cacheFlag,80$
        ldx     #0
        ldy     #0
10$     jsr     BClockIn
        bne     20$
        sec
        rts

20$     dey
        bne     10$
        dex
        bne     10$
DoFlushCache.
        ldx     #JOB4
        jsr     JTRKWRT
        LoadB   cacheFlag,#0
80$     clc
        rts

cacheFlag:
        .block 1
LED_Off:
        sec
        jsr     Setddra
        lda     ciapa
        and     #MASKBIT6
        sta     ciapa
        rts

LED_On:
```

```
              sec
              jsr     Setddra
              lda     ciapa
              ora     #BIT6
              sta     ciapa
              clc
       Setddra:
              lda     ciaddra
              and     #MASKBIT6
              bcc     10$
              ora     #BIT6
       10$    sta     ciaddra
              rts
       _LogDiskIn:
              bbeq    zFloppyChg,80$
              jsr_a   RunJob,#SEEKHD_DV      ;Read first header block
              cmp     #2
              bge     10$                    ;-> error occured
              lda     #0
clda   10$,           #1
              sta     zFloppyChg
              lda     #3
       80$    sta     zJ4Status
              jmp     _ReadStatus
       _ReadSector:
              lda     #0
clda   _ReadLnk,      #2
              pha
              jsr     ReadBlk
              PopY
              jsr     SendBuff
       _ReadStatus:
              MoveB   zJ4Status,buffer4
              ldy     #1
              jmp     SendBuff
       _WriteSector:
              jsr     SBufPtr
              jsr     RSerialBlock
              jsr     WriteBlk
              jmp     _ReadStatus
       WriteBlk:
              LoadB   wpstatus,#0
              bbsf    6,ciapb,10$            ;-> Write protect not active
              LoadB   wpstatus,#WPROTECT
              sta     zJ4Status
              rts

       10$    LoadB   cacheFlag,#WRTSD_DV    ;Write Buffer
clda   ReadBlk,       #READ_DV               ;Read Buffer
              ldx     blkTrack
              stx     zJ4Track
              cpx     lastTrack
              beq     20$
              stx     lastTrack
              ldx     #0
              stx     cacheFlag
       20$    ldx     blkSector
              stx     zJ4Sector
       RunJob:
              ldx     #JOB4
              jmp     CONTROLLER
```

```
       SBufPtr:
              LoadW  zBufPtr,#buffer4
              rts

       SendBuff:
              jsr    SBufPtr
              jmp    SendFastSerial

       lastTrack:    .block 1
TurboEnd:@c81.con
.include geo.con
.include geo.azp.con


RELEASE=5
CMD_TYPE      = NULL
GW=1

.if 1
C128=TRUE
.include ge8.con
.include ge8.vdc.con
ADD1_B=$20

.else

C128=FALSE
DOUBLE_W=NULL
DOUBLE_B=NULL
ADD1_W=NULL
ADD1_B=NULL

.endif

DISK_BASE=$9000

HSPACE=$A0
VMAJ='1'
VMIN='1'

CMDCH=#%11110000 | 15       ;secondary address of command channel
;c81.sym
.include geo.sym
.include geo.cia.sym
.include ge8.sym
.include geo.azp.sym
.include cbm.ser.sym


.psect

c81.mac
       .include geo.mac
.macro CmpIW immed,source
       lda    #](immed)
       cmp    source+1
       bne    z
       lda    #[(immed)
       cmp    source
z:
.endm
```

```
.macro jsr_xa target,value
       ldx    #]value
       lda    #[value
       jsr    target
.endm
.macro AddAB augend
       clc
       adc    augend
       sta    augend
.endm
.macro AddABS augend,sum
       clc
       adc    augend
       sta    sum
.endm
.macro SubBAS subtrahend,difference
       sec
       sbc    subtrahend
       sta    difference
.endm
.macro AddCBS addend,augend,sum
       lda    augend
       adc    addend
       sta    sum
.endm
.macro jsr_ax procedure, param1, param2
       lda    param1
       ldx    param2
       jsr    procedure
.endm

.macro PushWI value
       PushB  #](value)
       PushB  #[(value)
.endm

.macro EorB target,mask
       lda    target
       eor    mask
       sta    target
.endm
.macro EorB2 target,mask
       lda    mask
       eor    target
       sta    target
.endm
.macro AndB target,mask
       lda    target
       and    mask
       sta    target
.endm

.macro OraB target,mask
       lda    target
       ora    mask
       sta    target
.endm


.macro StxaW addr
       stx    addr+1
```

```
       sta    addr
.endm

.macro LxadW value
       ldx    #]value
       lda    #[value
.endm

.macro LyadW value
       ldy    #]value
       lda    #[value
.endm

.macro csec Label
       .byte  $B0    ;bcs
Label:
       sec
.endm

.macro cclc Label
       .byte  $90    ;bcc
Label:
       clc
.endm

.macro cSEC Label
       .byte  $24    ;bcs
Label:
       sec
.endm

.macro cCLC Label
       .byte  $24    ;bcc
Label:
       clc
.endm

.macro rmbf2 bitNumber,dest
       lda    dest
       and    #[~(1 << bitNumber)
       sta    dest
.endm

.macro smbf2 bitNumber,dest
       lda    dest
       ora    #(1 << bitNumber)
       sta    dest
.endm

.macro LdWW2 dest,dest2,value
       lda    #](value)
       sta    dest+1
       sta    dest2+1
       lda    #[(value)
       sta    dest
       sta    dest2
.endm
.macro AddVW2 value,augend
       clc
       lda    augend
       adc    #[(value)
```

```
       sta    augend
.if    (value >= 0) && (value <= 255)
       bcc    z
       inc    augend+1
z:
.else
       lda    augend+1
       adc    #](value)
       sta    augend+1
.endif
.endm

.macro SubBWS2 subtrahend,minuend,difference
       lda    minuend
       sec
       sbc    subtrahend
       sta    difference
       lda    minuend +1
       sbc    #0
       sta    difference+1
.endmc81.Inc
.if Pass1
       .noeqin
       .include c81.con
       .include c81.mac
       .eqin
       .include c81.sym
.endif

.noeqin
.psectc81.inc
.if Pass1
       .noeqin
       .noglbl
       .include c81.con
       .include c81.mac
       .include c81.sym
       .glbl
.endif

.noeqin
.psect
```