



.D0 Internal  
.LSTON  
.Page  
.FIN

```
Buf2_To_RBuf:  Ld    !r0,#.HIBYTE. Buf2Array
               Ld    !r1,#.LOWBYTE. Buf2Array
               Ld    !r2,#.HIBYTE. RDummy
               Ld    !r3,#.LOWBYTE. RDummy
               Jr    B_Move

WrBuf_To_Buf2: Ld    !r0,#.HIBYTE. ( WBuffer1-1 )
               Ld    !r1,#.LOWBYTE. ( WBuffer1-1 )
               Jr    X_To_Buf2

Buf2_To_WrBuf: Ld    !r0,#.HIBYTE. Buf2Array
               Ld    !r1,#.LOWBYTE. Buf2Array
               Ld    !r2,#.HIBYTE. ( WBuffer1-1 )
               Ld    !r3,#.LOWBYTE. ( WBuffer1-1 )
               Jr    B_Move

Spr_To_RBuf:   Ld    !r0,#.HIBYTE. SpareArray
               Ld    !r1,#.LOWBYTE. SpareArray
               Ld    !r2,#.HIBYTE. RBuffer1
               Ld    !r3,#.LOWBYTE. RBuffer1
               Jr    B_Move

RBuf_To_Spr:   Ld    !r0,#.HIBYTE. RBuffer1
               Ld    !r1,#.LOWBYTE. RBuffer1
X_To_Spr:      Ld    !r2,#.HIBYTE. SpareArray
               Ld    !r3,#.LOWBYTE. SpareArray
               Jr    B_Move

WrBuf_To_Spr:  Ld    !r0,#.HIBYTE. WBuffer1
               Ld    !r1,#.LOWBYTE. WBuffer1
               Jr    X_To_Spr

RBuf_To_Buf2: Cp    DataType,#User_Type ;nop if sparetable data
               Jr    Nz,Return_Vector
               Ld    !r0,#.HIBYTE. RDummy
               Ld    !r1,#.LOWBYTE. RDummy
X_To_Buf2:     Ld    !r2,#.HIBYTE. Buf2Array
               Ld    !r3,#.LOWBYTE. Buf2Array
B_Move:        Call  BlockMove
Return_Vector: Jp    Bank_Ret

Spr_To_WrBuf:  Ld    !r0,#.HIBYTE. SpareArray
               Ld    !r1,#.LOWBYTE. SpareArray
               Ld    !r2,#.HIBYTE. WBuffer1
               Ld    !r3,#.LOWBYTE. WBuffer1
               Jr    B_Move

Zero_RdBuf:    Ld    !r2,#.HIBYTE. ReadArray
               Ld    !r3,#.LOWBYTE. ReadArray
               Call  ZeroBlock
               Jr    Return_Vector

Zero_Fmt:     Ld    !r2,#.HIBYTE. ReadArray
               Ld    !r3,#.LOWBYTE. ReadArray
               Ld    !rE,#.HIBYTE. ( FBuffer1 - ReadArray )
               Ld    !r0,#0 ;init block to pattern
Zero_FLp:     Lde   @!!r2,!r0
               Incw  !!r2
```















.FIN

```
Get_Type:      Call    Load_Logical

               Ld     !r0,!r8
               Ld     !r8,#User_Type

               Or     !r0,!r0 ;IF ( Driver_Type = Profile )
               Jr     Nz,Get_Type_Check

               Ld     !r0,$$FF ;IF BlockNumber = $-2...
               Ld     !r1,$$FF
               Ld     !r2,$$FE
               Call   Sub3 ;compare
               Or     !r0,!r1
               Or     !r0,!r2
               Jr     Nz,G_T_ChkID

               Ld     !r8,#SprTbl_Type

G_T_ProCnvrt:  Clr    !rC
               Clr    !rD
               Ld     !rE,#1
               Jr     Get_Type_Check

G_T_ChkID:    Ld     !r0,$$FF ;IF BlockNumber = $$-1...
               Ld     !r1,$$FF
               Ld     !r2,$$FF
               Call   Sub3 ;compare
               Or     !r0,!r1
               Or     !r0,!r2
               Jr     Nz,Get_Type_Check
               Ld     !r8,#ID_Type
               Jr     G_T_ProCnvrt

Get_Type_Check: Ld     !r0,#HiMaxLogical ;IF ( BlockNumber > MaxLogical )...
               Ld     !r1,#MidMaxLogical
               Ld     !r2,#LoMaxLogical
               Call   Sub3 ;compare
               Jr     Uge,Get_Type_End

               Ld     !r0,#2 ;byte 2
               Ld     !r1,#Illegal_Block
               Call   SetStatus

               Call   Abort

Get_Type_End: Jp     Bank_Ret
```

.LSTOFF