Table of contents

```
                                        .TITLE  TSDUMP -- TSX-Plus crash dump
                                        .ENABL  LC
                                        .DSABL  CBL
                                        .ENABL  AMA
000000                                  .CSECT  TSDUMP
000000  015430                          .RAD50  /DMP/           ;System overlay id
                                ;-----------------------------------------------------------------
                                ;  This module implements the TSX-Plus crash dump facility.
                                ;
                                ;  Copyright (c) 1985.
                                ;  S&H Computer Systems, Inc.
                                ;  Nashville, Tennessee  USA
                                ;  All rights reserved.
                                ;
                                ;  The crash dump facility provides diagnostic information when
                                ;  a fatal system error occurs.  It is entered with interrupts
                                ;  disabled and prior initialization of several stored values.
                                ;
```

```
                        ;-----------------------------------------------------------------------
                        ;  GLOBAL Definitions.
                        ;
                                .GLOBL  DODUMP

                        ;-----------------------------------------------------------------------
                        ;  GLOBAL References.
                        ;
                                .GLOBL  SS,SSEND,INTSTK,INTSND,JSTK,JSTKND
                                .GLOBL  VDMTCR,PSW
                                .GLOBL  DMPTXT,DMPOVL,DMPHND,DIEAR0,DIEPC,DIESP
                                .GLOBL  TSGEN,TSTIO,TSEXEC,TSEMT,TSINIT
                                .GLOBL  TRPAR5,KPAR6,VPAR5,VPAR6
                                .GLOBL  CORUSR,NUMDEV,MAXDEV
                                .GLOBL  NPL,NSL,NDL,NIOL,NLINES,TNHL
                                .GLOBL  SYSDAT                  ;System date (TSGEN)
                                .GLOBL  SYTIMH,SYTIML           ;System time (TSGEN)
                                .GLOBL  TK1SEC                  ;Number of ticks per second
                                .GLOBL  OVRADD                  ;Address of overlay tables
```

```
                        ;-----------------------------------------------------------------------
                        ;  MACRO Definitions.
                        ;
                        ;-----------------------------------------------------------------------
                        ;  DISABLE macro used to disable all interrupts.

                                .MACRO  DISABL          ;DISABLE INTERRUPTS
                                BIS     #340,@#PSW
                                .ENDM   DISABL

                        ;-----------------------------------------------------------------------
                        ;  STKDEF macro used to define a stack where:
                        ;       BASE    = base of stack
                        ;       TOP     = top of stack
                        ;       LENGTH  = length (in words) of stack

                                .MACRO  STKDEF  top, base, length
                        base:                                   ;Base address of stack
                                .BLKW   length                  ;Allocate stack space
                        top:                                    ;Top address of stack
                                .ENDM   STKDEF

                        ;-----------------------------------------------------------------------
                        ;  PRINT macro used to define a print string where
                        ;       stradr  = address of asciz string to print

                                .MACRO  PRINT   ?stradr
                                .NARG   ARG                     ;Determine number of arguments
                                .IF     NE, ARG                 ;Only if argument exists
                                MOV     stradr,R1               ;Point to string address
                                .ENDC   ; NE, ARG
                                CALL    OUTSTR                  ;Display asciz string
                                .ENDM   PRINT

                        ;-----------------------------------------------------------------------
                        ;  TTYOUT macro used to output a single character where
                        ;       char    = character to output

                                .MACRO  TTYOUT  ?char
                                .NARG   ARG                     ;Determine number of arguments
                                .IF     NE, ARG                 ;Only if argument exists
                                MOVB    char,r0                 ;Get output character
                                .ENDC   ;NE ARG
                                CALL    OUTCHR                  ;Display ascii character
                                .ENDM   TTYOUT
```

```
                        ;-----------------------------------------------------------------
                        ;  TBEG macro used to define a table entry print field where:
                        ;       LABEL   = address label of table
                        ;       NAME    = table title
                        ;       ROWNAM  = row name label
                        ;       ATTR    = index number attribute
                        ;       SINDX   = starting index
                        ;       EINDX   = ending index
                        ;
                        .MACRO  TBEG    label, name, rownam, attr, sindx, eindx
                        .NARG   ARGS                    ;Number of arguments to macro call
                        .PSECT  DUMP                    ;Define DUMP program section
                                label   =       .       ;Define starting address
                        .PSECT  TSDUMP                  ;Restore TSDUMP program section
                        .PSECT  STRING                  ;Define TEXT program section
                                STRADD  =       .       ;Point to the starting address
                                .ASCII  /name/          ;Store the ascii label
                                .BYTE   0               ;Terminate as asciz string
                        .PSECT  DUMP                    ;Define DUMP program section
                                .WORD   STRADD          ;Store the string address
                                .IF     EQ, ARGS-2      ;Only two arguments passed
                                        TENABL  = 0     ;Flag list definition in progress
                                        .MEXIT          ;Terminate macro expansion
                                .ENDC   ; EQ, ARGS-2
                                TENABL  = 1             ;Flag table definition in progress
                        .PSECT  STRING                  ;Define STRING program section
                                STRADD  =       .       ;Point to the starting address
                                .ASCII  /rownam/        ;Store the ascii label
                                .BYTE   200             ;Terminate as asciz string
                        .PSECT  DUMP                    ;Define DUMP program section

                                .WORD   STRADD          ;Store the string address
                                .WORD   attr            ;Store table attributes
                                .WORD   STRADD          ;Store the string address
                                .WORD   sindx           ;Store starting table index
                                .WORD   eindx           ;Store ending table index
                        .PSECT  TSDUMP                  ;Restore TSDUMP program section
                        .ENDM   TBEG
```

```
                        ;-------------------------------------------------------------------
                        ;  PVAL macro used to define a single print field where:
                        ;       NAME    = ascii string title
                        ;       LABEL   = address label containing print value
                        ;       ATTR    = field attributes
                        ;       SINX    = starting table index
                        ;       EINX    = ending table index
                        ;               Format attributes:
                        ;               $RJUST = (bit 14) right justify (default is left)
                        ;               $ZRPAD = (bit 13) zero pad (default is blank)
                        ;               Storage attributes:
                        ;               $BYTE = (bit 0) storage requirement (default is word)
                        ;               $DEC = (bit 2) decimal representation (default is octal)
                        ;               $RAD50 = (bit 3) rad50 representation
                        ;

                        ;  Definition of field attribute bit definitions.

040000                          $RJUST  = 40000                 ;Right justify
020000                          $ZRPAD  = 20000                 ;Zero pad
                        ;  Storage attributes.
000001                          $BYTE   = 1                     ;Byte storage cell
000002                          $DEC    = 2                     ;Decimal numeric representation
000004                          $RAD50  = 4                     ;Rad50 representation

                                .MACRO  PVAL name, label, attr, sindx, eindx
                                .PSECT  STRING                  ;Define STRING program section
                                STRADD  =       .               ;Point to the starting address
                                .ASCII  /name/                  ;Store the ascii label
                                .BYTE   200                     ;Terminate as asciz string
                        .PSECT  DUMP                            ;Define DUMP program section
                                .WORD   STRADD                  ;Store the string address
                                .WORD   attr                    ;Store field attributes
                                .GLOBL  label                   ;Declare variable as global
                                .WORD   label                   ;Store location of value field
                                .IF     NE, tenabl              ;Has table definition been enabled?
                                .WORD   sindx                   ;Store starting table index
                                .WORD   eindx                   ;Store ending table index
                                .ENDC   ; NE, tenabl
                        .PSECT  TSDUMP                          ;Restore TSDUMP program section
                        .ENDM   PVAL
```

```
                        ;-----------------------------------------------------------------------
                        ;  TEND macro used to define the end of a table or list.
                        ;

                                .MACRO  TEND
                                .PSECT  DUMP                    ;Define DUMP program section
                                        .WORD   0               ;End of table or list
                                .PSECT  TSDUMP                  ;Restore TSDUMP program section
                                .ENDM   TEND
```

```
                                        .SBTTL  Output list for registers
                                ;-------------------------------------------------------------
                                ;  Generate list for output of register contents.
                                ;
000002                                  TBEG    PRTREG, <Register Contents>     ;Beginning of print list
000002                                  PVAL    <R0>,SVR0                       ;Register 0
000002                                  PVAL    <R1>,SVR1                       ;Register 1
000002                                  PVAL    <R2>,SVR2                       ;Register 2
000002                                  PVAL    <R3>,SVR3                       ;Register 3
000002                                  PVAL    <R4>,SVR4                       ;Register 4
000002                                  PVAL    <R5>,SVR5                       ;Register 5
000002                                  PVAL    <SP>,DIESP                      ;Stack pointer
000002                                  PVAL    <PC>,DIEPC                      ;Program counter
000002                                  PVAL    <KPAR 5>,SVPAR5                 ;Kernal PAR 5
000002                                  PVAL    <KPAR 6>,SVPAR6                 ;Kernal PAR 6
000002                                  TEND                                    ;End of print list
```

```
                                        .SBTTL  Output list for TSX-Plus addresses
                                ;-----------------------------------------------------------------
                                ;  Generate list for output of TSX-Plus address values.
                                ;
000002                                  TBEG    ADDR, <TSX-Plus module addresses>
000102                                  PVAL    <TSGEN >,ADGEN          ;Address of TSGEN
000002                                  PVAL    <TSTIO >,ADTIO          ;Address of TSTIO
000002                                  PVAL    <TSEXEC>,ADEXEC         ;Address of TSEXEC
000002                                  PVAL    <TSEMT >,ADEMT          ;Address of TSEMT
000002                                  PVAL    <TSINIT>,ADINIT         ;Address of TSINIT
000002                                  PVAL    <TSTIOX>,ADTIOX         ;Address of TSTIOX
000002                                  PVAL    <TSLOCK>,ADLOCK         ;Address of TSLOCK
000002                                  PVAL    <TSCASH>,ADCASH         ;Address of TSCASH
000002                                  TEND
```

```
                        .SBTTL  Output list for miscellaneous fields
                ;-----------------------------------------------------------------
                ;  Generate list for output of miscellaneous fields.
                ;
000002                  TBEG    VMISC, <Miscellaneous values>   ;Beginning of print list
000166                  PVAL    <CLKPC>,CLKPC                   ;Clock interrupted PC (for PM)
000002                  PVAL    <CLKPS>,CLKPS                   ;Clock interrupted PS (for PM)
000002                  PVAL    <CSHBAS>,CSHBAS                 ;Physical add of TSCASH
000002                  PVAL    <CURFRK>,CURFRK                 ;Address of current FORK call
000002                  PVAL    <CXBJOB>,CXBJOB,<$BYTE> ;Context block job contents
000002                  PVAL    <CXBOWN>,CXBOWN,<$BYTE> ;Context block job owner
000002                  PVAL    <DOSCHD>,DOSCHD,<$BYTE> ;Do scheduling flag
000002                  PVAL    <EXCJOB>,EXCJOB,<$BYTE> ;Exclusive system job owner
000002                  PVAL    <FREIOQ>,FREIOQ                 ;Free I/O queue head
000002                  PVAL    <FRKCQE>,FRKCQE                 ;Fork queue head
000002                  PVAL    <FREFRK>,FREFRK                 ;Free fork queue head
000002                  PVAL    <FRKPRI>,FRKPRI,<$BYTE> ;Fork priority
000002                  PVAL    <INBSY>,INBSY,<$BYTE>   ;Inswap flag
000002                  PVAL    <INTLVL>,INTLVL,<$BYTE> ;Interrupt level
000002                  PVAL    <INTPRI>,INTPRI                 ;Interrupt priority
000002                  PVAL    <IOABFL>,IOABFL                 ;I/O abort flag (0=rundown; 1=abort)
000002                  PVAL    <LOKBAS>,LOKBAS                 ;PAR base of TSLOCK
000002                  PVAL    <LOKSWP>,LOKSWP                 ;Job requiring memory lock
000002                  PVAL    <MAPUSR>,MAPUSR,<$BYTE> ;Job currently mapped
000002                  PVAL    <MEM256>,MEM256,<$BYTE> ;Extended memory flag
000002                  PVAL    <MEMSWP>,MEMSWP                 ;Job requiring memory expansion
000002                  PVAL    <MIOFL0>,MIOFL0,<$BYTE> ;I/O mapping flag
000002                  PVAL    <MSGBAS>,MSGBAS                 ;PAR base of TSMSG
000002                  PVAL    <NMFREQ>,NMFREQ                 ;Number of free user's queue elements
000002                  PVAL    <NUMON>,NUMON,<$BYTE>   ;Num of hardware lines on
000002                  PVAL    <OUTBSY>,OUTBSY,<$BYTE> ;Outswap flag
000002                  PVAL    <PVON>,PVON,<$BYTE>     ;Num of primary and virtual lines on
000002                  PVAL    <SPDJOB>,SPDJOB,<$BYTE> ;SPD job owner
000002                  PVAL    <SR3FL0>,SR3FL0,<$BYTE> ;Extended memory management reg. flag
000002                  PVAL    <STKLVL>,STKLVL,<$BYTE> ;Stack level
000002                  PVAL    <TIOBAS>,TIOBAS                 ;Physical add of TSTIOX
000002                  PVAL    <TOTON>,TOTON,<$BYTE>   ;Num of total lines on
000002                  PVAL    <UBUSMP>,UBUSMP,<$BYTE> ;Unibus map flag
000002                  PVAL    <UIOCNT>,UIOCNT                 ;User I/O count
000002                  PVAL    <USP>,USP                       ;User stack pointer
000002                  PVAL    <USRBAS>,USRBAS                 ;PAR base of TSUSR
000002                  PVAL    <USRJOB>,USRJOB,<$BYTE> ;USR job owner
000002                  PVAL    <VBUSTP>,VBUSTP,<$BYTE> ;Bus type (0 = UNIBUS; 1 = QBUS)
000002                  PVAL    <VCSHNB>,VCSHNB                 ;Num of generalized cache blocks
000002                  PVAL    <VMXSF>,VMXSF                   ;Max num of shared files
000002                  PVAL    <VMXSFC>,VMXSFC                 ;Max num of shared file channels
000002                  PVAL    <VNGR>,VNGR                     ;Num of global PLAS regions
000002                  PVAL    <VNUMDC>,VNUMDC                 ;Num of shared data cache blocks
000002                  PVAL    <VPLAS>,VPLAS                   ;Num of PLAS blocks allocated
000002                  PVAL    <VSWPFL>,VSWPFL,<$BYTE> ;Swap flag (0 = swap; 1 = noswap)
000002                  PVAL    <VSWPSL>,VSWPSL                 ;Num of swap file jobs allocated
000002                  PVAL    <VUXIFL>,VUXIFL,<$BYTE> ;Unexpected interrupt (1 = abort)
000002                  TEND
```

```
                        .SBTTL  Output list for line status tables
                ;-----------------------------------------------------------------
                ;  Generate table for output of line status.
                ;
000002                  TBEG    JLIN, <Job Status Tables>, <Job>, <$BYTE!$DEC>, 1, NLINES
000002                  PVAL    <LSW>,LSW,<$RJUST!$ZRPAD>           ;Job status word
000002                  PVAL    <LSW2>,LSW2,<$RJUST!$ZRPAD>         ;Job status word 2
000002                  PVAL    <LSW3>,LSW3,<$RJUST!$ZRPAD>         ;Job status word 3
000002                  PVAL    <LSW4>,LSW4,<$RJUST!$ZRPAD>         ;Job status word 4
000002                  PVAL    <LSW5>,LSW5,<$RJUST!$ZRPAD>         ;Job status word 5
000002                  PVAL    <LSW6>,LSW6,<$RJUST!$ZRPAD>         ;Job status word 6
000002                  PVAL    <LSW7>,LSW7,<$RJUST!$ZRPAD>         ;Job status word 7
000002                  PVAL    <LSW8>,LSW8,<$RJUST!$ZRPAD>         ;Job status word 8
000002                  PVAL    <LSW9>,LSW9,<$RJUST!$ZRPAD>         ;Job status word 9
000002                  PVAL    <LSW10>,LSW10,<$RJUST!$ZRPAD>       ;Job status word 10
000002                  PVAL    <LSW11>,LSW11,<$RJUST!$ZRPAD>       ;Job status word 11
000002                  PVAL    <LSTATE>,LSTATE,<$RJUST!$ZRPAD>     ;Job state
000002                  PVAL    <LBASE>,LBASE,<$RJUST!$ZRPAD>       ;Base 512-block num of context
000002                  PVAL    <LPARBS>,LPARBS,<$RJUST!$ZRPAD>     ;Base PAR of program
000002                  PVAL    <LNBLKS>,LNBLKS,<$RJUST!$ZRPAD>     ;Num of mem pages for job
000002                  PVAL    <LNSBLK>,LNSBLK,<$RJUST!$ZRPAD>     ;Num of mem pages for PLAS
000002                  PVAL    <LIOCNT>,LIOCNT,<$RJUST!$ZRPAD>     ;Active I/O counter
000002                  PVAL    <LINCNT>,LINCNT,<$RJUST!$ZRPAD>     ;Input chars pending
000002                  PVAL    <LACTIV>,LACTIV,<$RJUST!$ZRPAD>     ;Activ chars pending
000002                  PVAL    <LPRG1>,LPRG1,<$RAD50>              ;Program name
000002                  PVAL    <LPRG2>,LPRG2,<$RAD50>              ;Program name
000002                  PVAL    <LCDTYP>,LCDTYP,<$RJUST!$ZRPAD>,1,TNHL  ;Hardware line status
000002                  PVAL    <LNPRIM>,LNPRIM,<$RJUST!$ZRPAD>,1,NPL+NDL;Primary line index
000002                  PVAL    <LPARNT>,LPARNT,<$RJUST!$ZRPAD>     ;Parent number index
000002                  PVAL    <LBSPRI>,LBSPRI,<$RJUST!$ZRPAD>     ;Job base priority
000002                  PVAL    <LCLUNT>,LCLUNT,<$RJUST!$ZRPAD>,1,NPL+NSL+NDL+NIOL;CL unit
000002                  PVAL    <LMEMIN>,LMEMIN,<$RJUST!$ZRPAD>     ;Num mem pages require
000002                  PVAL    <LIOHLD>,LIOHLD,<$RJUST!$ZRPAD>     ;I/O hold for swap
000002                  PVAL    <LAFSIZ>,LAFSIZ,<$RJUST!$ZRPAD>     ;Field width activate
000002                  PVAL    <LFWLIM>,LFWLIM,<$RJUST!$ZRPAD>     ;Field limit activate
000002                  PVAL    <LCMPL>,LCMPL,<$RJUST!$ZRPAD>       ;Pending completion
000002                  PVAL    <LSWPBK>,LSWPBK,<$RJUST!$ZRPAD>     ;Swap file block num
000002                  PVAL    <LJSW>,LJSW,<$RJUST!$ZRPAD>         ;Job status word
000002                  PVAL    <LEMTPC>,LEMTPC,<$RJUST!$ZRPAD>     ;Last EMT PC address
000002                  PVAL    <LSPND>,LSPND,<$RJUST!$ZRPAD>       ;Job .SPND counter
000002                  PVAL    <LXCL>,LXCL,<$RJUST!$ZRPAD>,1,NPL;Cross connect CL unit index
000002                  TEND
```

```
                                        .SBTTL  Output list for device handler status
                                ;-------------------------------------------------------------
                                ;  Generate table for output of device handler status.
                                ;
000002                                  TBEG    DHAN,<Device Handler Tables>,<Offset>,<$BYTE!$DEC>,0,MAXDEV-1
000002                                  PVAL    <PNAME>,PNAME,<$RAD50>              ;Program name
000002                                  PVAL    <HANENT>,HANENT,<$RJUST>            ;Handler entry point
000002                                  PVAL    <HANPAR>,HANPAR,<$RJUST>            ;Handler PAR
000002                                  PVAL    <HANSIZ>,HANSIZ,<$RJUST>            ;Handler size
000002                                  PVAL    <HANIOC>,HANIOC,<$RJUST>            ;Handler I/O count
000002                                  PVAL    <DVFLAG>,DVFLAG,<$RJUST!$ZRPAD>     ;Device handler flags
000002                                  PVAL    <DVSTAT>,DVSTAT,<$RJUST!$ZRPAD>     ;Device status flags
000002                                  PVAL    <DEVSIZ>,DEVSIZ,<$RJUST!$ZRPAD>     ;Device size (256 word blks)
000002                                  TEND
```

```
                                        .SBTTL  Data areas
                                ;-------------------------------------------------------------------
                                ;  Internal storage.
                                ;
                                ;  Assignments.
                                ;
000015                          CR      =       15              ;Carriage return
000012                          LF      =       12              ;Line feed
000021                          CTRLQ   =       21              ;Control-Q
000023                          CTRLS   =       23              ;Control-S
000055                          DASH    =       '-              ;Dash
000014                          NCOL    =       12.             ;Number of columns for table output

                                ;
                                ;  Internal stack and register contents.
                                ;
000002  000000                  SVR0:   .WORD   0
000004  000000                  SVR1:   .WORD   0
000006  000000                  SVR2:   .WORD   0
000010  000000                  SVR3:   .WORD   0
000012  000000                  SVR4:   .WORD   0
000014  000000                  SVR5:   .WORD   0               ;Saved registers (r5 to r0)
000016'                         SVREG   = .                     ;Past saved registers
000016  000000                  SVPAR5: .WORD   0               ;Saved kernel PAR 5
000020  000000                  SVPAR6: .WORD   0               ;Saved kernel PAR 6
                                ;
                                ;  Root module addresses.
                                ;
000022  000000                  ADGEN:  .WORD   0               ;Address of TSGEN
000024  000000                  ADTIO:  .WORD   0               ;Address of TSTIO
000026  000000                  ADEXEC: .WORD   0               ;Address of TSEXEC
000030  000000                  ADEMT:  .WORD   0               ;Address of TSEMT
000032  000000                  ADINIT: .WORD   0               ;Address of TSINIT
000034  000000                  ADTIOX: .WORD   0               ;Address of TSTIOX
000036  000000                  ADLOCK: .WORD   0               ;Address of TSLOCK
000040  000000                  ADCASH: .WORD   0               ;Address of TSCASH
                                ;
                                ;  Miscellaneous storage.
                                ;
                                ;  Word bounded storage.
                                ;
000042  000000                  TBUF:   .WORD   0               ;Transmit buffer
000044  000000                  TCSR:   .WORD   0               ;Transmit CSR
000046  000000                  RBUF:   .WORD   0               ;Receiver buffer
000050  000000                  RCSR:   .WORD   0               ;Receiver CSR
000052  000000                  COL:    .WORD   0               ;Current column position
                                ;
                                ;  Format tables for columnar output.
                                ;
000054                          TSCOL:  .BLKW   NCOL+1          ;Starting column on table output
000106                          TECOL:  .BLKW   NCOL+1          ;Ending column on table output
000140                          TSINX:  .BLKW   NCOL+1          ;Starting index for column
000172                          TEINX:  .BLKW   NCOL+1          ;Ending index for column
000224                          TATT:   .BLKW   NCOL+1          ;Attributes of table
000256                          TADR:   .BLKW   NCOL+1          ;Base address of table
                                ;
                                ;  Data.
```

Data areas

```
                                        ;
000310  037266  023112  050572  MONTH:  .RAD50  /JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC/
000316  004322  050601  040726
000324  040724  004617  073630
000332  057114  054756  014713
                                        ;
                                        ;  Byte bounded storage.
                                        ;
000340      021                 XFLG:   .BYTE   CTRLQ           ;XON/XOFF flag
                                        ;
                                        ;  Output strings.
                                        ;
                                        .NLIST  BEX
000341      101     162     147 TXARG:  .ASCII  /Arg. value = /<200>
000357      123     145     147 TXSEG:  .ASCII  /Seg. value = /<200>
000375      117     166     145 TXOID:  .ASCII  /Overlay: /<200>
000407      104     145     166 TXDEV:  .ASCII  /Device name: /<200>
000425      040     075     040 EQSTR:  .ASCII  / = /<200>
000431      040     040     040 SPSTR:  .ASCII  /     /<200>
000437      014     200         FF:     .BYTE   14, 200         ;Form feed
000441      015     012     200 CRLF:   .BYTE   CR, LF, 200     ;Carriage return / line feed
000444      124     123     130 OVLREG: .ASCIZ  /TSX-Plus overlay regions/
000475      123     171     163 STKSS:  .ASCIZ  /System /<200>
000506      111     156     164 STKINT: .ASCIZ  /Interrupt /<200>
000522      105     115     124 STKEMT: .ASCIZ  /EMT /<200>
000530      123     164     141 STKCNT: .ASCIZ  /Stack Contents/
000547      040     057     040 STKDLM: .ASCII  \ / \<200>
000553      111     156     166 INVSTK: .ASCIZ  /Invalid stack pointer/
000601      111     156     164 INTLEV: .ASCIZ  /Interrupt Level/
000621      106     157     162 FRKLEV: .ASCIZ  /Fork Level/
000634      123     171     163 STKOVR: .ASCIZ  /System stack overflow/
000662      112     157     142 CURJOB: .ASCII  /Job executing at time of dump/<200>
                                        .LIST   BEX
                                        .EVEN
```

```
                                        .SBTTL  DODUMP -- Dump entry point
                                ;-----------------------------------------------------------------
                                ;  TSDUMP entry point.  TSDUMP is entered from TSEXEC with the
                                ;  following known values:
                                ;       DMPTXT  = Asciz text of die message
                                ;       DMPOVL  = Rad50 overlay name (or zero)
                                ;       DMPHND  = Rad50 handler name (or zero)
                                ;       DIEMSG  = Address of die message
                                ;       DIEARG  = Argument of die message
                                ;       DIESP   = Original stack pointer at time of crash
                                ;       DIEPC   = Program counter of DIE call
                                ;       TRPAR5  = PAR 5 value at trap
                                ;       SP      = All registers - R0,R1,R2,R3,R4,R5
                                ;
                                ;  A non-interrupt output routine is used to display information to
                                ;  an XON/XOFF device or a parallel port printer.  The following
                                ;  information is output:
                                ;       Registers (R0 through R5)
                                ;       SP (stack pointer)
                                ;       PC (program counter from error)
                                ;       PSW
                                ;       Stack contents
                                ;       Miscellaneous data cells
                                ;       Line definition tables
                                ;       Device definition tables
                                ;
000720                          DODUMP:
                                ;
                                ;  Disable interrupt and locate the dump device registers.
                                ;
                                ;       DISABL                          ;Disable all interrupts
000720  012737  001124' 000004          MOV     #DMPTRP,@#4             ;Capture the trap 4 address
                                ;       MOV     #340,@#6                ;Retain disable interrupt status
000726  013705  0000000                 MOV     VDMTCR,R5               ;Find the display device CSR
000732  042705  000004                  BIC     #4,R5                   ;Adjust for receiver CSR
000736  005015                          CLR     (R5)                    ;Check for receiver CSR
000740  103411                          BCS     1$                      ;Br if receiver not available
000742  010537  000050'                 MOV     R5,RCSR                 ;Insert receiver address
000746  010537  000046'                 MOV     R5,RBUF                 ;Insert receiver buffer
000752  062737  000002  000046'         ADD     #2,RBUF                 ; offset 2 from CSR
000760  117700  177062                  MOVB    @RBUF,R0                ;Throw away any pending input
000764  062705  000004          1$:     ADD     #4,R5                   ;Ajust to transmit CSR
000770  005015                          CLR     (R5)                    ;Check for transmitter CSR
000772  103450                          BCS     50$                     ;Br if transmitter not available
000774  010537  000044'                 MOV     R5,TCSR                 ;Insert transmiter CSR
001000  010537  000042'                 MOV     R5,TBUF                 ;Insert transmiter buffer address
001004  062737  000002  000042'         ADD     #2,TBUF                 ; offset 2 from CSR
                                ;
                                ;  Save values on entry (registers, kernel mapping, etc.).
                                ;
001012  013737  0000000 000020'         MOV     @#KPAR6,SVPAR6          ;Save kernel PAR 6
001020  013737  0000000 000016'         MOV     TRPAR5,SVPAR5           ;Save kernel PAR 5
001026  012705  000016'                 MOV     #SVREG,R5               ;Save registers on current stack
001032  012700  000006                  MOV     #6,R0                   ;Save registers R0-R6
001036  012645                  10$:    MOV     (SP)+,-(R5)             ;Save all the register contents
001040  077002                          SOB     R0,10$                  ;Continue until all register saved
```

```
                                        ;
                                        ;  Output information concerning the current state.
                                        ;
001042                                          PRINT   #FF             ;Output form feed character
001052  004737  001134'                         CALL    DATTIM          ;Output date and time information
001056  004737  001462'                         CALL    ERROR           ;Output error status information
001062  004737  002400'                         CALL    JOB             ;Output current job information
001066  004737  001626'                         CALL    REG             ;Output the register contents
001072  004737  001652'                         CALL    MODADR          ;Output TSX-Plus module addresses
001076  004737  001776'                         CALL    OVRADR          ;Output TSX-Plus overlay regions
001102  004737  002300'                         CALL    MISC            ;Output miscellaneous information
001106  004737  002460'                         CALL    LINE            ;Output line table information
001112  004737  002472'                         CALL    DEVICE          ;Output handler table information
001116  004737  002072'                         CALL    STACK           ;Output the stack contents
001122  000000                        50$:     HALT                    ;Now where to?
                                        ;
                                        ;  Trap to 4 control.
                                        ;
001124  052766  000001  000002  DMPTRP: BIS     #1,2(SP)                ;Buffer not found, set c-bit
001132  000002                                  RTI                     ;Return from trap
```

```
                        .SBTTL  DATTIM -- Display date and time information
                ;-----------------------------------------------------------------
                ;  DATTIM is called to display the current date and time information.
                ;
001134          DATTIM: PRINT   #CRLF           ;Display CR/LF
001144  013705  0000000         MOV     SYSDAT,R5       ;Get current date
001150  001451                  BEQ     10$             ;No date set - skip date
001152  010501                  MOV     R5,R1           ;Copy to working register
001154  072127  177773          ASH     #-5,R1          ;Find the day (bits 5-9)
001160  042701  177740          BIC     #^C37,R1        ;Mask unwanted bits
001164  012703  000002          MOV     #$DEC,R3        ;Force decimal display
001170  004737  003626'         CALL    OUTVAL          ;Print the day
001174                          TTYOUT  #'-             ;Output "-"
001204  010501                  MOV     R5,R1           ;Copy date to working register
001206  072127  177767          ASH     #-9.,R1         ;Find the month (bits 10-13)
001212  042701  177741          BIC     #^C36,R1        ;Mask unwanted bits (word offset)
001216  016101  000306'         MOV     MONTH-2(R1),R1  ;Offset into month table
001222  012703  000004          MOV     #$RAD50,R3      ;Force rad50 display
001226  004737  004022'         CALL    OUTRAD          ;Output three char month
001232                          TTYOUT  #'-             ;Output "-"
001242  010501                  MOV     R5,R1           ;Copy date to working register
001244  042701  177740          BIC     #^C37,R1        ;Find the year (bits 0-4)
001250  062701  000110          ADD     #72.,R1         ;Year is relative to 1972
001254  012703  000002          MOV     #$DEC,R3        ;Force decimal display
001260  004737  003626'         CALL    OUTVAL
001264                          PRINT   #SPSTR          ;Print blank space
                ;
                ;  Output the current time.
                ;
001274  013700  0000000 10$:    MOV     SYTIMH,R0       ;Get the current time (high)
001300  013701  0000000         MOV     SYTIML,R1       ;  clock ticks past midnight (low)
001304  013703  0000000         MOV     TK1SEC,R3       ;Get number of clock ticks per second
001310  004737  001424'         CALL    DIV32           ;Divide clock ticks per second
001314  010246                  MOV     R2,-(SP)        ;Save ticks
001316  012703  000074          MOV     #60.,R3         ;Get 60. per unit
001322  012705  000002          MOV     #2,R5           ;Process seconds, minutes, hours
001326  004737  001424' 11$:    CALL    DIV32           ;Divide by 60. per unit
001332  010246                  MOV     R2,-(SP)        ;Save remainder
001334  077504                  SOB     R5,11$          ;Continue for seconds, minutes, hours
001336  010146                  MOV     R1,-(SP)        ;Save final quotient
001340  012705  000004          MOV     #4,R5           ;Loop for minutes, seconds, and ticks
001344  000404                  BR      21$             ;Output the time past midnight
001346                  20$:    TTYOUT  #':             ;Output ":"
001356  012601          21$:    MOV     (SP)+,R1        ;Get the next value
001360  005000                  CLR     R0              ;Clear high order
001362  071027  000012          DIV     #10.,R0         ;Divide by ten
001366  062700  000060          ADD     #'0,R0          ;Convert to ascii number
001372                          TTYOUT                  ;Display high digit (quotient)
001376  062701  000060          ADD     #'0,R1          ;Convert to ascii number
001402                          TTYOUT  R1              ;Display low digit (remainder
001410  077522                  SOB     R5,20$          ;Continue for hours, minutes, seconds, ticks
001412                          PRINT   #CRLF           ;Display CR/LF
001422  000207                  RETURN
```

```
                                        .SBTTL  DIV32  -- Divide 16-bit into 32-bit
                                ;-----------------------------------------------------------------
                                ;  DIV32 divides a 16-bit number in R3 into a 32-bit number
                                ;  in R0 (high order) and R1 (low order).  The quotient is
                                ;  returned in R0, R1 and the remainder is returned in R2.
                                ;  All other registers are preserved.
                                ;
001424  010446                  DIV32:  MOV     R4,-(SP)        ;Save registers
001426  005002                          CLR     R2              ;Initialize remainder
001430  012704  000037                  MOV     #31.,R4         ;Initialize shift count
001434  006301                  1$:     ASL     R1              ;Shift from low order
001436  006100                          ROL     R0              ; through high order
001440  006102                          ROL     R2              ; into remainder
001442  020203                          CMP     R2,R3           ;Check remainder for subtraction
001444  103402                          BLO     2$              ;Br if unable to subtract
001446  160302                          SUB     R3,R2           ;Subtract divisor
001450  005201                          INC     R1              ;Increment quotient
001452  005304                  2$:     DEC     R4              ;Check remaining bit shift count
001454  100367                          BPL     1$              ;Continue if more to consider
001456  012604                          MOV     (SP)+,R4        ;Restore registers
001460  000207                          RETURN
```

```
                                        .SBTTL  ERROR  -- Display error status information
                                ;-----------------------------------------------------------------
                                ;  ERROR is called to display the error status information.
                                ;
001462                          ERROR:  PRINT   #CRLF           ;Display CR/LF
001472                                  PRINT   #DMPTXT         ;Display fatal error message
001502                                  PRINT   #TXARG          ;Display "Arg. value"
001512  013701  0000000                 MOV     DIEARG,R1       ;Get the argument value
001516  005003                          CLR     R3              ;Set display characteristics
001520  004737  003626'                 CALL    OUTVAL          ;Output argument value
001524                                  PRINT   #CRLF           ;Display CR/LF
001534  013705  0000000                 MOV     DMPOVL,R5       ;Get the rad50 overlay identifier
001540  001413                          BEQ     1$              ;Br if zero, not in overlay
001542                                  PRINT   #TXOID          ;Display "Overlay: "
001552  010501                          MOV     R5,R1           ;Copy overlay identifier
001554  004737  004022'                 CALL    OUTRAD          ;Output rad50 overlay identifier
001560                                  PRINT   #CRLF           ;Display CR/LF
001570  013705  0000000         1$:     MOV     DMPHND,R5       ;Get the rad50 handler identifier
001574  001413                          BEQ     2$              ;Br is zero, not in handler
001576                                  PRINT   #TXDEV          ;Display "Device name: "
001606  010501                          MOV     R5,R1           ;Copy device handler identifier
001610  004737  004022'                 CALL    OUTRAD          ;Output rad50 handler identifier
001614                                  PRINT   #CRLF           ;Display CR/LF
001624  000207                  2$:     RETURN                  ;Finished
```

```
      1                                         .SBTTL  REG     -- Display contents of registers
      2                                 ;-----------------------------------------------------------
      3                                 ;  REG is called to display the contents of the registers.
      4                                 ;
      5 001626                          REG:    PRINT   #CRLF           ;Display CR/LF
      6 001636  005002                          CLR     R2              ;Clear offset pointer
      7 001640  012705  000000'                 MOV     #PRTREG,R5      ;Point to register output table
      8 001644  004737  002504'                 CALL    OLIST           ;Output the list of values
      9 001650  000207                          RETURN
```

```
                                        .SBTTL  MODADR  -- Display TSX.SAV module addresses
                                ;-----------------------------------------------------------
                                ;  MODADR is called to display the module address contained in TSX.SAV.
                                ;
001652                          MODADR:
001652                                  PRINT   #CRLF           ;Display CR/LF
001662  012737  0000000 000022'         MOV     #TSGEN,ADGEN    ;Save the TSGEN address
001670  012737  0000000 000024'         MOV     #TSTIO,ADTIO    ;Save the TSTIO address
001676  012737  0000000 000026'         MOV     #TSEXEC,ADEXEC  ;Save the TSEXEC address
001704  012737  0000000 000030'         MOV     #TSEMT,ADEMT    ;Save the TSEMT address
001712  012737  0000000 000032'         MOV     #TSINIT,ADINIT  ;Save the TSINIT address
001720  013705  0000000                 MOV     TIOBAS,R5       ;Get PAR of TSTIOX
001724  072527  000006                  ASH     #6,R5           ;Convert to address
001730  010537  000034'                 MOV     R5,ADTIOX       ;Get the TSTIOX address
001734  013705  0000000                 MOV     LOKBAS,R5       ;Save PAR of TSLOCK
001740  072527  000006                  ASH     #6,R5           ;Convert to address
001744  010537  000036'                 MOV     R5,ADLOCK       ;Save the TSLOCK address
001750  013705  0000000                 MOV     CSHBAS,R5       ;Get PAR of TSCASH
001754  072527  000006                  ASH     #6,R5           ;Convert to address
001760  010537  000040'                 MOV     R5,ADCASH       ;Save the TSCASH address
001764  012705  000100'                 MOV     #ADDR,R5        ;Point to address output list
001770  004737  002504'                 CALL    OLIST           ;Output address contents
001774  000207                          RETURN
```

```
                                        .SBTTL  OVRADR -- TSX-Plus overlay region
                                ;-----------------------------------------------------------------
                                ;  OVRADR is called to display the overlay region locations.
                                ;
                                ;  Overlay table structure:
                                ;
                                ;       OVRADD --> $OVTAB:
                                ;                       .WORD   <IDENTIFIER>,<KPAR5>,<WORD COUNT>
                                ;                      DUMMY SUBROUTINES FOR ALL OVERLAY SEGMENTS
                                ;
001776                          OVRADR:
001776                                  PRINT   #CRLF           ;Display CR/LF
002006                                  PRINT   #OVLREQ         ;Display header
002016  013705  000000@                 MOV     OVRADD,R5       ;Find the overlay table address
002022  012501                  10$:    MOV     (R5)+,R1        ;Get the overlay identifier
002024  004737  004022'                 CALL    OUTRAD          ;Display rad50 id
002030                                  PRINT   #EQSTR          ;Display " = "
002040  012501                          MOV     (R5)+,R1        ;Get the KPAR5 value
002042  005003                          CLR     R3              ;Default octal display value
002044  004737  003626'                 CALL    OUTVAL          ;Display KPAR5 value
002050  005725                          TST     (R5)+           ;
002052                                  PRINT   #CRLF           ;Display CR/LF
002062  021527  004537                  CMP     (R5),#4537      ;Compare with a <JSR R5,$OVRH> instruction
002066  001355                          BNE     10$             ;Br if not end of the table
002070  000207                          RETURN
```

```
                                        .SBTTL  STACK  -- Display contents of stack
                                ;-----------------------------------------------------------------
                                ;  STACK is called to display the contents of the stack.
                                ;
                                ;  There are three possible stack configurations.  The users system
                                ;  stack is stored in the each job's context region.  This stack
                                ;  is bounded by JSTK (top of stack) and JSTKND (bottom of stack).
                                ;  The interrupt stack is used during interrupt processing and is
                                ;  allocated over TSINIT code.  The address boundary of this stack
                                ;  is stored in INTSTK (top of stack) and INTSND (bottom of stack)
                                ;  located in TSEXEC.  The system stack is used during scheduling
                                ;  and is located between SS (top of stack) and SSEND (bottom of stack).
                                ;
002072                          STACK:  PRINT   #CRLF           ;Display CR/LF
002102  013705  0000000               MOV     DIESP,R5        ;Get the stack pointer
002106  010504                        MOV     R5,R4           ;Copy the base of the stack
002110  020527  0000000         5$:   CMP     R5,#SS          ;Below the top of the system stack?
002114  101013                        BHI     10$             ;No, SP not in system stack
                                ;
                                ;  Stack pointer is in the system stack area.
                                ;
002116  162704  0000000               SUB     #SS,R4          ;Calculate number of bytes to dump
002122  001465                        BEQ     50$             ;No stack space used
002124  005404                        NEG     R4              ;Convert to positive byte count
002126                                PRINT   #STKSS          ;Display "System Stack Contents"
002136  004737  002640'               CALL    OSTACK          ;Output the stack contents
002142  000207                        RETURN
002144  020537  0000000         10$:  CMP     R5,INTSTK       ;Below the top of the interupt stack?
002150  101013                        BHI     20$             ;No, SP not in interrupt stack
                                ;
                                ;  Stack pointer is in the interrupt stack area.
                                ;
002152  163704  0000000               SUB     INTSTK,R4       ;Calculate number of bytes to dump
002156  001447                        BEQ     50$             ;No stack space used
002160  005404                        NEG     R4              ;Convert to positive byte count
002162                                PRINT   #STKINT         ;Display "Interrupt Stack Contents"
002172  004737  002640'               CALL    OSTACK          ;Output the stack contents
002176  000207                        RETURN
002200  020527  0000000         20$:  CMP     R5,#JSTKND      ;Above the end of the context stack?
002204  103416                        BLO     30$             ;Invalid stack pointer
002206  020527  0000000               CMP     R5,#JSTK        ;Below the top of the context stack?
002212  101013                        BHI     30$             ;No, SP not in job context stack
                                ;
                                ;  Stack pointer is in the job's context stack.
                                ;
002214  162704  0000000         21$:  SUB     #JSTK,R4        ;Calculate number of bytes to dump
002220  001426                        BEQ     50$             ;No stack space used
002222  005404                        NEG     R4              ;Convert to positive byte count
002224                                PRINT   #STKEMT         ;Display "EMT Stack Contents"
002234  004737  002640'               CALL    OSTACK          ;Output the stack contents
002240  000207                        RETURN
                                ;
                                ;  Stack pointer is invalid.  Output 40 words where SP is pointing.
                                ;
002242                          30$:  PRINT   #INVSTK         ;Display "Invalid stack"
002252  020527  0000000               CMP     R5,#VPAR5       ;Check for EMT stack overflow
002256  103403                        BLO     40$             ;Br if below 120000, unknown stack
```

STACK -- Display contents of stack

```
     58 002260  012705  0000000                MOV     #VPAR6,R5       ;Adjust stack pointer to EMT stack
     59 002264  000753                          BR      21$             ;Dump job context stack
     60 002266  012704  000310          40$:    MOV     #100.*2,R4      ;Dump 100. words from stack pointer
     61 002272  004737  002640'                CALL    OSTACK          ;Output the stack contents
     62 002276  000207                  50$:    RETURN
```

```
                                        .SBTTL  MISC    -- Display misc system values
                                ;-----------------------------------------------------------------
                                ;  MISC is called to display the miscellaneous system values.
                                ;
002300                          MISC:
002300                                  PRINT   #CRLF                   ;Display CR/LF
002310  105737  0000000                 TSTB    INTLVL                  ;Check interrupt level
002314  002404                          BLT     10$                     ;Br if not executing interrupt
002316                                  PRINT   #INTLEV                 ;Display "Interrupt level"
002326  105737  0000000         10$:    TSTB    FRKPRI                  ;Check fork priority
002332  001404                          BEQ     20$                     ;Br if not executing fork
002334                                  PRINT   #FRKLEV                 ;Display "Fork level"
002344  023727  0000000 123456  20$:    CMP     SSEND,#123456           ;Check end of system stack
002352  001404                          BEQ     30$                     ;Br if stack end is valid
002354                                  PRINT   #STKOVR                 ;Display "System stack overflow"
002364  005002                  30$:    CLR     R2                      ;Clear offset pointer
002366  012705  000164'                 MOV     #VMISC,R5               ;Point to miscellaneous output list
002372  004737  002504'                 CALL    OLIST                   ;Output the list of values
002376  000207                          RETURN
```

```
                                                .SBTTL  JOB     -- Display info about current job
                                        ;-----------------------------------------------------------------
                                        ;  JOB is called to display information concerning the executing job.
                                        ;
002400                                  JOB:    PRINT   #CRLF           ;Display CR/LF
002410                                          PRINT   #CURJOB         ;Disp "Job executing at time of dump"
002420                                          PRINT   #EQSTR          ;Display " = "
002430  113701  0000000                         MOVB    CORUSR,R1       ;Get the current operating job index
002434  006201                                  ASR     R1              ;Divide by two for the job number
002436  012703  000002                          MOV     #$DEC,R3        ;Force decimal display
002442  004737  003626'                         CALL    OUTVAL          ;Output the job number
002446                                          PRINT   #CRLF           ;Display CR/LF
002456  000207                                  RETURN
```

```
                                        .SBTTL  LINE   -- Display line definition tables
                                ;-----------------------------------------------------------------
                                ;  LINE is called to display the generated line definition tables.
                                ;
002460  012705  000622'         LINE:   MOV     #JLIN,R5        ;Point to job line table
002464  004737  002714'                 CALL    OTABLE          ;Output line definition values
002470  000207                         RETURN
```

```
                                        .SBTTL  DEVICE -- Display device definition tables
                                ;-----------------------------------------------------------------
                                ;  DEVICE is called to display the generated device definition tables.
                                ;
002472  012705  001410'        DEVICE: MOV     #DHAN,R5        ;Point to device table
002476  004737  002714'                CALL    OTABLE          ;Output device table values
002502  000207                        RETURN
```

```
                                        .SBTTL  OLIST  -- Output a list of information
                                ;-----------------------------------------------------------------
                                ;  Output a list of information.
                                ;
002504  010546                 OLIST:  MOV     R5,-(SP)        ;Save registers
002506  010446                         MOV     R4,-(SP)
002510  010346                         MOV     R3,-(SP)
002512  010246                         MOV     R2,-(SP)
002514  010146                         MOV     R1,-(SP)
                                ;
                                ;  Display the list header
                                ;
002516  012501                         MOV     (R5)+,R1        ;Point to list header
002520  001437                         BEQ     100$            ;Br if not header
002522                                  PRINT                   ;Display the list header
002526  000434                         BR      100$            ;Output entire display list
                                ;
                                ;  Display identifier
                                ;
002530                          10$:    PRINT   (R5)            ;Print identifier
002536                          30$:    PRINT   #EQSTR          ;Point to string " = "
002546  016503  000002                 MOV     2(R5),R3        ;Get field attributes
002552  016504  000004                 MOV     4(R5),R4        ;Get the field address
002556  060204                         ADD     R2,R4           ;Add field bias
002560  032703  000001                 BIT     #$BYTE,R3       ;Word or byte storage element?
002564  001404                         BEQ     31$             ;Br if word element
002566  111401                         MOVB    @R4,R1          ;Get numeric value
002570  042701  177400                 BIC     #^C377,R1       ;Clear sign extend
002574  000401                         BR      32$             ;Continue
002576  011401                 31$:    MOV     @R4,R1          ;Get the numeric value
002600  004737  003626'         32$:    CALL    OUTVAL          ;Output the numeric value
002604                                  PRINT   #CRLF           ;Point to carriage return / line feed
002614  062705  000006         50$:    ADD     #6,R5           ;Increment to next list entry
                                ;
                                ;  Continue output of list elements.
                                ;
002620  005715                 100$:   TST     (R5)            ;Point to the next list entry
002622  001342                         BNE     10$             ;Br if list continues
                                ;
                                ;  Finished
                                ;
002624  012601                         MOV     (SP)+,R1        ;Restore registers
002626  012602                         MOV     (SP)+,R2
002630  012603                         MOV     (SP)+,R3
002632  012604                         MOV     (SP)+,R4
002634  012605                         MOV     (SP)+,R5
002636  000207                         RETURN                  ;Finished
```

```
                                        .SBTTL  OSTACK -- Display stack contents
                                ;-----------------------------------------------------------------
                                ;  Output the stack contents.
                                ;
002640                          OSTACK:
002640  006204                          ASR     R4              ;Convert byte count to word count
002642                                  PRINT   #STKCNT         ;Display header
002652  005003                          CLR     R3              ;Set for octal display
002654  010501                  10$:    MOV     R5,R1           ;Copy the stack address
002656  004737  003626'                 CALL    OUTVAL          ;Display stack address
002662                                  PRINT   #STKDLM         ;Display delimiter
002672  012501                          MOV     (R5)+,R1        ;Get the stack contents
002674  004737  003626'                 CALL    OUTVAL          ;Display stack contents
002700                                  PRINT   #CRLF           ;Display CR/LF
002710  077417                          SOB     R4,10$          ;Continue until done
002712  000207                          RETURN
```

```
      1                                       .SBTTL  OTABLE -- Output tables of information
      2                                ;-----------------------------------------------------------------
      3                                ;  Output tables of information.
      4                                ;
      5 002714                         OTABLE:
      6 002714  004737  002752'                CALL    TINIT           ;Initialize for table output
      7 002720  004737  003006'        1$:     CALL    TCLR            ;Clear output tables
      8 002724  004737  003156'                CALL    TACCR           ;Accrue table information
      9 002730  004737  003432'                CALL    TOUT            ;Display table values
     10 002734  005715                         TST     (R5)            ;Check for more table information
     11 002736  001370                         BNE     1$              ;Br if more to display
     12 002740                                  PRINT   #CRLF           ;Display CR/LF
     13 002750  000207                         RETURN
```

```
                                        ;
                                        ;  Initialize for display start or continuation of table format.
                                        ;
002752                          TINIT:  PRINT   #CRLF           ;Display CR/LF
002762                                  PRINT   (R5)+           ;Display the table title
002770  005037  000052'                 CLR     COL             ;Clear column counter
002774                                  PRINT   #CRLF           ;Display CR/LF
003004  000207                         RETURN
                                        ;
                                        ;  Clear table entries for continuation of table format.
                                        ;
003006  012702  000014          TCLR:   MOV     #NCOL,R2        ;Initilize table storage information
003012  006302                          ASL     R2              ;Convert to word index
003014  005062  000054'         10$:    CLR     TSCOL(R2)       ;Clear starting column number
003020  005062  000106'                 CLR     TECOL(R2)       ;Clear ending column number
003024  005062  000140'                 CLR     TSINX(R2)       ;Clear starting index number
003030  005062  000172'                 CLR     TEINX(R2)       ;Clear ending index number
003034  005062  000224'                 CLR     TATT(R2)        ;Clear attribute flags
003040  005062  000256'                 CLR     TADR(R2)        ;Clear address location
003044  162702  000002                 SUB     #2,R2           ;Subtract from table index
003050  003361                          BGT     10$             ;Continue for all entries
003052  002427                          BLT     20$             ;All entries are clear
003054  005737  000052'                 TST     COL             ;Check column counter
003060  001755                          BEQ     10$             ;Br if first table initialization call
                                        ;
                                        ;  Table continuation display.
                                        ;
003062                                  PRINT   #CRLF           ;Print CR/LF
003072                                  PRINT   #CRLF           ;Print CR/LF
003102                                  PRINT   TADR(R2)        ;Print columnar heading
003112  016201  000106'                 MOV     TECOL(R2),R1    ;Find ending column information
003116  004737  004144'                 CALL    BLKPAD          ;Blank fill
003122                                  PRINT   #SPSTR          ;Print blanks for spacing
003132  000207                  20$:    RETURN
```

```
                                        ;
                                        ;  Accrue information concerning table display while printing column headings.
                                        ;
003134                          NXTCOL:
003134  016562  000004  000256' MOV     4(R5),TADR(R2)  ;Store print string address
003142  062705  000012          ADD     #10.,R5         ;Offset to the next column entry
003146                          PRINT   #SPSTR          ;Print blanks for spacing
003156  062702  000002          TACCR:  ADD     #2,R2           ;Increment the field index
003162  020227  000014          CMP     R2,#NXCOL       ;End of columns allowed?
003166  002120                  BGE     50$             ;Br is no more column storage
003170  005715                  TST     (R5)            ;See if a column entry exists
003172  001516                  BEQ     50$             ;End of table found
003174  013762  000052' 000054' MOV     COL,TSCOL(R2)   ;Point to starting field position
003202                          PRINT   (R5)            ;Display the column heading
003210  013762  000052' 000106' MOV     COL,TECOL(R2)   ;Set ending column position
003216  016562  000002  000224' MOV     2(R5),TATT(R2)  ;Store field attributes
003224  016500  000006          MOV     6(R5),R0        ;Get the starting index number
003230  001002                  BNE     1$              ;Br if start is specified
003232  013700  000140'          MOV     TSINX,R0        ;Set as starting table number
003236  010062  000140'          1$:     MOV     R0,TSINX(R2)    ;Store starting index number
003242  016500  000010          MOV     10(R5),R0       ;Get the ending index number
003246  001002                  BNE     2$              ;Br is end is specified
003250  013700  000172'          MOV     TEINX,R0        ;Set as ending table number
003254  010062  000172'          2$:     MOV     R0,TEINX(R2)    ;Store ending index number
003260  012700  000006          MOV     #6,R0           ;Assume 6 digit (word element)
003264  032762  000001  000224' BIT     #$BYTE,TATT(R2) ;Check field attributes
003272  001402                  BEQ     10$             ;Br if word element
003274  012700  000003          MOV     #3,R0           ;Assume 3 digit (byte element)
003300  066200  000054'          10$:    ADD     TSCOL(R2),R0    ;Add starting position
003304  026200  000106'          CMP     TECOL(R2),R0    ;Determine if ending position
003310  001427                  BEQ     30$             ;Field is exactly enough
003312  101013                  BHI     20$             ;Br if field too long
003314  010001                  MOV     R0,R1           ;Copy ending column address
003316  166201  000106'          SUB     TECOL(R2),R1    ;Find difference
003322  010062  000106'          MOV     R0,TECOL(R2)    ;Force ending column longer
003326  112700  000040          MOVB    #' ,R0          ;Store blank
003332                          11$:    TTYOUT                  ;Pad output with blanks
003336  077103                  SOB     R1,11$          ;Continue until padded
003340  000413                  BR      30$             ;Go on to the next entry
003342  166200  000106'          20$:    SUB     TECOL(R2),R0    ;Subtract the ending column
003346  032762  040000  000224' BIT     #$RJUST,TATT(R2);Check for right justify
003354  001403                  BEQ     21$             ;Br if left justify
003356  160062  000054'          SUB     R0,TSCOL(R2)    ;Adjust starting column position
003362  000402                  BR      30$
003364  060062  000106'          21$:    ADD     R0,TECOL(R2)    ;Adjust ending column position
003370  026227  000106' 000120  30$:    CMP     TECOL(R2),#80.  ;Check for line overflow
003376  103656                  BLO     NXTCOL          ;Continue collecting columns
003400  005062  000054'          CLR     TSCOL(R2)       ;Clear partial entry
003404  005062  000106'          CLR     TECOL(R2)
003410  005062  000140'          CLR     TSINX(R2)
003414  005062  000172'          CLR     TEINX(R2)
003420  005062  000224'          CLR     TATT(R2)
003424  005062  000256'          CLR     TADR(R2)
003430  000207                  50$:    RETURN
```

```
                                        ;
                                        ;  Display index value and data.
                                        ;
003432  010446                  TOUT:   MOV     R4,-(SP)        ;Save some registers
003434  010546                          MOV     R5,-(SP)
                                        ;
                                        ;  Display index offset into table.
                                        ;
003436  013705  000140'                 MOV     TSINX,R5        ;Find the starting index
003442                          1$:     PRINT   #CRLF           ;Display CR/LF
003452  010501                          MOV     R5,R1           ;Copy index number
003454  005002                          CLR     R2              ;Clear column index
003456  016203  000224'                 MOV     TATT(R2),R3     ;Numeric attribute
003462  004737  003626'                 CALL    OUTVAL          ;Display index value
003466  000430                          BR      40$             ;Enter column output stream
                                        ;
                                        ;  Determine if the value should be displayed.
                                        ;
003470  020562  000140'         10$:    CMP     R5,TSINX(R2)    ;Is the current index < starting
003474  002425                          BLT     40$             ;Yes, skip entry
003476  020562  000172'                 CMP     R5,TEINX(R2)    ;Is the current index > ending
003502  003022                          BGT     40$             ;Yes, skip entry
                                        ;
                                        ;  Output rows of information.
                                        ;
003504  004737  004144'                 CALL    BLKPAD          ;Pad with spaces
003510  016203  000224'                 MOV     TATT(R2),R3     ;Get the attribute
003514  016201  000256'                 MOV     TADR(R2),R1     ;Check for existing entry
003520  060501                          ADD     R5,R1           ;Add the index offset (byte)
003522  032703  000001                  BIT     #$BYTE,R3       ;Check for byte storage
003526  001404                          BEQ     20$             ;Br if word storage
003530  111101                          MOVB    (R1),R1         ;Get the stored byte value
003532  042701  177400                  BIC     #^C377,R1       ;Kill sign extend bits
003536  000402                          BR      30$             ;Continue
003540  060501                  20$:    ADD     R5,R1           ;Add the index offset (word)
003542  011101                          MOV     (R1),R1         ;Get the stored word value
003544  004737  003626'         30$:    CALL    OUTVAL          ;Output the value
003550  062702  000002          40$:    ADD     #2,R2           ;Increment the index
003554  016201  000054'                 MOV     TSCOL(R2),R1    ;Get the starting column
003560  001343                          BNE     10$             ;End of table entries
003562  005205                          INC     R5              ;Increment the index offset
003564  020537  000172'                 CMP     R5,TEINX        ;Check for ending index
003570  101724                          BLOS    1$              ;Br if more tables to output
003572  012605                          MOV     (SP)+,R5        ;Restore registers
003574  012604                          MOV     (SP)+,R4
003576  000207                          RETURN
```

```
                                        .SBTTL  OUTSTR -- Output an asciz string
                                ;-----------------------------------------------------------------
                                ;  OUTSTR - Output an asciz string.
                                ;
                                ;  Arguments --
                                ;       R1      =       String address
                                ;

003600                          OUTSTR:
003600  112100                  1$:     MOVB    (R1)+,R0        ;Get next output byte
003602  001404                          BEQ     10$             ;Zero, output CR / LF
003604  100407                          BMI     20$             ;Exit on end of output string
003606                                  TTYOUT                  ;Output the character
003612  000772                          BR      1$              ;Contine until end of string
003614                          10$:    PRINT   #CRLF           ;Display carriage return / line feed
003624  000207                  20$:    RETURN
```

```
                                        .SBTTL  OUTVAL -- Convert and print a value
                                ;-----------------------------------------------------------------
                                ;  OUTVAL - Convert and output a value.
                                ;
                                ;  Arguments -
                                ;       R1      = Numeric value
                                ;       R3      = Display string attributes
                                ;
003626  010546          OUTVAL: MOV     R5,-(SP)        ;Save registers
003630  010446                  MOV     R4,-(SP)
003632  010346                  MOV     R3,-(SP)
003634  010246                  MOV     R2,-(SP)
003636  032703  000004          BIT     #$RAD50,R3      ;Check for RAD50 value
003642  001403                  BEQ     1$              ;Br if not RAD50
003644  004737  004022'          CALL    OUTRAD          ;Output RAD50
003650  000420                  BR      30$             ;Continue
003652  012704  000010  1$:     MOV     #10,R4          ;Assume octal base
003656  032703  000002          BIT     #$DEC,R3        ;Check for decimal base
003662  001402                  BEQ     10$             ;Br if numeric display in octal
003664  012704  000012          MOV     #10.,R4         ;Set for decimal base
003670  012702  000006  10$:    MOV     #6,R2           ;Assume word value
003674  032703  000001          BIT     #$BYTE,R3       ;Check for byte storage
003700  001402                  BEQ     20$             ;Br if storage is word value
003702  012702  000003          MOV     #3,R2           ;Otherwise byte value
003706  004737  003724'  20$:    CALL    CONVAL          ;Convert numeric value
003712  012602  30$:    MOV     (SP)+,R2        ;Restore registers
003714  012603                  MOV     (SP)+,R3
003716  012604                  MOV     (SP)+,R4
003720  012605                  MOV     (SP)+,R5
003722  000207                  RETURN
                                ;
003724                  CONVAL:
003724  005302                  DEC     R2              ;Say one more digit converted
003726  002411                  BLT     2$              ;Br if all display digits converted
003730  005000                  CLR     R0              ;Clear high order result
003732  071004                  DIV     R4,R0           ;Divide number by numeric base
003734  062701  000060          ADD     #60,R1          ;Convert remainder to ascii digit
003740  010146                  MOV     R1,-(SP)        ;Save ascii digit on stack
003742  010001                  MOV     R0,R1           ;Set result register
003744  001406                  BEQ     3$              ;Br if no more digits to convert
003746  004737  003724'          CALL    CONVAL          ;Call conversion for next digit
003752  112600  2$:     MOVB    (SP)+,R0        ;Get the next digit to display
003754                  TTYOUT                  ;Output the ascii character
003760  000207                  RETURN
003762  005702  3$:     TST     R2              ;Check field output size
003764  001772                  BEQ     2$              ;No pad necessary
003766  032703  040000          BIT     #$RJUST,R3      ;Check for right justify
003772  001767                  BEQ     2$              ;Br if not right justify
003774  112700  000040          MOVB    #' ,R0          ;Default pad to blank
004000  032703  020000          BIT     #$ZRPAD,R3      ;Check for zero pad
004004  001407                  BEQ     4$              ;Br if blank pad chosen
004006  112700  000060          MOVB    #'0,R0          ;Change to zero pad
004012  004016  077203  4$:     TTYOUT                  ;Output the charater
004020  000754                  SOB     R2,4$           ;Continue until all ouput
                                BR      2$              ;Display remaining characters
```

```
                                        .SBTTL  OUTRAD -- Convert and print a RAD50 value
                                ;-----------------------------------------------------------------
                                ;  OUTRAD - Convert and output rad50 data.
                                ;       R1      = Numeric rad50 value
                                ;
 004022  010146                 OUTRAD: MOV     R1,-(SP)
 004024  005000                         CLR     R0              ;Clear high order
 004026  071027  003100                 DIV     #50*50,R0       ;Divide for 1st byte
 004032                                 TTYOUT  R50CHR(R0)      ;Display rad50 character
 004042  005000                         CLR     R0              ;Clear high order
 004044  071027  000050                 DIV     #50,R0          ;Divide for 2nd byte
 004050                                 TTYOUT  R50CHR(R0)      ;Display rad50 character
 004060                                 TTYOUT  R50CHR(R1)      ;Display rad50 character
 004070  012601                         MOV     (SP)+,R1
 004072  000207                         RETURN
                                ;
 004074      040     101     102 R50CHR: .ASCII  / ABCDEFGHIJKLMNOPQRSTUVWXYZ$. 0123456789/
     004077      103     104     105
     004102      106     107     110
     004105      111     112     113
     004110      114     115     116
     004113      117     120     121
     004116      122     123     124
     004121      125     126     127
     004124      130     131     132
     004127      044     056     040
     004132      060     061     062
     004135      063     064     065
     004140      066     067     070
     004143      071
                                        .EVEN

                                        .SBTTL  BLKPAD -- Blank pad output stream
                                ;-----------------------------------------------------------------
                                ;  BLKPAD - Blank pad ouput stream.
                                ;       R1      = ending column number
                                ;
 004144  112700  000040         BLKPAD: MOVB    #' ,R0          ;Initialize ouput to space
 004150  163701  000052'               SUB     COL,R1          ;Figure number of blanks
 004154  003403                         BLE     20$             ;Br if already past column
 004156                         10$:    TTYOUT                  ;Display space
 004162  077103                         SOB     R1,10$          ;Continue
 004164  000207                 20$:    RETURN
```

```
                                        .SBTTL  OUTCHR -- Output an ascii character
                                ;-----------------------------------------------------------------
                                ;  OUTCHR - Output an ascii character to a serial or parallel port.
                                ;       R0      = character to output
                                ;
004166  005737  000050'         OUTCHR: TST     RCSR            ;Check for receiver buffers
004172  001415                          BEQ     OPRINT          ;No receiver CSR, transmit character
004174                          OTERM:
004174  105777  173650          10$:    TSTB    @RCSR           ;Check input port
004200  100006                          BPL     20$             ;Br if no input pending
004202  117737  173640  000340'         MOVB    @RBUF,XFLG      ;Input the character
004210  142737  177600  000340'         BICB    #^C177,XFLG     ;Strip eighth bit
004216  123727  000340' 000021  20$:    CMPB    XFLG,#CTRLQ     ;Check for XON
004224  001363                          BNE     10$             ;Br if not XON state active
004226                          OPRINT:
004226  105777  173612          30$:    TSTB    @TCSR           ;Check transmit done flag
004232  100375                          BPL     30$             ;Wait for transmit done
004234  110077  173602                  MOVB    R0,@TBUF        ;Output to transmitter
004240  120027  000015                  CMPB    R0,#CR          ;Is character a carriage return?
004244  001003                          BNE     40$             ;Br if not carriage return
004246  005037  000052'                 CLR     COL             ;Clear column counter
004252  000405                          BR      50$             ;Finished
004254  120027  000040          40$:    CMPB    R0,#40          ;Is character printable?
004260  103402                          BLO     50$             ;Br if not printable
004262  005237  000052'                 INC     COL             ;Increment column position
004266  000207                  50$:    RETURN                  ;Finished

        000001                          .END
```

Errors detected: 0

*** Assembler statistics

Work file reads: 0
Work file writes: 0
Size of work file: 9623 Words (38 Pages)
Size of core pool: 17920 Words (70 Pages)
Operating system: RT-11

Elapsed time: 00:00:58.48
DK:TSDUMP,LP:TSDUMP=DK:TSDUMP.MAC/C/N:SYM

Cross reference table (CREF V05.04)

```
$BYTE    5-22#   9-10    9-11    9-12    9-13    9-17    9-18    9-19    9-24    9-25    9-27    9-30
         9-31    9-32    9-33    9-34    9-35    9-37    9-38    9-42    9-43    9-50    9-52    10-5
        11-5    25-25   29-26   30-31   32-22
$DEC     5-23#   10-5    11-5    14-11   14-24   22-10   32-18
$RAD50   5-24#   10-25   10-26   11-6    14-18   32-13
$RJUST   5-19#   10-6    10-7    10-8    10-9    10-10   10-11   10-12   10-13   10-14   10-15   10-16
        10-17   10-18   10-19   10-20   10-21   10-22   10-23   10-24   10-27   10-28   10-29   10-30
        10-31   10-32   10-33   10-34   10-35   10-36   10-37   10-38   10-39   10-40   10-41   11-7
        11-8    11-9    11-10   11-11   11-12   11-13   29-41   32-47
$ZRPAD   5-20#   10-6    10-7    10-8    10-9    10-10   10-11   10-12   10-13   10-14   10-15   10-16
        10-17   10-18   10-19   10-20   10-21   10-22   10-23   10-24   10-27   10-28   10-29   10-30
        10-31   10-32   10-33   10-34   10-35   10-36   10-37   10-38   10-39   10-40   10-41   11-11
        11-12   11-13   32-50
ADCASH   8-13    8-13    12-36#  18-20*
ADDR     8-5#    18-21
ADEMT    8-9     8-9     12-32#  18-10*
ADEXEC   8-8     8-8     12-31#  18-9*
ADGEN    8-6     8-6     12-29#  18-7*
ADINIT   8-10    8-10    12-33#  18-11*
ADLOCK   8-12    8-12    12-35#  18-17*
ADTIO    8-7     8-7     12-30#  18-8*
ADTIOX   8-11    8-11    12-34#  18-14*
ARG     13-61   13-61#   14-5    14-5#   14-13   14-13#  14-20   14-20#  14-26   14-26#  14-43   14-43#
        14-48   14-48#   14-50   14-50#   14-52   14-52#  16-5    16-5#   16-6    16-6#   16-7    16-7#
        16-11   16-11#   16-14   16-14#   16-17   16-17#  16-20   16-20#  16-23   16-23#  17-5    17-5#
        18-6    18-6#    19-12   19-12#   19-13   19-13#  19-17   19-17#  19-22   19-22#  20-14   20-14#
        20-25   20-25#   20-36   20-36#   20-49   20-49#  20-55   20-55#  21-6    21-6#   21-9    21-9#
        21-12   21-12#   21-15   21-15#   22-5    22-5#   22-6    22-6#   22-7    22-7#   22-12   22-12#
        25-15   25-15#   25-20   25-20#   25-21   25-21#  25-32   25-32#  26-7    26-7#   26-11   26-11#
        26-14   26-14#   27-12   27-12#   28-4    28-4#   28-5    28-5#   28-7    28-7#   28-28   28-28#
        28-29   28-29#   28-30   28-30#   28-33   28-33#  29-7    29-7#   29-14   29-14#  29-37   29-37#
        30-11   30-11#   31-13   31-13#   31-15   31-15#  32-43   32-43#  32-53   32-53#  33-9    33-9#
        33-12   33-12#   33-13   33-13#   33-28   33-28#
ARGS     7-5     7-5#    8-5     8-5#    9-5     9-5#    10-5    10-5#   11-5    11-5#
BLKPAD  28-32   30-27   33-25#
CLKPC    9-6     9-6
CLKPS    9-7     9-7
COL     12-46#  28-6*   28-23   29-13   29-15   33-26   34-21*  34-25*
CONVAL  32-25   32-32#  32-41
CORUSR   2-14    22-8
CR      12-7#   12-75   34-19
CRLF    12-75#   14-5    14-52   16-5    16-11   16-17   16-23   17-5    18-6    19-12   19-22   20-14
        21-6    22-5    22-12   25-32   26-14   27-12   28-4    28-7    28-28   28-29   30-11   31-15
CSHBAS   9-8     9-8     18-18
CTRLQ   12-9#   12-63   34-13
CTRLS   12-10#
CURFRK   9-9     9-9
CURJOB  12-86#   22-6
CXBJOB   9-10    9-10
CXBOWN   9-11    9-11
DASH    12-11#
DATTIM  13-62    14-5#
DEVICE  13-70    24-5#
DEVSIZ  11-13    11-13
DHAN    11-5#    24-5
DIEARG   2-11    16-8
```

Cross reference table (CREF V05.04)

DIEPC	2-11	7-13	7-13	
DIESP	2-11	7-12	7-12	20-15
DIV32	14-33	14-37	15-8#	
DMPHND	2-11	16-18		
DMPOVL	2-11	16-12		
DMPTRP	13-32	13-76#		
DMPTXT	2-11	16-6		
DODUMP	2-4	13-27#		
DOSCHD	9-12	9-12		
DVFLAG	11-11	11-11		
DVSTAT	11-12	11-12		
EQSTR	12-72#	19-17	22-7	25-21
ERROR	13-63	16-5#		
EXCJOB	9-13	9-13		
FF	12-74#	13-61		
FREFRK	9-16	9-16		
FREIOQ	9-14	9-14		
FRKCQE	9-15	9-15		
FRKLEV	12-84#	21-12		
FRKPRI	9-17	9-17	21-10	
HANENT	11-7	11-7		
HANIOC	11-10	11-10		
HANPAR	11-8	11-8		
HANSIZ	11-9	11-9		
INBSY	9-18	9-18		
INTLEV	12-83#	21-9		
INTLVL	9-19	9-19	21-7	
INTPRI	9-20	9-20		
INTSND	2-9			
INTSTK	2-9	20-28	20-33	
INVSTK	12-82#	20-55		
IOABFL	9-21	9-21		
JLIN	10-5#	23-5		
JOB	13-64	22-5#		
JSTK	2-9	20-41	20-46	
JSTKND	2-9	20-39		
KPAR6	2-13	13-51		
LACTIV	10-24	10-24		
LAFSIZ	10-34	10-34		
LBASE	10-18	10-18		
LBSPRI	10-30	10-30		
LCDTYP	10-27	10-27		
LCLUNT	10-31	10-31		
LCMPL	10-36	10-36		
LEMTPC	10-39	10-39		
LF	12-8#	12-75		
LFWLIM	10-35	10-35		
LINCNT	10-23	10-23		
LINE	13-69	23-5#		
LIOCNT	10-22	10-22		
LIOHLD	10-33	10-33		
LJSW	10-38	10-38		
LMEMIN	10-32	10-32		
LNBLKS	10-20	10-20		
LNPRIM	10-28	10-28		
LNSBLK	10-21	10-21		

Cross reference table (CREF V05.04)

```
LOKBAS    9-22      9-22     18-15
LOKSWP    9-23      9-23
LPARBS   10-19     10-19
LPARNT   10-29     10-29
LPRG1    10-25     10-25
LPRG2    10-26     10-26
LSPND    10-40     10-40
LSTATE   10-17     10-17
LSW      10-6      10-6
LSW10    10-15     10-15
LSW11    10-16     10-16
LSW2     10-7      10-7
LSW3     10-8      10-8
LSW4     10-9      10-9
LSW5     10-10     10-10
LSW6     10-11     10-11
LSW7     10-12     10-12
LSW8     10-13     10-13
LSW9     10-14     10-14
LSWPBK   10-37     10-37
LXCL     10-41     10-41
MAPUSR    9-24      9-24
MAXDEV    2-14     11-5
MEM256    9-25      9-25
MEMSWP    9-26      9-26
MIOFLG    9-27      9-27
MISC     13-68     21-5#
MODADR   13-66     18-5#
MONTH    12-59#    14-17
MSGBAS    9-28      9-28
NCOL     12-12#    12-50     12-51     12-52     12-53     12-54     12-55     28-12     29-9
NDL       2-15     10-28     10-31
NIOL      2-15     10-31
NLINES    2-15     10-5
NMFREQ    9-29      9-29
NPL       2-15     10-28     10-31     10-41
NSL       2-15     10-31
NUMDEV    2-14
NUMON     9-30      9-30
NXTCOL   29-4#     29-47
OLIST    17-8      18-22     21-18     25-5#
OPRINT   34-7      34-15#
OSTACK   20-26     20-37     20-50     20-61     26-5#
OTABLE   23-6      24-6      27-5#
OTERM    34-8#
OUTBSY    9-31      9-31
OUTCHR   14-13     14-20     14-43     14-48     14-50     29-37     31-13     32-43     32-53     33-9      33-12     33-13
         33-28     34-6#
OUTRAD   14-19     16-16     16-22     19-16     32-15     33-6#
OUTSTR   13-61     14-5      14-26     14-52     16-5      16-6      16-7      16-11     16-14     16-17     16-20     16-23
         17-5      18-6      19-12     19-13     19-17     19-22     20-14     20-25     20-36     20-49     20-55     21-6
         21-9      21-12     21-15     22-5      22-6      22-7      22-12     25-15     25-20     25-21     25-32     26-7
         26-11     26-14     27-12     28-4      28-5      28-7      28-28     28-29     28-30     28-33     29-7      29-14
         30-11     31-9#     31-15
OUTVAL   14-12     14-25     16-10     19-20     22-11     25-31     26-10     26-13     30-15     30-38     32-9#
OVLREG   12-76#    19-10
```

Cross reference table (CREF V05.04)

```
OVRADD    2-19      19-14
OVRADR   13-67      19-11#
PNAME    11-6       11-6
PRTREG    7-5#      17-7
PSW       2-10
PVON      9-32       9-32
R50CHR   33-9       33-12     33-13     33-17#
RBUF     12-44#     13-39#    13-40*    13-41     34-11
RCSR     12-45#     13-38#    34-6      34-9
REG      13-65      17-5#
SPDJOB    9-33       9-33
SPSTR    12-73#     14-26     28-33     29-7
SR3FLG    9-34       9-34
SS        2-9       20-17     20-22
SSEND     2-9       21-13
STACK    13-71      20-14#
STKCNT   12-80#     26-7
STKDLM   12-81#     26-11
STKEMT   12-79#     20-49
STKINT   12-78#     20-36
STKLVL    9-35       9-35
STKOVR   12-85#     21-15
STKSS    12-77#     20-25
STRADD    7-5        7-5#      7-6       7-6#      7-7       7-7#      7-8       7-8#      7-9       7-9#      7-10      7-10#
          7-11       7-11#     7-12      7-12#     7-13      7-13#     7-14      7-14#     7-15      7-15#     8-5       8-5#
          8-6        8-6#      8-7       8-7#      8-8       8-8#      8-9       8-9#      8-10      8-10#     8-11      8-11#
          8-12       8-12#     8-13      8-13#     9-5       9-5#      9-6       9-6#      9-7       9-7#      9-8       9-8#
          9-9        9-9#      9-10      9-10#     9-11      9-11#     9-12      9-12#     9-13      9-13#     9-14      9-14#
          9-15       9-15#     9-16      9-16#     9-17      9-17#     9-18      9-18#     9-19      9-19#     9-20      9-20#
          9-21       9-21#     9-22      9-22#     9-23      9-23#     9-24      9-24#     9-25      9-25#     9-26      9-26#
          9-27       9-27#     9-28      9-28#     9-29      9-29#     9-30      9-30#     9-31      9-31#     9-32      9-32#
          9-33       9-33#     9-34      9-34#     9-35      9-35#     9-36      9-36#     9-37      9-37#     9-38      9-38#
          9-39       9-39#     9-40      9-40#     9-41      9-41#     9-42      9-42#     9-43      9-43#     9-44      9-44#
          9-45       9-45#     9-46      9-46#     9-47      9-47#     9-48      9-48#     9-49      9-49#     9-50      9-50#
          9-51       9-51#     9-52      9-52#     10-5      10-5      10-5      10-5#     10-5#     10-6      10-6#     10-7
         10-7#      10-8      10-8#     10-9      10-9#     10-10     10-10#    10-11     10-11#    10-12     10-12#    10-13
         10-13#     10-14     10-14#    10-15     10-15#    10-16     10-16#    10-17     10-17#    10-18     10-18#    10-19
         10-19#     10-20     10-20#    10-21     10-21#    10-22     10-22#    10-23     10-23#    10-24     10-24#    10-25
         10-25#     10-26     10-26#    10-27     10-27#    10-28     10-28#    10-29     10-29#    10-30     10-30#    10-31
         10-31#     10-32     10-32#    10-33     10-33#    10-34     10-34#    10-35     10-35#    10-36     10-36#    10-37
         10-37#     10-38     10-38#    10-39     10-39#    10-40     10-40#    10-41     10-41#    11-5      11-5      11-5
         11-5#      11-5#     11-6      11-6#     11-7      11-7#     11-8      11-8#     11-9      11-9#     11-10     11-10#
         11-11      11-11#    11-12     11-12#    11-13     11-13#
SVPAR5    7-14       7-14     12-24#    13-52*
SVPAR6    7-15       7-15     12-25#    13-51*
SVR0      7-6        7-6      12-17#
SVR1      7-7        7-7      12-18#
SVR2      7-8        7-8      12-19#
SVR3      7-9        7-9      12-20#
SVR4      7-10       7-10     12-21#
SVR5      7-11       7-11     12-22#
SVREG    12-23#     13-53
SYSDAT    2-16       14-6
SYTIMH    2-17       14-30
SYTIML    2-17       14-31
TACCR    27-8       29-8#
```

Cross reference table (CREF V05.04)

```
TADR     12-55#  28-19#  28-30   29-5*   29-53*  30-29
TATT     12-54#  28-18#  29-16*  29-26   29-41   29-52*  30-14   30-28
TBUF     12-42#  13-46#  13-47*  34-18*
TCLR     27-7    28-12#
TCSR     12-43#  13-45#  34-16
TECOL    12-51#  28-15#  28-31   29-15*  29-30   29-34   29-35*  29-40   29-45*  29-46   29-49*
TEINX    12-53#  28-17#  29-23   29-24*  29-51*  30-22   30-43
TENABL    7-5#    7-6     7-7     7-8     7-9     7-10    7-11    7-12    7-13    7-14    7-15    8-5#
          8-6     8-7     8-8     8-9     8-10    8-11    8-12    8-13    9-5#    9-6     9-7     9-8
          9-9     9-10    9-11    9-12    9-13    9-14    9-15    9-16    9-17    9-18    9-19    9-20
          9-21    9-22    9-23    9-24    9-25    9-26    9-27    9-28    9-29    9-30    9-31    9-32
          9-33    9-34    9-35    9-36    9-37    9-38    9-39    9-40    9-41    9-42    9-43    9-44
          9-45    9-46    9-47    9-48    9-49    9-50    9-51    9-52    10-5#   10-6    10-7    10-8
         10-9    10-10   10-11   10-12   10-13   10-14   10-15   10-16   10-17   10-18   10-19   10-20
         10-21   10-22   10-23   10-24   10-25   10-26   10-27   10-28   10-29   10-30   10-31   10-32
         10-33   10-34   10-35   10-36   10-37   10-38   10-39   10-40   10-41   11-5#   11-6    11-7
         11-8    11-9    11-10   11-11   11-12   11-13
TINIT    27-6    28-4#
TIOBAS    9-36    9-36   18-12
TK1SEC    2-18   14-32
TNHL      2-15   10-27
TOTON     9-37    9-37
TOUT     27-9    30-5#
TRPAR5    2-13   13-52
TSCOL    12-50#  26-14*  29-13*  29-29   29-43*  29-48*  30-40
TSEMT     2-12   18-10
TSEXEC    2-12   18-9
TSGEN     2-12   18-7
TSINIT    2-12   16-11
TSINX    12-52#  28-16*  29-19   29-20*  29-50*  30-10   30-20
TSTIO     2-12   18-8
TXARG    12-68#  16-7
TXDEV    12-71#  16-20
TXOID    12-70#  16-14
TXSEG    12-69#
UBUSMP    9-38    9-38
UIOCNT    9-39    9-39
USP       9-40    9-40
USRBAS    9-41    9-41
USRJOB    9-42    9-42
VBUSTP    9-43    9-43
VCSHNB    9-44    9-44
VDMTCR    2-10   13-34
VMISC     9-5#   21-17
VMXSF     9-45    9-45
VMXSFC    9-46    9-46
VNGR      9-47    9-47
VNUMDC    9-48    9-48
VPAR5     2-13   20-06
VPAR6     2-13   20-08
VPLAS     9-49    9-49
VSWPFL    9-50    9-50
VSWPSL    9-51    9-51
VUXIFL    9-52    9-52
XFLG     12-63#  34-11*  34-12*  34-13
```

Cross reference table (CREF V05.04)