



```

Start_Command:
    Srp    #Wrk_Sys ;get into a reasonable state
    Clr    Sph
    Ld     Spl,*Stack_Top

    Ld     !r2,#.HIBYTE. StackPtr ;init external stack
    Ld     !r3,#.LOWBYTE. StackPtr
    Ld     !r0,#.HIBYTE. TopOfStack
    Lde    @!!r2,!r0
    Incw   !!r2
    Ld     !r0,#.LOWBYTE. TopOfStack
    Lde    @!!r2,!r0

    Cp     Cache_Index,#CacheLength
    Jr     Lt,St_Load_Cmdnd

    Clr    Cache_Index

St_Load_Cmdnd:
    Ld     !r2,#.HIBYTE. Cmdnd_Ptr
    Ld     !r3,#.LOWBYTE. Cmdnd_Ptr
    Ld     !rE,#.HIBYTE. CStatus4
    Ld     !rF,#.LOWBYTE. CStatus4
    Ld     !r1,#8 ;move 8 bytes

St_L_Lp:
    Lde    !r0,@!!r2
    Lde    @!!rE,!r0
    Incw   !!r2
    Incw   !!rE
    Djnz   !r1,St_L_Lp

Strt_ZH:
    Ld     !rE,#.HIBYTE. CStatus4
    Ld     !rF,#.LOWBYTE. CStatus4
    Lde    !r4,@!!rE

    Ld     !r6,!r4
    And    !r6,#CmdndType
    Cp     !r6,#$F0 ;check for 'free bus command'
    Jr     Nz,Strt_ChkDiag

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

Strt_ChkDiag:
    Cp     !r6,#$10 ;check for diagnostic command
    Jr     Nz,Strt_Swap

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

Strt_Swap:
    Ld     Seek_Type,#Access_Offset
    Swap   !r6 ;r6 := CommandType
    Jr     Z,Chk_Inst ;jump if no check byte in string

    Ld     !r8,!r4
    And    !r8,#$0F ;get length of command string
    Incw   !!rE ;get opcode
    Lde    !r4,@!!rE
    Ld     !rE,#.HIBYTE. Cmdnd_Ptr
    Ld     !rF,#.LOWBYTE. Cmdnd_Ptr
    Call   Chk_Chk_Byte
    Jr     Z,Chk_Inst

    Call   Abort

```











```
Rd_Cmn_RdErr:  Ld    !r4, #Error+Ex_ReadErr
                Jr    Rd_Cmn_End

Rd_BadBlock:   Ld    !r4, #Error+Ex_BadBlock

Rd_Cmn_End:    Ld    !r0, !r4
                Tcm  !r0, #RdError
                Jp   Bank_Ret

                .LSTOFF
```