


```

        .Page
        .FIN

Pro_Write:
        Ld      Wrk_Lo+$A, #Wr_Response
        Call   Pro_Wr_SetUp

        Tm      BlkStat, #B_Block ; IF BadBlock
        Jr      Z, ProWr_1

Pro_Wr_BB:
        Call   Wr_BBBlock
        Jr      Wr_Leave

ProWr_1:
        Call   Wr_Common
Pro_Wr_Exit:
        Jr      Nz, Wr_Leave

        Call   Data_Ex_Handler

Wr_Leave:
        Jp      Rd_Leave

        .Page

Pro_Wr_SetUp:
        Call   ClnNormStat

        Call   Get_Wr_Data

        Ld      !r0, #Pro_Log_Offset
        Call   Ld_LgclBlk ; LogicalBlockNumber := ...

        Ld      !r8, #Widget
        Ld      !r2, #.HIBYTE. Get_Type
        Ld      !r3, #.LOWBYTE. Get_Type
        Call   Bank_Call ; check block bounds

        Ld      Seek_Type, #Access_Offset
        Ld      Data_Type, #User_Type
        Or      DiskStat, #Wr_Op

        Ld      !r2, #.HIBYTE. OverLap
        Ld      !r3, #.LOWBYTE. OverLap
        Call   Bank_Call

        Tm      DiskStat, #Offset_On
        Jr      Nz, P_WLS_End

        Call   ReSeek

P_WLS_End:
        Ret

;*****

Wr_BBBlock:
        Ld      !r2, #.HIBYTE. WrBuf_To_Buf2
        Ld      !r3, #.LOWBYTE. WrBuf_To_Buf2
        Call   Bank_Call

        Ld      !r0, #Error+Ex_SprBlock
        Call   Data_Ex_Handler
        Ret

        .LSTOFF
        .DO      External
        .LSTON
        .Page

```



```

    Ld    !r2,*.HIBYTE. Chk_Password
    Ld    !r3,*.LOWBYTE. Chk_Password
    Call  Bank_Call
    Jr    Nz,Wr_Cmn_OK

    Ld    !r0,#0 ;byte zero
    Ld    !r1,#WrBuf_OR ;buffer over run
    Call  SetStatus
    Call  Abort

Wr_Cmn_OK:
    Push  !r4

    Or    DiskStat,#Wr_Op
    Clr   !r4
    Call  WriteBlock
    Jr    Nz,Wr_Cmn_End

    Ld    !r2,*.HIBYTE. WrBuf_To_Buf2
    Ld    !r3,*.LOWBYTE. WrBuf_To_Buf2
    Call  Bank_Call

    Ld    !r4,#Error ;ErrorCode := Error, Undetermined

    Tm    Excpt_Stat,#Recovery ;IF Recovery THEN ...
    Jr    Z,Wr_Cmn_End

    Ld    !r5,#4 ;Retry := 4

Wr_Cmn_Lp:
    Ld    !r2,*.HIBYTE. SrvoRcvry
    Ld    !r3,*.LOWBYTE. SrvoRcvry
    Call  Bank_Call

    Ld    !r2,*.HIBYTE. Buf2_To_WrBuf
    Ld    !r3,*.LOWBYTE. Buf2_To_WrBuf
    Call  Bank_Call

    Call  WriteBlock
    Clr   !r4
    Jr    Nz,Wr_Cmn_End

    Djnz  !r5,Wr_Cmn_Lp
    Ld    !r4,#Error

Wr_Cmn_Servo:
    Tm    WrStat,#WrSrvoErr
    Jr    Z,Wr_Cmn_Hdr

    Clr   !r0 ;Byte 0
    Ld    !r1,#Stat_Srvo
    Call  SetStatus
    Call  Abort

Wr_Cmn_Hdr:
    Tm    WrStat,#WrNoHdrFnd
    Jr    Z,Wr_Cmn_End

    Ld    !r4,#Error+Ex_HdrSpr

Wr_Cmn_End:
    Ld    !r0,!r4
    Pop   !r4 ;retrieve register from stack
    Tcm   !r0,#WrError
    Jp    Bank_Ret

.LSTOFF
.DO    External

```



```
Ld    !r2,*Error+Ex_SprBlock
Jr    Nz,WrVer_Ret
```

```
WrU_Chg_Ex:  Ld    !r0,!r2
```

```
WrVer_Ret:  Push  !r0
            Call  ExtPop_Vector
            Pop   !r0
            Tcm  !r0,*Error ;set error condition
            Jp    Bank_Ret
```

```
.LSTOFF
```