


```

    Ld    !r3,*.LOWBYTE. ClnormStat
    Call  Bank_Call

    Ld    Wrk_Io+$A,*D_Scan_Response
    Call  Rck_Read

Scan:    Tm    Siftst_Result,*No_SprTbl
        Jr    Z,Scan_1

        Call  Abort

Scan_1:  Ld    !r0,*0
        Ld    !r1,*7
        Ld    !r2,*Wrk_Sys+$09 ;init PBlock and Track-Head-Sector
Scan_Init: Ld    @!r2,!r0
        Inc  !r2
        Djnz !r1,Scan_Init

Scan_Lp:  Ld    Seek_Type,*Access
        Call  New_Seek

Scan_Read: Ld    Sector,!rF
        Ld    Wrk_Sys2+$09,!r9 ;save some regs
        Ld    Wrk_Sys2+$0A,!rA ;save some regs
        Ld    Wrk_Sys2+$0B,!rB ;save some regs
        Ld    !r2,*.HIBYTE. Read_Common
        Ld    !r3,*.LOWBYTE. Read_Common
        Call  Bank_Call
        Ld    !r9,Wrk_Sys2+$09 ;save some regs
        Ld    !rA,Wrk_Sys2+$0A ;save some regs
        Ld    !rB,Wrk_Sys2+$0B ;save some regs
        Jr    Nz,Scan_More

        And  !r0,$$FF-RdError ;check for non-fatal error
        Jr    Z,Scan_More

        Push !r0 ;save error code
        Call  Ext_Push

        Ld    !r1,*0 ;convert to logical interleave
        Ld    !r2,*.HIBYTE. Map_Table
        Ld    !r3,*.LOWBYTE. Map_Table
        Ld    !r4,*NbrSctrs
Scan_Map_Lp: Lde  !r0,@!r2
        Cp    !r0,!rF
        Jr    Z,Scan_Map_End
        Incw !!r2
        Inc  !r1
        Djnz !r4,Scan_Map_Lp

        Call  Abort

Scan_Map_End: Add  !rB,!r1 ;add in the sector offset
        Adc  !rA,*0
        Adc  !r9,*0

        Ld    !r0,!r9 ;check for logical block zero
        Or   !r0,!rA
        Or   !r0,!rB
        Jr   Z,Scan_L_Err

```

; Convert the Physical block number to a Logical number

```

.DO      W_10MB
Or       !rB,!rB ;check for spareblock
Jr       Z,Scan_M_Pop
Sub      !rB,!rA
Sbc      !rA,#0
Sbc      !r9,#0
.FIN

```

```

.DO      W_20MB
Ld       !r0,!rA ;check for spare block
And      !r0,#$01
Or       !r0,!rB
Jr       Z,Scan_M_Pop
Ld       !r1,!r9
Ld       !r2,!rA
Rcf
Rrc      !r1
Rrc      !r2
Sub      !rB,!r2
Sbc      !rA,!r1
Sbc      !r9,#0
.FIN

```

```

.P_To_L:
.DO      W_40MB
Ld       !r0,!rA ;check for spare block
And      !r0,#$01
Or       !r0,!rB
Jr       Z,Scan_M_Pop
Ld       !r1,!r9
Ld       !r2,!rA
Ld       !r0,#2
Rcf
Rrc      !r1
Rrc      !r2
Djnz    !r0,P_To_L
Sub      !rB,!r2
Sbc      !rA,!r1
Sbc      !r9,#0
.FIN

```

```

Scan_L_Err:
Ld       !r2,*.HIBYTE. LogicalBlock
Ld       !r3,*.LOWBYTE. LogicalBlock
Ld       !r1,#4
Ld       !r0,*Wrk_Sys+$09
Scan_LB_Lp:
Ldei    @!r2,@r0
Djnz    !r1,Scan_LB_Lp

```

```

Ld       Data_Type,*User_Type
Call    Load_Cache
Call    Load_Logical

```

```

Ld       !r2,*.HIBYTE. SC_Vector
Ld       !r3,*.LOWBYTE. SC_Vector
Call    Bank_Call

```

```

Pop      !r0 ;retrieve error code
Tm      BkStat,*B_Block+S_Block ;check for bad block
Jr       Nz,Scan_M_Pop

```

```

Scan_Rd_Err:
Ld       !r2,*.HIBYTE. Data_Ex_Handler
Ld       !r3,*.LOWBYTE. Data_Ex_Handler
Call    Bank_Call

```



```

Ld      !r3,#.LOWBYTE. Siftst_Table
Add     !r3,Free_Siftst
Acl     !r2,#0
Inc     Free_Siftst ;make value odd
Ldc     !rE,@!!r2
Incw    !!r2
Ldc     !rF,@!!r2
Jp      @!rE

```

```

Siftst_Table: .DW      Free_Ram
              .DW      Chk_Pk_Jp
              .DW      Free_Eprom
              .DW      Chk_Pk_Jp      ;nop
              .DW      Free_MtrSpd
              .DW      Chk_Pk_Jp      ;"
              .DW      Free_SctrCnt
              .DW      Chk_Pk_Jp      ;"
              .DW      Chk_Pk_Jp      ;"
              .DW      Free_Rw

```

```

Free_Ram:    Ld      !r2,#.HIBYTE. Chk_SprChk
              Ld      !r3,#.LOWBYTE. Chk_SprChk
              Call    Bank_Call
              Jr      Nz,Chk_Pk_Jp

              Or      Siftst_Result,#Ram_Fail
              Jr      Nz,Chk_Pk_Jp

```

```

Free_Eprom:  Ld      !r0,#Eprom2
              Call    EpromTest
              Jr      Z,Chk_Pk_Jp

              Or      Siftst_Result,#Eprom_Fail
              Jr      Chk_Pk_Jp

```

```

Free_SctrCnt: Ld      !r2,#.HIBYTE. SctrCount
              Ld      !r3,#.LOWBYTE. SctrCount
              Call    Bank_Call
              Jr      Z,Chk_Pk_Jp

              Or      Siftst_Result,#Sector_Cnt
              Jr      Chk_Pk_Jp

```

```

Free_MtrSpd: Ld      !r2,#.HIBYTE. MtrSpd
              Ld      !r3,#.LOWBYTE. MtrSpd
              Call    Bank_Call
              Jr      Z,Chk_Pk_Jp

              Or      Siftst_Result,#Disk_Speed
              Jr      Chk_Pk_Jp

```

```

Free_Rw:     Or      Siftst_Result,#Rw_Fail ;assume failure
              Ld      !r2,#.HIBYTE. RwTest1
              Ld      !r3,#.LOWBYTE. RwTest1
              Call    Bank_Call
              Jr      Z,Chk_Pk_Jp

              And     Siftst_Result,#$FF-Rw_Fail

              Clr     Free_Siftst
              Jr      Strt_FreeProcess ;check for arm sweep

```

.LSTOFF


```
Ld    RdErrCnt,*CrcStat
Jr    RdH_ChkEcc
```

```
RdH_End:  Ld    !r0,!r9 ;send status back to caller
          Ld    RdStat,!r9
```

```
Tcm    !r0,*RdHError ;set zero flag if error
Jp     Bank_Ret
```

```
.LSTOFF
```