Table of contents

```
                                        .TITLE  TSTIO   -- TSX-Plus terminal interrupt driver
000000                                  .PSECT  TSTIO
                                        .ENABL  LC
                                        .ENABL  AMA
                                ;
                                ;  TSTIO contain the interrupt processing code for sending and receiving
                                ;  characters to DL11, DZ11, and DH11 lines.
                                ;
                                        .DSABL  GBL
                                ;
                                ;  Global definitions
                                ;
                                        .GLOBL  TSTIO,DLINT,DZOINT,CDSTRT,CDSTOP,NEDSOT
                                        .GLOBL  CDIRTN,CDIFLG,CDORTN,CDOFLG,NEDCDI,NEDCDO
                                        .GLOBL  ININT,CDSSPD,LINRTS,TTRSAV,TTINPT
                                        .GLOBL  CDCLOK,CDGDSS,CDSDSS,CDSBRK,SETSPD
                                        .GLOBL  CDSTR2,CDSXON,CDSXOF,DLSTRT,DZSTRT
                                        .GLOBL  SILFET,TRNSTR,NEDCLO,TIOVEC,GETDSS,SETDSS
                                ;
                                ;  Global references
                                ;
                                        .GLOBL  $OITIM,$XCHAR
                                        .GLOBL  KPAR6,LMXLN
                                        .GLOBL  DHSSPD,VHSSPD,DHSBRK,VHSBRK
                                        .GLOBL  LMXNUM
                                        .GLOBL  LSW3,LSW5
                                        .GLOBL  MXCSR,MXLNT
                                        .GLOBL  MXTBUF,MXTCR,NEDCHR,PR7,PSW
                                        .GLOBL  RLINE
                                        .GLOBL  TBR,TRINT,TRRDY,TSR
                                        .GLOBL  DHSTRT,DHSTOP,DLSBRK,DZSBRK,DZSSPD
                                        .GLOBL  VHSTRT,DLCLOK,DLSSPD,$DHCDO
                                        .GLOBL  DHXON,DHXOFF,VHXON,VHXOFF
                                        .GLOBL  DZCLOK,DLGDSS,DZGDSS,DHGDSS,VHGDSS
                                        .GLOBL  DLSDSS,DZSDSS,DHSDSS,VHSDSS
                                        .GLOBL  LOUTIR,KPAR5
                                        .GLOBL  LSW10
                                        .GLOBL  $HISTP
                                        .GLOBL  $SXOFF,LCDTYP
                                        .GLOBL  $SXON
                                        .GLOBL  VHSTOP,INTPRI
                                        .GLOBL  DHCLOK,VHCLOK
```

```
                        ;+-----------------------------------------------------------------------
                        ;  Macro definitions:
                        ;
                        ;  Disable interrupts
                        ;
                              .MACRO  DISABL          ;Disable interrupts
                              BIS     #340,@#PSW
                              .ENDM   DISABL
                        ;
                        ;  Enable interrupts
                        ;
                              .MACRO  ENABL           ;Enable interrupts
                              BIC     INTPRI,@#PSW
                              .ENDM   ENABL
                        ;
                        ;  Call a routine in a system overlay
                        ;
                              .MACRO  OCALL   ENTADD
                              .IF     B,ENTADD
                                      .ERROR  ;OCALL without entry address
                              .ENDC
                              CALL    OVRHC
                              .WORD   ENTADD
                              .ENDM   OCALL
                        ;
                        ;  The TTMAP and TTMAPX macros are used to map kernel-mode par6 to the
                        ;  terminal character buffer area.  The previous contents of par6 map
                        ;  register are pushed on the stack and may be restored by using the
                        ;  UNMAP or UNMAPX macros.
                        ;  R1 must contain the line index number of the line whose buffers
                        ;  are being accessed.
                        ;  The difference between the TTMAP-UNMAP macros and the TTMAPX-UNMAPX
                        ;  macros is that the X-versions are more efficient but may only be
                        ;  used from within interrupt service routines where we are guaranteed
                        ;  to be running on the system stack.
                        ;  The TTMAP and UNMAP versions of the macros must only be
                        ;  used in sections of code where the interrupts are disabled.
                        ;
                              .MACRO  TTMAPX
                              MOV     LTTPAR(R1),@#KPAR6
                              .ENDM   TTMAPX

                        ;
                        ;

                              .MACRO  UNMAPX
                              .ENDM   UNMAPX
                        ;
                              .MACRO  TTMAP
                              MOV     @#KPAR6,MAPHLD
                              MOV     LTTPAR(R1),@#KPAR6
                              .ENDM   TTMAP
                        ;
                              .MACRO  UNMAP
                              MOV     MAPHLD,@#KPAR6
                              .ENDM   UNMAP
```

```
                                ;-----------------------------------------------------------------------
                                ;  Address vectors used to transfer control to device specific routines
                                ;  based on the device type code (CDX$xxx).
                                ;  Note: the entries in these vectors must be arranged in the same order
                                ;  as the CDX$xxx values defined in TSGEN.
                                ;
                                ;  Table of routines to call to initiate transmission to a line
                                ;
                                ;  Inputs:
                                ;   R1 = Physical line index number.
                                ;
000000                          TSTIO:
000000  000456'                CDSTRT: .WORD   DLSTRT           ;DL11
000002  000600'                        .WORD   DZSTRT           ;DZ11
000004  000640'                        .WORD   REQCDO           ;DH11
000006  000640'                        .WORD   REQCDO           ;DHV11
000010  000240'                        .WORD   DORTS            ;Professional console
000012  000240'                        .WORD   DORTS            ;Professional communications port
000014  000240'                        .WORD   DORTS            ;Professional printer port
000016  000240'                        .WORD   DORTS            ;Professional quad serial line unit
                                ;
                                ;  Table of routines used to perform actual hardware startup
                                ;  of output transmission
                                ;
000020  000240'                CDSTR2: .WORD   DORTS            ;DL11
000022  000240'                        .WORD   DORTS            ;DZ11
000024  000000G                        .WORD   DHSTRT           ;DH11
000026  000000G                        .WORD   VHSTRT           ;DHV11
000030  000240'                        .WORD   DORTS            ;Professional console
000032  000240'                        .WORD   DORTS            ;Professional communications port
000034  000240'                        .WORD   DORTS            ;Professional printer port
000036  000240'                        .WORD   DORTS            ;Professional quad serial line unit
                                ;
                                ;  Table of routines to call to stop transmission to a line.
                                ;  (These routines are used if a ctrl-S is received)
                                ;
                                ;  Inputs:
                                ;   R1 = Physical line index number.
                                ;
000040  000240'                CDSTOP: .WORD   DORTS            ;DL11
000042  000240'                        .WORD   DORTS            ;DZ11
000044  000000G                        .WORD   DHSTOP           ;DH11
000046  000000G                        .WORD   VHSTOP           ;DHV11
000050  000240'                        .WORD   DORTS            ;Professional console
000052  000240'                        .WORD   DORTS            ;Professional communications port
000054  000240'                        .WORD   DORTS            ;Professional printer port
000056  000240'                        .WORD   DORTS            ;Professional quad serial line unit
                                ;
                                ;  Table of routines called to stuff an XOFF (ctrl-S) into the output
                                ;  stream for a line.
                                ;
                                ;  Inputs:
                                ;   R1 = Line index number
                                ;
000060  000314'                CDSXOF: .WORD   DLXOFF            ;DL11
000062  000314'                        .WORD   DLXOFF            ;DZ11
000064  000000G                        .WORD   DHXOFF            ;DH11
```

```
                                        .WORD   VHXOFF          ;DHV11
                                        .WORD   DLXOFF          ;Professional console
                                        .WORD   DORTS           ;Professional communications port
                                        .WORD   DLXOFF          ;Professional printer port
                                        .WORD   DLXOFF          ;Professional quad serial line unit
                                ;
                                ;  Table of routine called to stuff an XON (ctrl-Q) into the output
                                ;  stream for a line.
                                ;
                                ;  Inputs:
                                ;   R1 = Line index number.
                                ;
                                CDSXON: .WORD   DLXON           ;DL11
                                        .WORD   DLXON           ;DZ11
                                        .WORD   DHXON           ;DH11
                                        .WORD   VHXON           ;DHV11
                                        .WORD   DLXON           ;Professional console
                                        .WORD   DORTS           ;Professional communications port
                                        .WORD   DLXON           ;Professional printer port
                                        .WORD   DLXON           ;Professional quad serial line unit
                                ;
                                ;  Table of routines to be called on a 0.5 second basis to do line checking.
                                ;
                                ;  Inputs:
                                ;   R1 = Physical line index number.
                                ;
                                CDCLOK: .WORD   DLCLOK          ;DL11
                                        .WORD   DZCLOK          ;DZ11
                                        .WORD   DHCLOK          ;DH11
                                        .WORD   VHCLOK          ;DHV11
                                        .WORD   DORTS           ;Professional console
                                        .WORD   DORTS           ;Professional communications port
                                        .WORD   DORTS           ;Professional printer port
                                        .WORD   DORTS           ;Professional quad serial line unit
                                ;
                                ;  Table of routines to be called to get the data set status
                                ;
                                ;  Inputs:
                                ;   R1 = Physical line index number.
                                ;
                                ;  Outputs:
                                ;   R0 = Generic data set status flags (MS$xxx)
                                ;
                                CDGDSS: .WORD   DLGDSS          ;DL11
                                        .WORD   DZGDSS          ;DZ11
                                        .WORD   DHGDSS          ;DH11
                                        .WORD   VHGDSS          ;DHV11
                                        .WORD   DORTS           ;Professional console
                                        .WORD   DORTS           ;Professional communications port
                                        .WORD   DORTS           ;Professional printer port
                                        .WORD   DORTS           ;Professional quad serial line unit
                                ;
                                ;  Table of routines to be called to set data set status
                                ;
                                ;  Inputs:
                                ;   R0 = Data set control flags (MS$xxx)
                                ;   R1 = Physical line index number.
```

```
                                        ;
000160  000000G                 CDSDSS: .WORD   DLSDSS          ;DL11
000162  000000G                         .WORD   DZSDSS          ;DZ11
000164  000000G                         .WORD   DHSDSS          ;DH11
000166  000000G                         .WORD   VHSDSS          ;DHV11
000170  000240'                         .WORD   DORTS           ;Professional console
000172  000240'                         .WORD   DORTS           ;Professional communications port
000174  000240'                         .WORD   DORTS           ;Professional printer port
000176  000240'                         .WORD   DORTS           ;Professional quad serial line unit
                                        ;
                                        ;  Table of routines to be called to control break transmission
                                        ;
                                        ;  Inputs:
                                        ;   R0 = Break control flag (MS$BRK)
                                        ;   R1 = Physical line index number.
                                        ;
000200  000000G                 CDSBRK: .WORD   DLSBRK          ;DL11
000202  000000G                         .WORD   DZSBRK          ;DZ11
000204  000000G                         .WORD   DHSBRK          ;DH11
000206  000000G                         .WORD   VHSBRK          ;DHV11
000210  000240'                         .WORD   DORTS           ;Professional console
000212  000240'                         .WORD   DORTS           ;Professional communications port
000214  000240'                          WORD   DORTS           ;Professional printer port
000216  000240'                         .WORD   DORTS           ;Professional quad serial line unit
                                        ;
                                        ;  Table of routines to be called to control transmission speed
                                        ;
                                        ;  Inputs:
                                        ;   R0 = Line speed code.
                                        ;   R1 = Line index number.
                                        ;
000220  000000G                 CDSSPD: .WORD   DLSSPD          ;DL11
000222  000000G                         .WORD   DZSSPD          ;DZ11
000224  000000G                         .WORD   DHSSPD          ;DH11
000226  000000G                         .WORD   VHSSPD          ;DHV11
000230  000240'                         .WORD   DORTS           ;Professional console
000232  000240'                         .WORD   DORTS           ;Professional communications port
000234  000240'                         .WORD   DORTS           ;Professional printer port
000236  000240'                         .WORD   DORTS           ;Professional quad serial line unit
                                        ;
                                        ;  Dummy routine to do a simple return
                                        ;
000240  000207                  DORTS:  RETURN
                                        ;
                                        ;-----------------------------------------------------------------------
                                        ;  Pointer vector to routines in TSTIOX module that is relocated over TSINIT.
                                        ;
000242                          TIOVEC:
000242  000000                  ININTP: .WORD   0
000244  000000                  TTINPT: .WORD   0
000246  000000                  SILFET: .WORD   0
000250  000000                  TTRSAV: .WORD   0
000252  000000                  TRNSTR: .WORD   0
000254  000000                  NEDCP:  .WORD   0
000256  000000                  CDORTN: .WORD   0
000260  000000                  CDIRTN: .WORD   0
000262  000000                  SNDFRE: .WORD   0
```

```
    172 000264  177777                          .WORD   -1              ;Signal end of vector
    173                                 ;
    174                                 ;-----------------------------------------------------------------------
    175                                 ;  Data areas
    176                                 ;
    177 000266  000000                  MAPHLD: .WORD   0               ;TEMP CELL USED BY TTMAP MACRO
    178 000270  000000                  NEDCDI: .WORD   0               ;Non-zero ==> Need clock driven input proc
    179 000272  000000                  NEDCDO: .WORD   0               ;Non-zero ==> Need clock driven output proc
    180 000274  000000                  NEDCLO: .WORD   0               ;Non-zero ==> Need CL output processing
    181 000276  000000                  NEDSOT: .WORD   0               ;Non-zero ==> Need output-low job scheduling
    182                                 ;
    183                                 ;  Byte data
    184                                 ;
    185 000300      000                 CDIFLG: .BYTE   0               ;Non-zero ==> Input char fork routine running
    186 000301      000                 CDOFLG: .BYTE   0               ;Non-zero ==> Output char fork routine running
    187                                 ;  Table used to select bit-within-word corresponding to a line
    188 000302      001     002     004 MXLBIT: .BYTE   1,2,4,10,20,40,100,200
        000305      010     020     040
        000310      100     200
    189                                         .EVEN
    190                                 ;
    191                                 ;  Ignore an interrupt
    192                                 ;
    193 000312  000207                  LINRTS: RETURN                  ;Ignore interrupt
```

```
                                        .SBTTL  DLXOFF -- Transmit XOFF character
                                ;-----------------------------------------------------------------------
                                ;  DLXOFF is called to transmit an XOFF (ctrl-S) character to stop
                                ;  transmission to us.
                                ;
                                ;  Inputs:
                                ;   R1 = Line index number.
                                ;
000314                          DLXOFF:
                                ;
                                ;  Set flag saying we have stopped the sender
                                ;
000314  052761  0000000 0000000         BIS     #$HISTP,LSW10(R1);Set flag saying we have stopped sender
                                ;
                                ;  Set flag to cause an XOFF character to be sent by the transmitter
                                ;  interrupt routine.
                                ;
000322  052761  0000000 0000000         BIS     #$SXOFF,LSW10(R1);Set flag saying to send XOFF
                                ;
                                ;  Start the transmitter
                                ;
000330  016100  0000000                 MOV     LCDTYP(R1),R0   ;Get device type code index
000334  004770  000000'                 CALL    @CDSTRT(R0)     ;Call routine to start the transmitter
                                ;
                                ;  Finished
                                ;
000340  000207                          RETURN
```

```
                                        .SBTTL  DLXON  -- Transmit XON character
                                ;-----------------------------------------------------------------------
                                ;  DLXON is called to transmit an XON (ctrl-Q) character.
                                ;
                                ;  Inputs:
                                ;   R1 = Line index number.
                                ;
000342                          DLXON:
                                ;
                                ;  Clear flag that says XOFF has been sent
                                ;
000342  042761  000000G 000000G        BIC     #$HISTP,LSW10(R1);Clear flag that says XOFF has been sent
                                ;
                                ;  Set flag to cause an XON character to be sent by the transmitter
                                ;  interrupt routine.
                                ;
000350  052761  000000G 000000G        BIS     #$SXON,LSW10(R1);Force transmission of XON
                                ;
                                ;  Start the transmitter
                                ;
000356  016100  000000G                MOV     LCDTYP(R1),R0   ;Get device type index number
000362  004770  000000'                CALL    @CDSTRT(R0)     ;Call routine to start the transmitter
                                ;
                                ;  Finished
                                ;
000366  000207                         RETURN
```

```
                        .SBTTL  DLINT  -- DL terminal output interrupt entry point
                ;-----------------------------------------------------------------------
                ;  DLINT contains the DL device specific output interrupt service routine.
                ;  The interrupt is vectored directly to an instruction sequence of the
                ;  form
                ;
                ;       JSR     R4,@#DLINT
                ;       .WORD   line_number
                ;
                ;  Thus on entry to DLINT R4 has been saved on the stack and currently
                ;  points to a word containing the line index number.
                ;
                ;
000370  010046          DLINT:  MOV     R0,-(SP)                ;Save registers (R4 already on stack)
                ;
                ;  Obtain the interrupting physical line number.
                ;
000372  011404          DLINT:  MOV     (R4),R4                 ;Get physical line index number
                ;
                ;  Get next character to transmit
                ;
000374  004774  000000G         CALL    @LOUTIR(R4)             ;Get next character for line
000400  103417                  BCS     DLCLR                   ;Br if no character available
                ;
                ;  Transmit the character contained in R0.
                ;
000402  052764  000000G 000000G         BIS     #$XCHAR,LSW3(R4)        ;Set transmitter busy flag
000410  042764  000000G 000000G         BIC     #$OITIM,LSW5(R4)        ;Start timer to catch lost interrupts
000416  032774  000000G 000000G 1$:     BIT     #TRRDY,@TSR(R4)         ;Make sure transmitter is ready
000424  001774                  BEQ     1$                      ;Should never loop here
000426  110074  000000G                 MOVB    R0,@TBR(R4)             ;Transmit the character
                ;
                ;  Finished.  Return from interrupt.
                ;
000432  012600          DLRTN:  MOV     (SP)+,R0
000434  012604                  MOV     (SP)+,R4                ;R4 was pushed by JSR instruction
000436  000002                  RTI
                ;
                ;  Disable transmit interrupts.
                ;
000440  042764  000000G 000000G DLCLR:  BIC     #$XCHAR,LSW3(R4)        ;Set not busy transmit flag
000446  042774  000000G 000000G         BIC     #TRINT,@TSR(R4)         ;Disable transmit interrupt enable
000454  000766                  BR      DLRTN                   ;Return from interrupt

                        .SBTTL  DLSTRT -- Start output to a DL11 line
                ;-----------------------------------------------------------------------
                ;  Start output to a DL11 line.
                ;  Enable transmit interrupts.
                ;    R1 = Physical line number
                ;
000456  052771  000000G 000000G DLSTRT: BIS     #TRINT,@TSR(R1)         ;Enable transmit interrupt enable
000464  000207                  RETURN
```

```
                        .SBTTL  DZOINT -- DZ terminal output interrupt entry point
                ;-----------------------------------------------------------------------
                ;  DZOINT contains the DZ device specific output interrupt service routine.
                ;  The interrupt is vectored directly to an instruction sequence of the
                ;  form
                ;
                ;       JSR     R5,@#DZOINT
                ;       .WORD   mux_number
                ;
                ;  Thus on entry to DZOINT R5 has been saved on the stack and currently
                ;  points to a word containing the mux index number.
                ;
                ;
000466  010046  DZOINT: MOV     R0,-(SP)                ;Save registers (R5 already on stack)
000470  010446          MOV     R4,-(SP)
                ;
                ;  Obtain the interrupting physical line number.
                ;
000472  011505          MOV     (R5),R5                 ;Get mux index number
000474  017500  000000G MOV     @MXCSR(R5),R0           ;Get the DZ CSR status register
000500  100022          BPL     DZRTN                   ;Br if no line interrupted
000502  042700  000000C BIC     #^C<RLINE>,R0           ;Issolate the interrupting line
000506  000300          SWAB    R0                      ;Shift to right byte
000510  066500  000000G ADD     MXLNT(R5),R0            ;Point to correct MUX line table
000514  111004          MOVB    (R0),R4                 ;Get the physical line number
000516  001413          BEQ     DZRTN                   ;Br if line not genned in system
                ;
                ;  Obtain the next character for the interrupting line.
                ;
000520  004774  000000G         CALL    @LOUTIR(R4)             ;Get the next character for line
000524  103414          BCS     DZCLR                   ;Br if no character available
                ;
                ;  Transmit the character contained in R0.
                ;
000526  052764  000000G 000000G BIS     #$XCHAR,LSW3(R4)        ;Set transmitter busy flag
000534  042764  000000G 000000G BIC     #$OITIM,LSW5(R4)        ;Start timer to catch lost interrupts
000542  110075  000000G         MOVB    R0,@MXTBUF(R5)          ;Transmit the character
                ;
                ;  Finished.  Return from interrupt.
                ;
000546  012604  DZRTN:  MOV     (SP)+,R4
000550  012600          MOV     (SP)+,R0
000552  012605          MOV     (SP)+,R5                ;R5 was saved by JSR
000554  000002          RTI
                ;
                ;  Disable transmit interrupts.
                ;
000556  042764  000000G 000000G DZCLR:  BIC     #$XCHAR,LSW3(R4)        ;Set not busy transmit flag
000564  016400  000000G         MOV     LMXLN(R4),R0            ;Get number of line within MUX
000570  146075  000302' 000000G BICB    MXLBIT(R0),@MXTCR(R5)   ;Disable transmitter interrupt
000576  000763          BR      DZRTN                   ;Return from interrupt
```

```
                        .SBTTL  DZSTRT -- Start output to a DZ11 line
                ;-----------------------------------------------------------------------
                ;  Start output to a DZ11 line.
                ;  Enable transmit interrupts.
                ;    R1 = Physical line number
                ;
000600  010046  DZSTRT: MOV     R0,-(SP)                ;Save registers
000602  010546          MOV     R5,-(SP)
000604  016105  000000G MOV     LMXNUM(R1),R5           ;Get the MUX index number
000610  016100  000000G MOV     LMXLN(R1),R0            ;Get number of line within MUX
000614  156075  000302' 000000G BISB    MXLBIT(R0),@MXTCR(R5)   ;Enable transmitter interrupt
000622  012605          MOV     (SP)+,R5                ;Restore registers
000624  012600          MOV     (SP)+,R0
000626  000207          RETURN
```

```
                                        .SBTTL  NEDCHR -- Get next character for output
                                ;-----------------------------------------------------------------
                                ;  This is a dummy resident routine which calls the relocated
                                ;  NEDCHR routine in the TSTIOX module.
                                ;
000630  000177  177420          NEDCHR: JMP     @NEDCP                  ;Call routine in TSTIOX
                                ;
                                ;-----------------------------------------------------------------
                                ;  This is a dummy resident routine which calls the relocated
                                ;  ININT routine in the TSTIOX module.
                                ;
000634  000177  177402          ININT:  JMP     @ININTP                 ;Call routine in TSTIOX

                                        .SBTTL  REQCDO -- Request clock-driven output processing
                                ;-----------------------------------------------------------------
                                ;  REQCDO is called to set a flag requesting clock-driven output
                                ;  character processing.
                                ;
                                ;  Inputs:
                                ;   R1 = Physical line index number.
                                ;
000640  052761  000000G 000000G REQCDO: BIS     #$DHCDO,LSW10(R1)       ;Request clock-driven processing
000646  005237  000272'                 INC     NEDCDO                  ;Request output processing
000652  000207                          RETURN                          ;Finished
```

```
                                        .SBTTL  SETSPD -- Set transmit/receive speed for a line
                                ;-----------------------------------------------------------------------
                                ;  SETSPD is called to set the transmit/receive speed for a line.
                                ;
                                ;  Inputs:
                                ;   R0 = Speed code.
                                ;   R1 = Line index number of line being set.
                                ;
000654  010246                  SETSPD: MOV     R2,-(SP)
000656  013746  000000G                 MOV     @#KPAR5,-(SP)   ;Save kpar 5 in case an overlay is calling us
                                ;
                                ;  Call hardware dependent routine to set the speed
                                ;
000662  016102  000000G                 MOV     LCDTYP(R1),R2   ;Get hardware device type code
000666  004772  000220'                 CALL    @CDSSPD(R2)     ;Call routine to set the speed
                                ;
                                ;  Finished
                                ;
000672  012637  000000G                 MOV     (SP)+,@#KPAR5
000676  012602                         MOV     (SP)+,R2
000700  000207                         RETURN
```

```
                                        .SBTTL  GETDSS/SETDSS -- Get/set data set status
                                ;-------------------------------------------------------------------
                                ;  Inputs: R1 contains line index
                                ;          For set, options are in R0 (for @CDSDSS routine)
                                ;  Outputs: For get, options are returned in R0 (by @CDGDSS routine)
                                ;
000702  005046                 GETDSS: CLR     -(SP)           ;Say we are getting status
000704  000402                         BR      DSSCOM          ;Br to common routine
000706  012746  000001         SETDSS: MOV     #1,-(SP)        ;Say we are setting data set status
                                ;
000712  010246                 DSSCOM: MOV     R2,-(SP)
000714  013746  000000G              MOV     @#KPAR5,-(SP)   ;Save KPAR 5 in case overlay is calling us
                                ;
                                ;  Call hardware dependent routine to get current data set status
                                ;
000720  016102  000000G              MOV     LCDTYP(R1),R2   ;Get hardware device type code
000724  005766  000004               TST     4(SP)           ;Get or set data set status?
000730  001003                       BNE     1$              ;Br if set
000732  004772  000140'               CALL    @CDGDSS(R2)     ;Get data set status (into R0)
000736  000402                       BR      2$              ;Br to exit
000740  004772  000160'         1$:   CALL    @CDSDSS(R2)     ;Set data set status (from R0)
                                ;
                                ;  Clean up and return
                                ;
000744  012637  000000G         2$:   MOV     (SP)+,@#KPAR5   ;Restore KPAR 5 mapping
000750  012602                       MOV     (SP)+,R2
000752  005726                       TST     (SP)+           ;Remove GET/SET flag from stack
000754  000207                       RETURN
                                ;
000001                               .END
Errors detected:  0

*** Assembler statistics


Work  file  reads:  0
Work  file writes:  0
Size of work file:  232 Words  ( 1 Pages)
Size of core pool:  17920 Words  ( 70 Pages)
Operating  system:  RT-11

Elapsed time:  00:00:08.52
DK:TSTIO,LP:TSTIO=DK:TSTIO/C/N:SYM
```

Cross reference table (CREF V05.05)

```
$DHCDO   1-32      9-22
$HISTP   1-38      4-13      5-12
$OITIM   1-22      6-28      7-36
$SXOFF   1-39      4-18
$SXON    1-40      5-17
$XCHAR   1-22      6-27      6-41      7-35      7-48
CDCLOK   1-16      3-84#
CDGDSS   1-16      3-101#    11-19
CDIFLG   1-14      3-185#
CDIRTN   1-14      3-170#
CDOFLG   1-14      3-186#
CDORTN   1-14      3-169#
CDSBRK   1-16      3-131#
CDSDSS   1-16      3-116#    11-21
CDSSPD   1-15      3-146#    10-15
CDSTOP   1-13      3-40#
CDSTR2   1-17      3-25#
CDSTRT   1-13      3-13#      4-23      5-22
CDSXOF   1-17      3-55#
CDSXON   1-17      3-70#
DHCLOK   1-42      3-86
DHGDSS   1-34      3-103
DHSBRK   1-24      3-133
DHSDSS   1-35      3-118
DHSSPD   1-24      3-148
DHSTOP   1-31      3-42
DHSTRT   1-31      3-27
DHXOFF   1-33      3-57
DHXON    1-33      3-72
DLCLOK   1-32      3-84
DLCLR    6-23      6-41#
DLGDSS   1-34      3-101
DLINT    1-13      6-14#
DLRTN    6-35#     6-43
DLSBRK   1-31      3-131
DLSDSS   1-35      3-116
DLSSPD   1-32      3-146
DLSTRT   1-17      3-13      6-51#
DLXOFF   3-55      3-56      3-59      3-61      3-62      4-9#
DLXON    3-70      3-71      3-74      3-76      3-77      5-8#
DORTS    3-17      3-18      3-19      3-20      3-25      3-26      3-29      3-30      3-31      3-32      3-40      3-41
         3-44      3-45      3-46      3-47      3-60      3-75      3-88      3-89      3-90      3-91      3-105     3-106
         3-107     3-108     3-120     3-121     3-122     3-123     3-135     3-136     3-137     3-138     3-150     3-151
         3-152     3-153     3-157#
DSSCOM   11-8      11-11#
DZCLOK   1-34      3-85
DZCLR    7-31      7-48#
DZGDSS   1-34      3-102
DZOINT   1-13      7-14#
DZRTN    7-21      7-26      7-41#     7-51
DZSBRK   1-31      3-132
DZSDSS   1-35      3-117
DZSSPD   1-31      3-147
DZSTRT   1-17      3-14      8-7#
GETDSS   1-18      11-7#
ININT    1-15      9-12#
```

Cross reference table (CREF V05.05)

```
ININTP   3-163#    9-12
INTPRI   1-41
KPAR5    1-36     10-10     10-19*    11-12     11-25*
KPAR6    1-23
LCDTYP   1-39      4-22      5-21     10-14     11-16
LINRTS   1-15      3-193#
LMXLN    1-23      7-49      8-10
LMXNUM   1-25      8-9
LOUTIR   1-36      6-22      7-30
LSW10    1-37      4-13*     4-18*     5-12*     5-17*     9-22*
LSW3     1-26      6-27*     6-41*     7-35*     7-48*
LSW5     1-26      6-28*     7-36*
MAPHLD   3-177#
MXCSR    1-27      7-20
MXLBIT   3-188#    7-50      8-11
MXLNT    1-27      7-24
MXTBUF   1-28      7-37*
MXTCR    1-28      7-50*     8-11*
NEDCDI   1-14      3-178#
NEDCDO   1-14      3-179#    9-23*
NEDCHR   1-28      9-6#
NEDCLO   1-18      3-180#
NEDCP    3-168#    9-6
NEDSOT   1-13      3-181#
PR7      1-28
PSW      1-28
REQCDO   3-15      3-16      9-22#
RLINE    1-29      7-22
SETDSS   1-18     11-9#
SETSPD   1-16     10-9#
SILFET   1-18      3-165#
SNDFRE   3-171#
TBR      1-30      6-31*
TIOVEC   1-18      3-162#
TRINT    1-30      6-42      6-51
TRNSTR   1-18      3-167#
TRRDY    1-30      6-29
TSR      1-30      6-29      6-42*     6-51*
TSTIO    1-13      3-12#
TTINPT   1-15      3-164#
TTRSAV   1-15      3-166#
VHCLOK   1-42      3-87
VHGDSS   1-34      3-104
VHSBRK   1-24      3-134
VHSDSS   1-35      3-119
VHSSPD   1-24      3-149
VHSTOP   1-41      3-43
VHSTRT   1-32      3-28
VHXOFF   1-33      3-58
VHXON    1-33      3-73
```

Cross reference table (CREF V05.05)

```
DISABL    2-6#
ENABL     2-12#
OCALL     2-18#
TTMAP     2-48#
TTMAPX    2-39#
UNMAP     2-53#
UNMAPX    2-45#
```