

```

1          .TITLE  TSGEN -- System Generation Parameters
2          .IDENT  /V6.40/
3 000000   .CSECT  TSGEN
4          .ENABL  LC
5          .DSABL  GBL
6          .NLIST  CND
7          ;-----
8          ; TSGEN *** PRO/TSX-Plus *** version 6.40
9          ;
10         ; This module contains the the definitions of system parameters
11         ; that define the characteristics of the TSX-Plus system
12         ; being generated.
13         ;
14         ; Written by Phil Sherrod.
15         ;
16         ; Copyright (c) 1978,1979,1980,1981,1982,1983,1984,1985,1986,1987,1988,1989.
17         ; S&H Computer Systems, Inc.
18         ; 1027 17th. Avenue South
19         ; Nashville, Tennessee U. S. A.
20         ; (615) 327-3670
21         ;
22         ; This software is furnished under a license for use only on a
23         ; single computer system and may be copied only with the inclusion
24         ; of the above copyright notice.
25         ; This software, or any copies thereof, may not be provided
26         ; or otherwise made available to any other person except for
27         ; use on such system and to one who agrees to these license terms.
28         ; Title to and ownership of the software shall at all times remain
29         ; with S&H Computer Systems, Inc.
30         ; This software is the valuable property of S&H Computer Systems, Inc.
31         ; All rights reserved.
32         ;
33         ; S&H will seek legal redress for any unauthorized use of this product.
34         ;
35         ;
36         ; Set FULLST to 1 for a full assembly listing.
37         ; Set FULLST to 0 for a normal short listing.
38         ;
39         ; FULLST =      1
40         ;
41         ;
42 000000   TSGEN:
43         ;
44         ; Global definitions
45         ;
46         .GLOBL  MXTTCT, HANDSK, HANRCB, HANRCO, KMONCE, EXCJOB, DOSCHD
47         .GLOBL  IOHANG, GETUMR, FREUMR, SYPSWD, SYSPSR, TSXVRS, VMNUAO
48         .GLOBL  TSGEN, RDBSEC, DTYPE, NAMSEC, LNMTOP, SASECT, SMONHD
49         .GLOBL  NPL, NSL, NLIN2, LSTHL, TNHL, NIOL, FSTIOL
50         .GLOBL  NLINES, NLCHN, $MEMSZ, MEMPTR, PHYMEM, CFSTS, CFABLV
51         .GLOBL  VNGR, SHRRCB, SHRRCN, SCPFHD, VNUIP, INSTBL, INSTBN
52         .GLOBL  SPLND, SPLNF, SPLNB, VU$CL, UCLNAM, MIOBHD, DETCBS
53         .GLOBL  SPLBHD, SFCBFH, KMNTOP, KMNHI, NESB, FASTIN
54         .GLOBL  SDCB, SDCBND, SPLDEV, SPLDVN, UKMNAM, IOHSTM
55         .GLOBL  NFRESB, SPLBLK, NSPLDV, VSWPSL, CLEOFS, VMXWIN
56         .GLOBL  INVEC, RSR, RBR, TSR, TTCSCH, VSLEDT, FRKINI, FRKGEN
57         .GLOBL  TBR, INRECV, OTRECV, LSTPL, LUNAME, LSTIOL, NUMFRK, NXIVMH
58         .GLOBL  QUAN1, QUAN2, LSTMX, R$TTOP, R$INST, LSTLIN, R$CFST

```

59	. GLOBL	INMXV, OTMXV, PVSPBL, LMXLN, LMXPRM, NDVRCB
60	. GLOBL	OTRASZ, LNMAP, LNPRIM, LPRI, VLDSYS, NSPLFL, NSPLBL
61	. GLOBL	CLVERS, CLORSZ, VMXMON, MONFGH, SPSTAT, LDVERS
62	. GLOBL	NCSILO, NCXOFF, NCXON, CSHALC, VUXIFL
63	. GLOBL	LSTSL, FSTSL, CTRLTT, VINABT, IOABFL
64	. GLOBL	MAXSEC, VEDIT, WILDFL, VPRIVR, CLXTRA, CLTOTL
65	. GLOBL	SWDBLK, TMIOL, TMIOH, TK1CNT, TK1SEC, NSCP
66	. GLOBL	TMSWPL, TMSWPH, TMIOWL, TMIOWH, VCSHNB
67	. GLOBL	DCTOTU, DCTRD, DCCRD, DCTWR, DCCWR
68	. GLOBL	SFCB, SFCBND, QUAN1A, LOGCHN, QUAN1B
69	. GLOBL	TSXSIT, CFCHAN, NUCHN, MVSIZ, VQUN1C, VNRFLG
70	. GLOBL	VDISPC, VDOSPC, VDMKTP, VSYDMP, VDMTCR, MODDAT
71	. GLOBL	VNCSLO, VNCXOF, VNCXON, VCTRLT
72	. GLOBL	CMFSEC, CMFTOP, VKEYMX, VSCHED, VDSKBU, MODTIM
73	. GLOBL	VTSLCH, MXSPAC, VPMSIZ, VVLSCH, VQUANO, VQUAN3, VVPWCH
74	. GLOBL	MXRBUF, MXRING, MXBRK, MH\$SCR, MH\$RCR, MH\$CAR
75	. GLOBL	MH\$BCR, MH\$BAR, MH\$SSR, DHBFSZ, VCXTRM, VCXCTL
76	. GLOBL	VH\$CSR, VH\$DBR, VH\$LPR, VH\$LSR, RUNCHN
77	. GLOBL	VH\$LCR, VH\$BA1, VH\$BA2, VH\$BCR, CDX\$PI
78	. GLOBL	CDX\$DL, CDX\$DZ, CDX\$DH, CDX\$VH, CDX\$PC, CDX\$PP
79	. GLOBL	VMSCHR, VMIOSZ, RPRCSR, RPRVEC, DWTYPE, CDX\$QP
80	. GLOBL	VMAXMC, VMXMSG, MSGBAS, USRBAS, VMXMRB, WINBAS
81	. GLOBL	NUMDCD, VNUMDC, DCAGE, NUMCDB, VMIOBF
82	. GLOBL	LOKBAS, LOKMEM, TIOBAS, LOKCSH
83	. GLOBL	VQUAN1, VQUN1A, VQUN1B, VQUAN2, VCORTM
84	. GLOBL	MAPPAR, RDB, RDBEND, NUMRDB, MIODBG
85	. GLOBL	CSHDEV, CSHDEV, VMXCSH, VNFCSH, CSHHD
86	. GLOBL	MAXALC, ALCTBL, ALCEND, SNDBX, USPLCH, UBUSMP, MEM256
87	. GLOBL	VOFFTM, VONTM, VTMIN, VTMLQC, VTMOU
88	. GLOBL	FREFRK, BOTDEV, BOTUNI, BOTCSR, MIOFLG
89	. GLOBL	AUTHAN, AHEND, SNBUX, UMSYTP, VINTIO
90	. GLOBL	VMXSF, VMXSFC, VMLBLK, RSFBLK, VPLAS, SEGCHN
91	. GLOBL	SDCBSZ, SDBULS, SPLANM, MIONWB, INDFIL
92	. GLOBL	NDL, LSTD, FSTD, VSWPFL, MAPUSR, R\$MFMV
93	. GLOBL	CORUSR, LINNUM, MUXNUM, NUMON, PVON, TOTON
94	. GLOBL	STPFLG, MINTIM, RTVECT, VHIPCT, FRKDLY
95	. GLOBL	SYSCHN, SYCHO, SYCH1, SYCH2, SYCH3, SYCH4, SYCH5
96	. GLOBL	SYCH6, SYCH7, SYCH10, SYCH11, SYCH12, SYCH13
97	. GLOBL	SYCH14, SYCH15, SYCH16, SYCH17, SYCH20
98	. GLOBL	BLKEY, CHKEY, SYSDAT, DFLG, QCOMP, VPRILO, VPRIHI
99	. GLOBL	SPUSR, SYUNIT, SYSVER, SYSUPD, CONFIG, MAXBLK, VPRIDF
100	. GLOBL	PNAME, HANSIZ, HANENT, DVSTAT, PROSLT, VIDCSR
101	. GLOBL	DEVSIZ, MAXDEV, NUMDEV, SYTIML, SYTIMH, SPLCHN, HANIOC
102	. GLOBL	SWPCHN, NUMIOQ, NUMSYQ, FREIOQ, NMFREQ, INTSSZ
103	. GLOBL	MONVEC, TK1VAL, VHIMEM, CCLSAV, INDSAV
104	. GLOBL	INDBL, INDTSV, INDBBS, R\$INTC
105	. GLOBL	SYINDEX, CONFQ2, SYSGEN, MAXGVL, LDDEVX, CLDEVX, C1DEVX
106	. GLOBL	TMTOTL, TMTOTH, TMUSRL, TMUSRH, SYNAME
107	. GLOBL	TMSWTL, TMSWTH, TMIDLL, TMIDLH, NUMCCB
108	. GLOBL	VMXFIL, VECBAS, VDFMEM, SNMSHD, MXJADR
109	. GLOBL	NMSNMB, NMUMB, CORTIM, RF\$WRT, GENTOP
110	. GLOBL	BASMAP, LOMAP, HIMAP, FREPGS
111	. GLOBL	JCXPGS, MXJMEM, DFJMEM, TK5VAL, TK3SVL
112	. GLOBL	R\$CHN, R\$DATE, R\$UBAS, R\$JOB, R\$CH17
113	. GLOBL	R\$XCHN, EXTCHN, MVWDS, TTOPTS, MAPSIZ
114	. GLOBL	DVFLAG, VBUSTP, QBUS, UNIBUS
115	. GLOBL	VUCLMC, UCLDAT, UCLBLK, VUCLOR, SR3FLG, IDSFLG

```

116 . GLOBL SMRSIZ, SRTSIZ, CSHSIZ, MIOWHD, MIOSYQ
117 . GLOBL PROFLG, MPARFL, PIDPTR
118 . GLOBL PROBRK, SPOLID, VDBFLG, PROODC
119 . GLOBL HANPAR, MHNSIZ, KUSECK, USWPCH, R$SWPC
120 . GLOBL CA$BLK, CA$DVU, CA$HBL, CA$HFL, CA$HSH, CA$UBL, CA$UFL, CA$WCT
121 . GLOBL CSHBFP, CSHLRU, CSHMRU, CSHFHD, CCBHD
122 . GLOBL CASTBR, CASCBR, CASTBW, CASCUP, CASTRO, CASTWO
123 . GLOBL CSHIO, CSHINI, CSHCLN, CSHFIN, CSHBAS, CSHVEC
124 . GLOBL LOKVEC, LOKINI, DOOPAP, DDCOPN, SFSVST, SFRSST, DORLK
125 . GLOBL DOTLK, DDCULK, DOULK1, DOSFCK, SFCLS, SFWRIT, CLSCDB
126 . GLOBL DCRD1, DCRD2, LOKVEC, CLKVEC, VUSPHN, ABRTOV
127 . GLOBL KMNSTK, KMNSTR, KMNPGS, KMNCHN
128 . GLOBL DATIML, DATIMH
129 . GLOBL CXTWDS, CXTPAG, CXTPDR, CXTRMN, CXTBUF, CXBBAS, CXBSIZ
130 ;
131 ; Global references
132 ;
133 . GLOBL ININT, DLINT, LINNUM, PROITP, FNDHRB
134 . GLOBL SD$HLD, MUXNUM, RMNBAS, SETERR
135 . GLOBL IOFIN, INTEN, PSW, SYNCH
136 . GLOBL LA36, LA120, VT52, VT100, ADM3A, VT200
137 . GLOBL DIABLO, QUME, HAZEL, TSDEFS, EXTP1, HANXMR, BLKMOV, CVTPHY
138 . GLOBL GETRTQ, QFREE, IOSTRT, FRKGET, FORKQ, QIO, QCOMPL
139 . GLOBL SCHED, DSKBUF, IOQSIZ
140 -----
141 ; Internal parameters
142 ;
143 001200 TSXVRS = 640. ;TSX-Plus version number
144 000010 MAXDEV = 8. ;Max number of devices that can be supported
145 000031 NUMFRK = 25. ;# Fork request blocks
146 000000 NXIVMH = 0. ;# extra interrupt vectors for mapped handlers
147 000120 DETCBS = 80. ;# characters for detached job startup cmd.
148 000004 FRKGEN = 4. ;# Fork blocks in TSGEN
149 000004 NMSNMB = 4. ;# System message buffers
150 000002 NMSYMB = 2. ;# message buffer reserved for system use
151 000024 NUCHN = 20. ;# I/O channels user may use
152 000001 MAXMUX = 1. ;Max # of DZ11's, DH11's, and DHV11's
153 000040 DHBFSZ = 32. ;# bytes for DH11 and DHV11 DMA output buffers
154 000012 TTCSCH = 10. ;# characters printed per scheduler check
155 000020 NUMIOQ = 16. ;# I/O queue elements
156 000006 NUMSYQ = 6. ;# I/O queue elements reserved for system I/O
157 000012 NUMCCB = 10. ;# data cache control blocks
158 000006 NSCP = 6. ;# swapper command packets
159 000454 INTSSZ = 300. ;# bytes for system interrupt stack
160 000123 MNUAOT = 83. ;Replaces TSEXC2 cell
161 000002 MIONWB = 2. ;Number of mapped I/O wait queue elements
162 000000 MIODBG = 0. ;1=Force I/O mapping (debugging use only)
163 000000 MPARFL = 0. ;1==>Enable mem parity traps, 0==>disable.
164 000000 PROBRK = 0. ;1==>Enable ODT break on PRO printer port
165 000003 PROODC = 3. ;Clock ticks per PI driver call
166 000000 FASTIN = 0. ;1==>Clock driven input character processing
167 000100 CLKVEC = 100 ;Clock interrupt vector
168 000006 DCAGE = 6. ;Shared file data cache ageing factor
169 000001 KUSECK = 1. ;0==>Don't check for usage on INIT & SQUEEZE.
170 000024 IOHLM = 20. ;# 0.1 secs I/O can be held for job swapping
171 000007 CLEQFS = 7. ;Max num of chars in CL ENDSTRING parameter
172 000002 NDRTDF = 2. ;Number of dummy shared run-time definitions

```

```

173      000000      $PRIV      =      0      ; Obsolete line-def privilege flag
174      000020      $NOVLN     =      20     ; Obsolete no-virtual-line privilege flag
175
176      ; -----
177      ; Fork priority values.
178      ; Unlike RT-11, TSX-Plus assigns priority values to its fork requests
179      ; and allows higher priority fork requests to interrupt lower priority ones.
180      ; The priority values range from 1 to 127. The higher the numerical value,
181      ; the higher the priority.
182
183      . GLOBL      FP$RT, FP$CKT, FP$DEF, FP$IOS, FP$IOF, FP$IOA, FP$MOV
184      . GLOBL      FP$CDI, FP$CDO, FP$CK1, FP$MAX, FP$FLG, FP$PIO
185
186      FP$FLG     =      100000      ; Flag saying this is a priority value
187      FP$MAX     =      FP$FLG+127. ; Max legal fork priority
188      FP$RT      =      FP$FLG+100. ; Real-time interrupts
189      FP$CKT     =      FP$FLG+70.  ; 50/60 Hz clock interrupt processing
190      FP$CDI     =      FP$FLG+60.  ; Terminal character input processing
191      FP$CDO     =      FP$FLG+55.  ; Terminal character output processing
192      FP$DEF     =      FP$FLG+50.  ; Default fork priority
193      FP$IOF     =      FP$FLG+50.  ; I/O finish
194      FP$IOA     =      FP$FLG+50.  ; I/O abort entry
195      FP$PIO     =      FP$FLG+50.  ; PI output interrupt processing
196      FP$CK1     =      FP$FLG+30.  ; 0.1 second clock processing
197      ; The following fork priorities are entered from a non-interrupt state.
198      FP$IOS     =      FP$FLG+12.  ; I/O initiation
199      FP$MOV     =      FP$FLG+10.  ; Move data to/from cache buffer
200
201      ; -----
202      ; Completion routine class priorities.
203      ; A completion routine with a higher (numerically larger) class priority
204      ; is allowed to interrupt a lower class priority completion routine.
205
206      . GLOBL      CP$STD, CP$RT, CP$SYN
207
208      CP$STD     =      1      ; Standard -- I/O completion, .TIMIO, etc.
209      CP$RT      =      2      ; Real-time completion routine
210      CP$SYN     =      3      ; .SYNCH completion routine
211
212      ; Type codes used to identify communication device controllers
213
214      CDX$DL     =      0      ; DL11
215      CDX$DZ     =      2      ; DZ11
216      CDX$DH     =      4      ; DH11
217      CDX$VH     =      6      ; DHV11
218      CDX$PI     =      10     ; Console terminal on Professional
219      CDX$PC     =      12     ; Communications port on Professional
220      CDX$PP     =      14     ; Printer port on Professional
221      CDX$QP     =      16     ; 4 line Multiplexer on Professional
222
223      CFCHAN     =      NUCHN      ; Channel to use for command file input
224      LOGCHN     =      NUCHN+1    ; Channel for log file
225      USPLCH     =      NUCHN+2    ; Channel to use to write to spool file
226      RUNCHN     =      NUCHN+3    ; Channel to use when loading a SAV file
227      USWPCH     =      NUCHN+4    ; Channel to use to access swap file
228      NLCHN      =      NUCHN+5    ; Total # channels allocated per job
229      LSTMX      =      0      ; Index to last mux

```

230	000000	CURMX	=	0	;Current mux #
231	000000	NUMRDB	=	0	;Count number of shared run-times declared
232	000000	CURCDX	=	CDX\$DL	;Comm device type for current line
233	000000	DHUSE	=	0	;Set to 1 if DH11 or DHV11 support needed

```

1          ; -----
2          ; Monitor fixed-offset value vector
3          ;
4          ; Table of addresses of TSX-Plus routines. The pointer to this vector is
5          ; stored at simulated RMON offset -2.
6          ;
7          ; Negative offsets from TSXVEC reserved to users
8 000000 002104' TSXVEC: .WORD NUMDEV          ; 0 Ptr to word with # devices in system
9 000002 000560'   .WORD HANENT           ; 2 Vector with handler entry points
10 000004 000702'  .WORD HANPAR           ; 4 64-byte phys mem base of mapped handlers
11 000006 000000G  .WORD GETRTQ           ; 6 Routine to get a free I/O queue element
12 000010 000000G  .WORD QFREE           ; 10 Routine to free an I/O queue element
13 000012 000000G  .WORD QIO            ; 12 Routine to queue an I/O request
14 000014 000000G  .WORD IOSTRT          ; 14 Routine to requeue an I/O request
15 000016 000000G  .WORD QCOMPL          ; 16 Routine to queue a completion request
16 000020 000000G  .WORD FRKGET          ; 20 Routine to get a fork request block
17 000022 000000G  .WORD FORKQ           ; 22 Routine to queue a fork request
18 000024 000000G  .WORD IOHANG          ; 24 Place I/O queue element on handler list
19 000026 000000G  .WORD GETUMR          ; 26 Allocate a Unibus map register
20 000030 000000G  .WORD FREUMR          ; 30 Free a Unibus map register
21 000032 001200   .WORD TSXVRS          ; 32 TSX-Plus version number
22 000034 000000G  .WORD IOQSIZ          ; 34 Size of an I/O queue element (bytes)
23          ;
24          ; Macro to reserve I/O channel space.
25          ;
26          .MACRO CHNRES
27          .WORD 0,0,0,0,0
28          .ENDM CHNRES
29          ;
30          ; -----
31          ; Fixed-offset vector
32          ;
33          ; The following vector of addresses and values corresponds to the fixed
34          ; offset cells in RT-11 RMON. These cells are mapped into user
35          ; address space through PAR7 (160000 - 177777).
36          ;
37 000036 000000' VECBAS: .WORD TSXVEC          ; -2 Pointer to vector of TSX addresses
38 000040 000167 000000G MONVEC: JMP INTEN          ; 0 Handler interrupt entry point
39          ;
40          ; System channel space
41          ;
42 000044          SYSCHN:
43 000044          SYCH0: CHNRES
44 000056          SYCH1: CHNRES
45 000070          SYCH2: CHNRES
46 000102          SYCH3: CHNRES
47 000114          SYCH4: CHNRES
48 000126          SYCH5: CHNRES
49 000140          SYCH6: CHNRES
50 000152          SYCH7: CHNRES
51 000164          SYCH10: CHNRES
52 000176          SYCH11: CHNRES
53 000210          SYCH12: CHNRES
54 000222          SYCH13: CHNRES
55 000234          SYCH14: CHNRES
56 000246          SYCH15: CHNRES
57 000260          SYCH16: CHNRES

```

58	000272		SYCH17: CHNRES	
59	000304		SYCH20: CHNRES	
60			;	
61	000316	000000	BLKEY: .WORD 0	;256 - # of directory block that is in core
62	000320	000000	CHKEY: .WORD 0	;260 - # of device whose dir block is in core
63	000322	000000	SYSDAT: .WORD 0	;262 - System date word
64	000324	000000	DFLG: .WORD 0	;264 - Directory op is in progress
65			;	
66			;	The following cells are documented for access by .GVAL
67			;	
68	000326	000100	USROFF: .WORD HIMEM	;266 - Base of USR
69	000330	000000G	GCOMP: .WORD IOFIN	;270 - I/O completion handler entry point
70	000332	000000	SPUSR: .WORD 0	;272 - USR error cell
71	000334	000000	SYUNIT: .WORD 0	;274 - Unit # of SY device
72			;	
73			;	To force the system version number response to a particular version, set
74			;	the version and update numbers below correspondingly. If SYSVER is 0
75			;	the version number will automatically be based on the monitor version
76			;	from which TSX-Plus is started, with a cut-off based on the latest
77			;	features actually supported (most particularly the EMTs implemented).
78			;	
79	000336	000	SYSVER: .BYTE 0	;276 - System version number
80	000337	000	SYSUPD: .BYTE 0	;277 - Release #
81	000340	000000	CONFIG: .WORD 0	;300 - System configuration word
82	000342		.BLKW 5.	;302 - 313 (unused)
83	000354	001750	MAXBLK: .WORD MAXFIL	;314 - Largest output file size
84			;	Word 316 in TSX-Plus is reserved for specific use. It must
85			;	be initialized to zero and not used by the operating system.
86			;	See the note in TSDEFS for more specific information.
87	000356	000000 000000	.WORD 0,0	;316 - 321 unused
88	000362	000000	CORUSR: .WORD 0	;322 - Current job number
89	000364	000000G	.WORD SYNCH	;324 - Address of .SYNCH request routine
90	000366		.BLKW 13.	;326 - 357 unused
91	000420	000445	BR MTPS	;360 - Move to PS routine
92	000422	000432	BR MFPS	;362 - Move from PS routine
93	000424	000000	SYINDX: .WORD 0	;364 - Device number of system device
94	000426	000000	CFSTS: .WORD 0	;366 - Command file status flags
95	000430	000000	CONFG2: .WORD 0	;370 - Extended configuration word
96	000432	000000	SYSGEN: .WORD 0	;372 - System generation options
97	000434	000002	.WORD 2	;374 - Size of USR
98	000436	014	CFABLV: .BYTE 14	;376 - Error abort severity level
99	000437	003	.BYTE 3	;377 - Max @file nesting level
100	000440	000000	EMTRTN: .WORD 0	;400 - EMT return point
101	000442	000000	FRKADR: .WORD 0	;402 - Fork routine
102	000444	000500	PNPTR: .WORD PNAME-MONVEC	;404 - Offset to permanent dev name table
103	000446	071677 142615	MONAME: .RAD50 /RT11XM/	;406 - 410 - System name
104	000452	000000	HSUFFIX: .WORD 0	;412
105	000454	000000	SPSTAT: .WORD 0	;414 - Spooler status flags
106	000456	000	.BYTE 0	;416 - Error byte for IND
107	000457	000	INSTA: .BYTE 0	;417 - IND status byte
108	000460	000000	#MEMSZ: .WORD 0	;420 - Total 32-word mem blocks avail
109	000462	000000	.WORD 0	
110	000464	000442G	#TCFIG: .WORD TTOPTS+RMNBAS	;424 - Address of TT config word
111	000466	000444G	#INDDV: .WORD INDOFF+RMNBAS	;426 - Pointer to IND device name word
112	000470	001102	MEMPTR: .WORD HNMEPT-MONVEC	;430 - Offset to memory control blocks
113	000472	001132'	P1EXT: .WORD PIXPTR	;432 - Kernel PAR1 routine
114	000474	000000	RPRCSR: .WORD 0	;434 - Get CSR address of PRO devices

```

115 000476 000000 RPRVEC: .WORD 0 ;436 - Get vector address of PRO devices
116 000500 000000 DWTYPE: .WORD 0 ;440 - Type of DW disk
117 ; Dummy cell corresponding to cell in RT-11 with TT option flags.
118 000442 TTOPTS = .-MONVEC ;Offset to TTOPTS cell
119 000502 000000 TTOP: .WORD 0 ;TTOPTS cell
120 ; Cell with name of IND RUN device
121 000444 INDOFF = .-MONVEC ;Offset to INDDEV cell
122 000504 123 131 060 INDDEV: .ASCII /SYO:/ ;Default device from which IND is run
    000507 072
123 ;-----
124 ; MFPS is called to return on the stack the contents of the low-order
125 ; byte of the processor status word.
126 ; Note: This only works when used within handlers. If a .MFPS macro
127 ; is used in a TSX-Plus user job, a value of zero is returned.
128 ;
129 ; Outputs:
130 ; Processor status word is on top of stack.
131 ;
132 000510 005046 MFPS: CLR -(SP)
133 000512 013716 000000G MFPMOV: MOV @PSW,(SP) ;Get the psw (** patched during job init **)
134 000516 016646 000002 MOV 2(SP),-(SP) ;Now push return address on top
135 000522 016666 000002 000004 MOV 2(SP),4(SP) ;Move down PS value
136 000530 012616 MOV (SP)+,(SP) ;Move down return address
137 000532 000207 RETURN
138 ;
139 ; MTPS is called to set the value of the low-order byte of the
140 ; processor status word.
141 ;
142 ; Inputs:
143 ; Value to be moved to psw is on top of stack before call.
144 ;
145 ; Outputs:
146 ; Value is moved to psw and popped from the stack.
147 ;
148 000534 000006 MTPS: RTT ;PC&PS are on stack, let RTT set PS and return
149 ;
150 ;-----
151 ; Device and handler information tables (Do not change the order).
152 ;
153 ; *** VM depends on PHYMEM being 1 word below PNAME ***
154 ;
155 000536 000000 PHYMEM: .WORD 0 ;*** Store actual physical memory size ***
156 000540 000010 PNAME: .REPT MAXDEV ;Table of permanent device names (Rad50)
157 .WORD 0
158 .ENDR
159 000560 000010 HANENT: .REPT MAXDEV ;Handler entry point
160 .WORD 0
161 .ENDR
162 000600 177777 .WORD -1 ;Flag to mark end of HANENT table
163 000602 000010 DVSTAT: .REPT MAXDEV ;Device status flags
164 .WORD 0
165 .ENDR
166 000622 000010 HANDSK: .REPT MAXDEV ;Location of handler on the disk
167 .WORD 0
168 .ENDR
169 000642 000010 HANSIZ: .REPT MAXDEV ;Size of device handler
170 .WORD 0

```

```

171
172 000662 000010      DEVSIZ: . ENDR
                       . REPT  MAXDEV      ;# 256-word blocks on device
173                       . WORD  0
174                       . ENDR
175 000702 000010      HANPAR:  . REPT  MAXDEV      ;64-byte base block for handler if mapped
176                       . WORD  0
177                       . ENDR
178 000722 000010      HANIQC:  . REPT  MAXDEV      ;# uncompleted I/O requests for handler
179                       . WORD  0
180                       . ENDR
181 000742 000010      DVFLAG:  . REPT  MAXDEV      ;Table of device characteristics
182                       . WORD  0
183                       . ENDR
184           000722      MAXGVL  =      .-MONVEC      ;Max offset allowed with .GVAL
185
186
187 ; Reserve space for extended channel space
188 ;
189 000762
190           000010      EXTCHN:
191                       . REPT  <NLCHN-17.>
192                       CHNRES
193                       . ENDR
194 ; Reserve space for channel used to access INDTMP file
195 ;
196 001102      INDTSV: CHNRES      ;Channel used for I/O to INDTMP file
197 ;
198 ; Address of channel block for swap file access
199 ;
200           001070'      SWPCHN  =      EXTCHN+<10.*<USWPCH-17.>>
201 ;
202 ; End of MONVEC pointer table area.
203 ;
204           001056      MVSIZ   =      .-VECBAS
205           000427      MVWDS   =      MVSIZ/2      ;# words in mon vector table
206 ;
207 ; Define offsets into monitor vector area
208 ;
209           000006      R$CHN   =      SYSCHN-VECBAS  ;Start of channel space
210           000234      R$CH17  =      SYCH17-VECBAS ;Offset to channel # 17
211           000724      R$XCHN  =      EXTCHN-VECBAS ;Offset to extended channel space
212           000264      R$DATE   =      SYSDAT-VECBAS ;Offset to date word
213           000324      R$JOB    =      CORUSR-VECBAS ;Offset to job number cell
214           000270      R$UBAS   =      USROFF-VECBAS ;Offset to usr base address cell
215           000444      R$TTOP   =      TTOP-VECBAS  ;Offset to TT option word
216           000421      R$INST   =      INSTA-VECBAS ;Offset to IND status byte
217           000370      R$CFST   =      CFSTS-VECBAS ;Offset to command file status word
218           001044      R$INTC   =      INDTSV-VECBAS ;Offset to INDTMP channel block
219           001032      R$SWPC   =      SWPCHN-VECBAS ;Offset to USWPCH channel block
220           000454      R$MFMV   =      MFPMOV-VECBAS ;Offset to MFPMOV instruction
221 ;
222
223 ; Vector of entry points to handler support routines
224 ; (Do not alter the order)
225 ;
226 001114 000167 000000G      JMP      CVTPHY      ;Routine to convert virtual to physical addr
227 001120 000167 000000G      JMP      FNDHRB      ;Routine to search for RCB for handler

```

```
228 001124 000167 000000G          JMP      HANXMR          ;Routine to allocate XM region for handler
229 001130 000402                   BR       BMJMP          ;Routine to do block move
230 001132 000167 000000G      PIXPTR: JMP      EXTP1          ;Routine to execute mapped code
231 001136 000167 000000G      BMJMP:  JMP      BLKMOV
232                                ;
233                                ;-----
234                                ; Memory allocation information for handlers
235                                ;
236 001142 000000      HNMEPT: .WORD  0
237 001144 000000                   .WORD  0
238 001146 000000      HANRCO: .WORD  0
```

```

1
2 ; -----
3 ; Misc data cells
4 001150 000002 VQUANO: .WORD QUANO
5 001152 000024 VQUAN1: .WORD QUAN1
6 001154 000002 VQUN1A: .WORD QUAN1A
7 001156 000002 VQUN1B: .WORD QUAN1B
8 001160 000001 VQUN1C: .WORD QUAN1C
9 001162 000012 VQUAN2: .WORD QUAN2
10 001164 000024 VQUAN3: .WORD QUAN3
11 001166 000002 VCORTM: .WORD CORTIM
12 001170 000050 VHIPCT: .WORD HIPRCT
13 001172 000036 VINTIO: .WORD INTIOC
14 001174 000062 VMXSF: .WORD MAXSF
15 001176 000144 VMXSFC: .WORD MAXSFC
16 001200 000000 VNUMDC: .WORD NUMDC
17 001202 000003 VMLBLK: .WORD MXLBLK
18 001204 000024 VUCLMC: .WORD UCLMNC
19 001206 001750 VMXFIL: .WORD MAXFIL
20 001210 000074 VNFCSH: .WORD NMFCSH
21 001212 000005 VMXMON: .WORD MAXMON
22 001214 000170 VTMIN: .WORD TIMIN
23 001216 000001 VTMLOC: .WORD TIMLOC
24 001220 000170 VTMOU: .WORD TIMOUT
25 001222 000170 VONTM: .WORD ONTIM
26 001224 000074 VOFFTM: .WORD OFFTIM
27 001226 000005 VMAXMC: .WORD MAXMC
28 001230 000310 VMSCHR: .WORD MSCHRS
29 001232 000006 VMXMSG: .WORD MAXMSG
30 001234 000005 VMXMRB: .WORD MAXMRB
31 001236 000100 VHIMEM: .WORD HIMEM
32 001240 000100 VDFMEM: .WORD DFLMEM
33 001242 000012 VSWPSL: .WORD SWPSLT ;# of job slots in swap file
34 001244 000764 VPLAS: .WORD SEGBLK ;# blocks for PLAS swap file
35 001246 000014 VNGR: .WORD NGR ;Number of global PLAS regions
36 001250 000000 NDVRCB: .WORD DEVXMR ;Number of PLAS regions for device handlers
37 001252 000012 VMXWIN: .WORD MAXWIN ;Maximum number of display windows
38 001254 000007 VKEYMX: .WORD KEYMAX ;Maximum # user-defined keys
39 001256 000004 VNUIP: .WORD NUIP ;Number of user programs that may be INSTALLED
40 001260 000454 VCSHNB: .WORD CACHE ;# blocks in use for generalized data cache
41 001262 000454 CSHALC: .WORD CACHE ;# blocks allocated for generalized data cache
42 001264 040150 VNRFLG: .WORD NRMFLG ;Default time-sharing line flags
43 001266 000000 VPMSIZ: .WORD PMSIZE ;Bytes for performance monitor
44 001270 000000G VSCHED: .WORD SCHED ;An entry point in TSEXEC
45 001272 000000G VDSKBU: .WORD DSKBUF ;A global from TSINIT
46 001274 000062 VNCSLD: .WORD NCSILO ;Default #bytes for TT and CL silos
47 001276 014 VNCXOF: .BYTE NCXOFF ;Default XOFF when only this many free
48 001277 004 VNCXON: .BYTE NCXON ;Default XON when this many remain
49 001300 000276 VDISPC: .WORD DINSPC ;Default line input buffer size
50 001302 000360 VDOSPC: .WORD DOTSPC ;Default line output buffer size
51 001304 000000 SYTIMH: .WORD 0 ;High-order system time word
52 001306 000000 SYTIML: .WORD 0 ;Low-order system time word
53 001310 000000 TK1SEC: .WORD 0 ;# clock ticks per second
54 001312 000000 TK1VAL: .WORD 0 ;# clock ticks per 0.1 second
55 001314 000000 TK1CNT: .WORD 0
56 001316 000000 TK5VAL: .WORD 0 ;# clock ticks per 0.5 seconds
57 001320 000000 TK3SVL: .WORD 0 ;# clock ticks per 3 seconds

```

58	001322	000000			TSXSIT:	.WORD	0	
59	001324	000000			FRKDLY:	.WORD	0	;Max clock ticks a fork request was delayed
60	001326	000000			CTRLTT:	.WORD	0	;# of operator's console
61	001330	000000			MINTIM:	.WORD	0	;Number of minutes of system up-time
62	001332				SEGCHN:	.BLKW	5	;Channel block used for PLAS region swapping
63	001344	000000			KMNTOP:	.WORD	0	;Abs address of top of TSKMON
64	001346	000000			KMNHI:	.WORD	0	;KMNTOP-KMNBAS
65	001350				CCLSAV:	.BLKW	5	;Savestatus for CCL.SAV file info
66	001362				INDSAV:	.BLKW	5	;Savestatus for IND.SAV file info
67	001374	000000			INDDBL:	.WORD	0	;Lowest block in IND.SAV file of data segment
68	001376	000000			INDDBS:	.WORD	0	;Number of blocks in IND.SAV data segment
69	001400	000000			USRBAS:	.WORD	0	;Phys 64-byte block # of TSUSR overlay
70	001402	000000			MSGBAS:	.WORD	0	;Phys 64-byte block # of TSMMSG overlay
71	001404	000000			WINBAS:	.WORD	0	;Phys 64-byte block # of TSWIN overlay
72	001406	000000			LOKBAS:	.WORD	0	;Phys 64-byte block # of TSLOCK overlay
73	001410	000000			CSHBAS:	.WORD	0	;Phys address of TSCASH code
74	001412	000000			TIOBAS:	.WORD	0	;Phys address of TSTIOX code
75	001414	000000			LOKMEM:	.WORD	0	;Phys 64-byte block # of rec locking data area
76	001416	000000			LOKCSH:	.WORD	0	;Phys 64-byte block # of shared file cache buf
77	001420	000000			NUMDCD:	.WORD	NUMDC	;Num of shared file data cache entries
78	001422	000144			NUMCDB:	.WORD	MAXSFC	;Number of free shared file channels
79	001424	000000			SNMSHD:	.WORD	0	;Head of free list of system message buffers
80	001426	000002			NMUMB:	.WORD	<NMSNMB-NMSYMB>	;# message buffers available for user access
81	001430	000000			CSHHD:	.WORD	0	;Head of directory cache list
82	001432	000000			MONFQH:	.WORD	0	;Head of free list of monitor control blocks
83	001434	000000			MIOBHD:	.WORD	0	;Head of mapped I/O control block list
84	001436	000000			MIOWHD:	.WORD	0	;Head of mapped I/O wait block list
85	001440	000000			MIOSYQ:	.WORD	0	;Pointer to 1st active mapped I/O wait block
86	001442	000000			SMONHD:	.WORD	0	;Head of job monitoring requests for all jobs
87	001444	000000			SFCB:	.WORD	0	;Start of spool file control block area
88	001446	000000			SFCBND:	.WORD	0	;End of spool file control block area
89	001450	000000			SFCBFH:	.WORD	0	;Head of free spool file control block list
90	001452	000036			NSPLFL:	.WORD	SPLNF	;Number of spool files
91	001454	000372			NSPLBL:	.WORD	SNDBX	;Number of blocks in spool file
92	001456	000000			NFRESB:	.WORD	0	;Number of public spool file blocks
93	001460	000000			SHRRCB:	.WORD	0	;Pointer to base of global RCB area
94	001462	000000			SHRRCN:	.WORD	0	;Pointer to end of global RCB area
95	001464	000000			INSTBL:	.WORD	0	;Pointer to base of INSTALL table
96	001466	000000			INSTBN:	.WORD	0	;Pointer past end of INSTALL table
97	001470	000000			ABRTOV:	.WORD	0	;Rad50 name of overlay during trap
98	001472	000036			VMXCSH:	.WORD	MAXCSH	;Max number of cached devices
99	001474	000000			CSHDEV:	.WORD	0	;Start of area with device cache blocks
100	001476	000000			CSHDVN:	.WORD	0	;End of area with device cache blocks
101	001500	000000			SCPFHD:	.WORD	0	;Head of free list of swap command packets
102	001502	177777			LDDEVX:	.WORD	-1	;Device index number of "LD" device
103	001504	177777			CLDEVX:	.WORD	-1	;Device index number of "CL" device
104	001506	177777			C1DEVX:	.WORD	-1	;Device index number of "C1" device
105	001510	000000	000000	000000	BOTDEV:	.WORD	0,0,0,0	;Device spec for device being booted from
	001516	000000						
106	001520	000000			BOTUNI:	.WORD	0	;Unit # of device being booted from
107	001522	000000			BOTCSR:	.WORD	0	;CSR of device being booted from
108	001524	000000			SPOLID:	.WORD	0	;Last spool file ID number
109	001526	000000			UMSYTP:	.WORD	0	;Address of top of unmapped system space
110	001530	000001			IOABFL:	.WORD	IOABT	;1==>Do I/O abort, 0==>Do I/O wait
111	001532	000000G			DEFBAS:	.WORD	TSDEFS	
112	001534	114716			SYNAME:	.RAD50	/XXN/	;Actual name of SY physical device
113	001536	075250	100020	101704	UCLNAM:	.RAD50	/SY TSXUCLSAV/	;Name of TSXUCL program

```

001544 073376
114 001546 000000          UCLBLK: .WORD 0          ;# blocks in TSXUCL data file for each job
115 001550 075250 102405 057760 UKMNAM: .RAD50 /SY UKMON SAV/ ;Name of user-provided TSKMON command processr
    001556 073376
116 001560 000000          PIDPTR: .WORD 0          ;Pointer to clock-driven PI handler routine
117 001562          PROSLT: .BLKW 9          ;ID # of device in each PRO option slot
118 001604 000000          VIDCSR: .WORD 0          ;Address of PRO video CSR
119 001606 177564          VDMTCR: .WORD DMPTCR      ;Transmitter control reg addr for dump device
120 001610 000000          MODDAT: .WORD 0          ;Date last modified by TSXMOD
121 001612 000000          MODTIM: .WORD 0          ;Time (3-sec) last modified by TSXMOD
122 001614 000000          HANRCB: .WORD 0          ;Pointer to start of handler RCB area
123
124          ; Data for generalized data cache
125
126 001616 000000          CA$BLK: .WORD 0          ;Block number associated with cache entry
127 001620 000000          CA$DVU: .WORD 0          ;Device and unit # associated with entry
128 001622 000000          CA$WCT: .WORD 0          ;Number of words in entry
129 001624 000000          CA$UFL: .WORD 0          ;LRU chain forward link
130 001626 000000          CA$UBL: .WORD 0          ;LRU chain backward link
131 001630 000000          CA$HFL: .WORD 0          ;Hash chain forward link
132 001632 000000          CA$HBL: .WORD 0          ;Hash chain backward link
133 001634 000000          CA$HSH: .WORD 0          ;Hash chains list head vector
134 001636 000000          CSHBFP: .WORD 0          ;64-byte block number of buffer area
135 001640 000000          CSHLRU: .WORD 0          ;Pointer to least-recently-used entry
136 001642 000000          CSHMRU: .WORD 0          ;Pointer to most-recently-used entry
137 001644 000000          CSHFHD: .WORD 0          ;Head of cache block free list
138 001646 000000          CCBHD: .WORD 0          ;Head of cache control block free list
139 001650 000000 000000          CASTRD: .WORD 0,0        ;Total # reads from mounted devices
140 001654 000000 000000          CASTBR: .WORD 0,0        ;Total # blocks read from mounted devices
141 001660 000000 000000          CASCBR: .WORD 0,0        ;Number of blocks that were read from cache
142 001664 000000 000000          CASTWD: .WORD 0,0        ;Total # writes to mounted devices
143 001670 000000 000000          CASTBW: .WORD 0,0        ;Total # blocks written to mounted devices
144 001674 000000 000000          CASCUP: .WORD 0,0        ;Number of blocks moved into data cache
145
146          ; Entry point vector for caching module
147 001700          CSHVEC:
148 001700 000000          CSHINI: .WORD 0          ;-
149 001702 000000          CSHIO: .WORD 0          ;-
150 001704 000000          CSHCLN: .WORD 0          ;-
151 001706 000000          CSHFIN: .WORD 0          ;-
152          .WORD -1          ;- End of pointer vector
153
154          ; Entry point vector for record locking module
155
155 001712          LOKVEC:
156 001712 000000          LOKINI: .WORD 0          ;-
157 001714 000000          DOOPAP: .WORD 0          ;-
158 001716 000000          DOCOPN: .WORD 0          ;-
159 001720 000000          SFSVST: .WORD 0          ;-
160 001722 000000          SFRSST: .WORD 0          ;-
161 001724 000000          DORLK: .WORD 0          ;-
162 001726 000000          DOTLK: .WORD 0          ;-
163 001730 000000          DOCULK: .WORD 0          ;-
164 001732 000000          DOULK1: .WORD 0          ;-
165 001734 000000          DOSFCK: .WORD 0          ;-
166 001736 000000          SFCLS: .WORD 0          ;-
167 001740 000000          SFWRIT: .WORD 0          ;-
168 001742 000000          CLSCDB: .WORD 0          ;-

```

```

169 001744 000000          DCRD1:  .WORD  0          ; -
170 001746 000000          DCRD2:  .WORD  0          ; -
171 001750 177777          .WORD  -1          ; - End of pointer vector
172          ;
173          ; Misc byte data
174          ;
175 001752 000          VSYDMP: .BYTE  SYSDMP          ;Generate dump on crash if non-zero
176 001753 000          VDMKTP: .BYTE  DMPKTP          ;Crash on any kernel trap if non-zero
177 001754 001          VSWPFL: .BYTE  SWAPFL
178 001755 001          VBUSTP: .BYTE  BUSTYP
179 001756 000          VINABT: .BYTE  INIABT
180 001757 000          VUXIFL: .BYTE  UXIFLG
181 001760 001          VU$CL:  .BYTE  U$CL
182 001761 002          VUCLOR: .BYTE  UCLORD
183 001762 001          VLDSYS: .BYTE  LDSYS
184 001763 001          VSLEDT: .BYTE  SLEDIT
185 001764 000          VDBFLG: .BYTE  DBGFLG
186 001765 023          VPRILO: .BYTE  PRILOW
187 001766 120          VPRIHI: .BYTE  PRIHI
188 001767 062          VPRIDF: .BYTE  PRIDEF
189 001770 012          VPRIVR: .BYTE  PRIVIR
190 001771 035          VTSLCH: .BYTE  TSLICH
191 001772 027          VVLSCH: .BYTE  VLSWCH
192 001773 002          VVPWCH: .BYTE  PWCH
193 001774 034          VCXTRM: .BYTE  CCXTRM
194 001775 001          VCXCTL: .BYTE  CCXCTL
195 001776 024          VCTRLT: .BYTE  STATCH
196 001777 003          VEDIT:  .BYTE  EDITOR
197 002000 000          VMIOBF: .BYTE  MIONBF
198 002001 017          VMIOSZ: .BYTE  MIOBSZ
199 002002 000          VUSPHN: .BYTE  PHONE          ; 0=local if no DCD; 1=always mon DCD if $phone
200 002003 123          VMNUAD: .BYTE  MNUAOT
201 002004 000          MAPUSR: .BYTE  0          ; Number of job memory mapping is set up for
202 002005 000          LINNUM: .BYTE  0
203 002006 000          MUXNUM: .BYTE  0
204 002007 000          NUMON:  .BYTE  0
205 002010 000          PVON:   .BYTE  0
206 002011 000          TON:    .BYTE  0
207 002012 000          PROFLG: .BYTE  0          ; Non-zero ==> Running on PRO-350
208 002013 000          STPFLG: .BYTE  0
209 002014 000          UBUSMP: .BYTE  0          ; 1==>Do Unibus mapping
210 002015 000          SR3FLG: .BYTE  0          ; NON-ZERO==>MEMORY MANAGEMENT REG 3 PRESENT
211 002016 000          IDSFLG: .BYTE  0          ; Non-zero==>machine implements D-space
212 002017 000          MEM256: .BYTE  0          ; Non-zero==>machine has at least 256kb
213 002020 000          MIOFLG: .BYTE  0          ; Non-zero==>I/O mapping needed for some device
214 002021 000          NSPLDV: .BYTE  0          ; Number of installed spooled devices
215 002022 001          KMONCE: .BYTE  1          ; Kmon initialization flag
216 002023 000          EXCJOB: .BYTE  0          ; Index of job with exclusive access
217 002024 000          DOSCHD: .BYTE  0          ; Scheduler needs to be called
218 002025 000          CLVERS: .BYTE  CLVRSN          ; CL handler version number
219 002026 000          LDVERS: .BYTE  0          ; LD translation table format (1<RTV5.4=<2)
220 002027 041          200          SYSPSR: .ASCII  /!/<200>          ; Prompt for system password
221          .EVEN
222          ; System time counters
223 002032 000000          TMTOTH: .WORD  0          ; Total uptime (0.1 second units)
224 002034 000000          TMTOTL: .WORD  0
225 002036 000000          TMUSRH: .WORD  0          ; Time spent in user jobs

```

```

226 002040 000000 TMUSRL: .WORD 0
227 002042 000000 TMSWTH: .WORD 0 ; Swap-wait time
228 002044 000000 TMSWTL: .WORD 0
229 002046 000000 TMIOH: .WORD 0 ; Time user i/o is active
230 002050 000000 TMIOL: .WORD 0
231 002052 000000 TMSWPH: .WORD 0 ; Time swapping is active
232 002054 000000 TMSWPL: .WORD 0
233 002056 000000 TMIQWH: .WORD 0 ; Time system is doing i/o-wait
234 002060 000000 TMIOWL: .WORD 0
235 002062 000000 TMIDLH: .WORD 0 ; Idle time
236 002064 000000 TMIDLL: .WORD 0
237 002066 000000 DATIML: .WORD 0
238 002070 000000 DATIMH: .WORD 0
239 ;
240 ; Shared file data cache statistics counters
241 ;
242 002072 000000 DCTOTU: .WORD 0 ; Total number of cache hits since last divisn
243 002074 000000 DCTRD: .WORD 0 ; Total number of reads from shared files
244 002076 000000 DCCRD: .WORD 0 ; Number of reads satisfied by data in cache
245 002100 000000 DCTWR: .WORD 0 ; Total number of writes to shared files
246 002102 000000 DCCWR: .WORD 0 ; Number of writes that update cache
247 ;
248 002104 000000 NUMDEV: .WORD 0 ; Byte index to last entry in device tables
249 002106 000000 FREIOQ: .WORD 0 ; Head of i/o queue element chain
250 ;
251 ; Define mux tables for DZ11's and DH11's.
252 ;
253 .MACRO MXTBL NAME
254 .NLIST
255 NAME = .-2
256 .GLOBL NAME
257 .REPT MAXMUX
258 .WORD 0
259 .ENDR
260 .LIST
261 .ENDM MXTBL
262 ;
263 002110 MXTBL MXTYPE ; DZ11 & DH11 type of mux (CDX#DZ or CDX#DH)
264 002112 MXTBL MXCSR ; DZ11 Control Status Register
265 002114 MXTBL MXLPR ; DZ11 Line Parameter Register
266 002116 MXTBL MXTCR ; DZ11 Transmit Control Register
267 002120 MXTBL MXDTR ; DZ11 Data Terminal Ready
268 002122 MXTBL MXTBUF ; DZ11 Transmitter Buffer Register
269 002124 MXTBL MXSBRK ; DZ11 Shadow register for hardware BRK reg.
270 002126 MXTBL MXCAR ; DZ11 Carrier Detect
271 002130 MXTBL MXVEC ; DZ11 & DH11 Vector address
272 002132 MXTBL MXLNT ; DZ11 & DH11 Addr of table to map mux # to Lin
273 002134 MXTBL MH$BRK ; DH11 Break control register
274 002136 MXTBL MH$LPR ; DH11 Line Parameter Register
275 002140 MXTBL MH$PBR ; DH11 Previous value of BAR register
276 002142 MXTBL DM$CSR ; DH11(DM11) Control Status Register
277 002144 MXTBL DM$LSR ; DH11(DM11) Line Status Register
278 002146 MXTBL DM$VEC ; DH11(DM11) Address of DM11 interrupt vector
279 002112' MXRBUF = MXLPR ; DZ11 Receiver Buffer Register
280 002120' MXRING = MXTBUF ; DZ11 Ring indicator flags
281 002124' MXBRK = MXCAR ; DZ11 Break control flags
282 ; Equates for DH11 control registers

```

```

283      002110'      MH$SCR =      MXCSR      ;DH11 System Control Register
284      002112'      MH$RCR =      MXRBUF     ;DH11 Received Character Register
285      002114'      MH$CAR =      MXTCR      ;DH11 Current Address Register
286      002116'      MH$BCR =      MXDTR      ;DH11 Byte Count Register
287      002120'      MH$BAR =      MXTBUF     ;DH11 Buffer Active Register
288      002124'      MH$SSR =      MXCAR      ;DH11 Silo Status Register
289      ; Equates for DHV11 control registers
290      002110'      VH$CSR =      MH$SCR     ;DHV11 Control and Status Register
291      002112'      VH$DBR =      MH$RCR     ;DHV11 Data Buffer Register
292      002134'      VH$LPR =      MH$LPR     ;DHV11 Line Parameter Register
293      002142'      VH$LSR =      DM$LSR     ;DHV11 Line Status Register
294      002140'      VH$LCR =      DM$CSR     ;DHV11 Line Control Register
295      002114'      VH$BA1 =      MH$CAR     ;DHV11 Buffer Address register 1
296      002124'      VH$BA2 =      MH$SSR     ;DHV11 Buffer Address register 2
297      002116'      VH$BCR =      MH$BCR     ;DHV11 Byte Count Register
298      ;
299      ; Generate FORK request blocks.
300      ;
301 002150 002154'    FREFRK: .WORD    FRKLST     ;Head of free list
302 002152 000000    FRKINI: .WORD    0         ;Pointer to fork blocks in init area
303 002154          FRKLST:
304          .REPT    FRKGEN     ;Gen in a few static fork blocks
305          .WORD    .+22.      ;Link to next block in free list
306          .WORD    0,0,0,0,0,0,0,0,0,0
307          .ENDR
308 002304 000000 000000 000000    .WORD    0,0,0,0,0,0,0,0,0,0 ;Last block with 0 forward link
      002312 000000 000000 000000
      002320 000000 000000 000000
      002326 000000 000000
309      ;
310      ; Symbolic equates for QBUS and UNIBUS machines.
311      ;
312      000001      QBUS    =      1
313      000000      UNIBUS  =      0
314      ;
315      ; Generate the memory size limit checking certain restrictions.
316      ; On non-extended machines, allow 256.Kb - 8.Kb I/O page
317      ; On extended machines, allow 4096.Kb - 256.Kb I/O page
318      ;
319      .MACRO    MEMORY    SIZE
320      SIZMEM    = SIZE
321      .IF      LE, SIZMEM
322      SIZMEM    = 3840.
323      .ENDC
324      .IF      GT, SIZMEM-3840.
325      SIZMEM    = 3840.
326      .ENDC
327      .IF      LT, <SIZMEM - 96.>
328      .ERROR   ;Memory size limit too small for running TSX-Plus
329      .ENDC
330      ; Allocate the memory size to examine.
331      .WORD    SIZMEM*20
332      .ENDM    MEMORY
333      ;
334      ; Memory management tables
335      ;
336 002332  MAPSIZ: MEMORY    MEMSIZ      ;PAR value of physical memory cutoff

```

```

337 002334 000000      BASMAP: .WORD 0          ;Pointer to base of memory map table
338 002336 000000      LOMAP:  .WORD 0          ;Pointer to 1st user page in MEMMAP
339 002340 000000      HIMAP:  .WORD 0          ;Pointer above top user page in memmap
340 002342 000000      MAPPAR: .WORD 0          ;Value to map PAR 5 to mem allocation table
341 002344 000000      FREPGS: .WORD 0          ;# free pages
342 002346 000000      JCXPGS: .WORD 0          ;# pages needed for job context block
343 002350 000000      MXJMEM: .WORD 0          ;Max # K-bytes a job may use
344 002352 000000      DFJMEM: .WORD 0          ;Default # K-bytes a job may use
345 002354 000000      MXJADR: .WORD 0          ;Address above top of largest job space
346 002356 000000      SMRSIZ: .WORD 0          ;# 64-byte blocks allocated to system overlays
347 002360 000000      MHNSIZ: .WORD 0          ;# 64-byte blocks allocated for mapped handler
348 002362 000000      SRTSIZ: .WORD 0          ;# 64-byte blocks allocated for shared run-tim
349 002364 000000      CSHSIZ: .WORD 0          ;# 64-byte blocks allocated for data cache
350
351      ; Information on KMON position and size.
352
353 002366 000000      KMNSTK: .WORD 0          ;Address of Kmon stack
354 002370 000000      KMNSTR: .WORD 0          ;Starting address of Kmon
355 002372 000000      KMNPGS: .WORD 0          ;# 256-word memory pages needed to run TSKMON
356 002374
357      KMNCHN: .BLKW 5          ;Save status for Kmon file channel
358
359      ; Information about job context region.
360
360 002406 000000      CXTWDS: .WORD 0          ;# words for job context block
361 002410 000000      CXTPAG: .WORD 0          ;# 512-byte pages for job context block
362 002412 000000      CXTPDR: .WORD 0          ;PDR value to map job context block
363 002414 000000      CXTRMN: .WORD 0          ;Address in context area of simulated RMON
364 002416 000000      CXTBUF: .WORD 0          ;Addr of buffer used for accessing cxt blk
365 002420 000000      CXBBAS: .WORD 0          ;Addr of data currently in CXTBUF
366 002422 000000      CXBSIZ: .WORD 0          ;Amt of data currently in CXTBUF
367
368      ; Start a CSECT to hold shared run-time descriptor blocks
369
370      .CSECT RDBSEC          ;CSECT for RDB entries
371 000000      RDBSEC:
372 000000      RDB:
373 002424      .CSECT TSGEN          ;Go back to standard TSGEN CSECT
374
375      ; Symbolic equates for system editor names.
376      ; Note these equates must match those in TSDEFS.
377
378      000001      EDIT      =      1
379      000002      TECO      =      2
380      000003      KED       =      3
381      000004      K52       =      4
382
383      ; Symbolic equates for UCL order
384
385      000001      FIRST     =      1
386      000002      MIDDLE    =      2
387      000003      LAST      =      3
388
389      ; Symbolic names used to define line transmit/receive speeds.
390
391      000000      S50       =      0          ;50      baud
392      000001      S75       =      1          ;75      baud
393      000002      S110      =      2          ;110     baud

```

394	000003	S134.5	=	3		; 134.5 baud
395	000004	S150	=	4		; 150 baud
396	000005	S300	=	5		; 300 baud
397	000006	S600	=	6		; 600 baud
398	000007	S1200	=	7		; 1200 baud
399	000010	S1800	=	10		; 1800 baud
400	000011	S2000	=	11		; 2000 baud
401	000012	S2400	=	12		; 2400 baud
402	000013	S3600	=	13		; 3600 baud
403	000014	S4800	=	14		; 4800 baud
404	000015	S7200	=	15		; 7200 baud
405	000016	S9600	=	16		; 9600 baud
406	000017	S19200	=	17		; 19200 baud
407						
408						; Symbolic names for parity codes
409						
410	040000	EVEN	=	040000		; Even parity
411	140000	ODD	=	140000		; Odd parity
412	000000	NONE	=	000000		; No parity

```

1          ; -----
2          ; The following macro define the device handler tables.
3          ; There are two psects use in the device definition - one
4          ; allocates and defines the rad50 device name - the second
5          ; defines the handler attributes.
6          ;
7          .MACRO  DEVBEG          ; DEFINE THE DEVICE GLOBAL ENTRIES
8          .CSECT  DNAME          ; DEFINE THE DEVICE NAME PSECT
9  AUTHAN:          ; GLOBAL LABEL FOR DEVICE NAMES
10         .CSECT  DTYPE          ; DEFINE THE DEVICE TYPE PSECT
11  DTYPE:
12         .CSECT  TSGEN
13         .ENDM   DEVBEG
14
15         ; -----
16         ; The following flag definitions must match the TSDEFS definitions.
17         ;
18
19         000001   DMA      =      1      ; DX$DMA - This is a DMA device
20         000002   MAPIO   =      2      ; DX$MAP - 18-bit controller -- may require mapped I/O
21         000004   EVNBUF  =      4      ; DX$EBA - Buffer must be on even byte boundary
22         000010   NDCACHE =     10      ; DX$NCA - Do not do caching for this device
23         000020   NOMOUNT =     20      ; DX$NMT - Do not allow mounts for this device
24         000040   REGALC  =     40      ; DX$RAL - Require device to be allocated before use
25         000100   MAPH    =    100      ; DX$MPH - Map the handler for this device
26         000200   NOMAPH  =    200      ; DX$NHM - Do not map the handler for this device
27         000400   HANBUF  =    400      ; DX$IBH - Handler contains internal I/O buffer
28         001000   HNSPDO  =   1000      ; DX$NRD - Do .SPFUN to tell handler about dir ops
29         002000   NOSET   =   2000      ; DX$NST - Do not reload handler after SET
30
31         000000   NODMA   =      0      ; This is not a DMA device
32         000000   NONDMA  =      NODMA
33
34         ; -----
35         ; The DEVDEF macro defines the device name and allocates
36         ; table entries for the device name and the device attributes.
37         ;
38         000001   DVNUM = 1
39
40         .MACRO  DEVDEF  DEVNAM, DFLG1, DFLG2, DFLG3, DFLG4, DFLG5, DFLG6, DFLG7, DFLG8, DFLG9
41
42         DVNUM = DVNUM + 1          ; Increment the device number
43         DVFLG =      0            ; Get device flags in DVFLG
44
45         .IF      LT, <MAXDEV-2 - DVNUM> ; Check the maximum devices allowed
46 .ERROR 1; More devices defined than MAXDEV
47         .MEXIT
48         .ENDC
49
50         ; Accumulate flags for the device definition
51
52         .IF      NB DFLG1          ; Check if argument exists
53         DVFLG = DVFLG!DFLG1        ; Include in device attributes
54         .ENDC ; NB DFLG1
55
56         .IF      NB DFLG2          ; Check if argument exists
57         DVFLG = DVFLG!DFLG2        ; Include in device attributes

```

```

58             . ENDC ;NB DFLG2
59
60             . IF      NB DFLG3           ; Check if argument exists
61 DVFLG = DVFLG!DFLG3                     ; Include in device attributes
62             . ENDC ;NB DFLG3
63
64             . IF      NB DFLG4           ; Check if argument exists
65 DVFLG = DVFLG!DFLG4                     ; Include in device attributes
66             . ENDC ;NB DFLG4
67
68             . IF      NB DFLG5           ; Check if argument exists
69 DVFLG = DVFLG!DFLG5                     ; Include in device attributes
70             . ENDC ;NB DFLG5
71
72             . IF      NB DFLG6           ; Check if argument exists
73 DVFLG = DVFLG!DFLG6                     ; Include in device attributes
74             . ENDC ;NB DFLG6
75
76             . IF      NB DFLG7           ; Check if argument exists
77 DVFLG = DVFLG!DFLG7                     ; Include in device attributes
78             . ENDC ;NB DFLG7
79
80             . IF      NB DFLG8           ; Check if argument exists
81 DVFLG = DVFLG!DFLG8                     ; Include in device attributes
82             . ENDC ;NB DFLG8
83
84             . IF      NB DFLG9           ; Check if argument exists
85 DVFLG = DVFLG!DFLG9                     ; Include in device attributes
86             . ENDC ;NB DFLG9
87
88 ; Enter the device name into the table defining handlers to load on startup
89
90             . CSECT  DNAME
91             X = .
92             . RAD50  /'DEVNAM'/           ; Include the device name in the PSECT
93             . IF      NE, <.-X-2>
94 . ERROR  2; Incorrect device name specified
95             . MEXIT
96             . ENDC
97
98 ; Enter the device specification flag into handler flags table
99
100            . CSECT  DTYPE
101            . WORD   DVFLG                 ; Include the device type in the PSECT
102            . CSECT  TSGEN
103
104            . ENDM   DEVDEF
105
106 ; -----
107 ; The DEVEND macro allocates the remainder of the table entries.
108 ;
109            . MACRO  DEVEND
110 L          = <MAXDEV-2-DVNUM>
111            . IF      GT, L
112            . REPT   L
113            DEVDEF <$$ >
114            . ENDR

```

115
116
117
118
119

. ENDC ;GT, L
. CSECT DNAME
AHEND:
. CSECT TSGEN
. ENDM DEVEND

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

;-----
; The OB macro creates a table with NLINES entries
; and defines the name of the table to be
; 1 word in front of the start of the table.
; The name if globally defined.
;

```

```

      .MACRO  OB      NAME
      .NLIST
NAME   =      .-2
      .GLOBL  NAME
      .REPT   NLINES
      .WORD   0
      .ENDR
      .LIST
      .ENDM   OB

```

```

;-----
; The OBP macro is similar to OB except it
; generates only NPL (# of primary lines) entries
; instead of NLINES entries.
;

```

```

      .MACRO  OBP     NAME
      .NLIST
NAME   =      .-2
      .GLOBL  NAME
      .REPT   NPL
      .WORD   0
      .ENDR
      .LIST
      .ENDM   OBP

```

```

;-----
; The OBH macro is similar to OB except it
; generates TNHL (# lines requiring hardware control tables) entries
; instead of NLINES entries.
;

```

```

      .MACRO  OBH     NAME
      .NLIST
NAME   =      .-2
      .GLOBL  NAME
      .REPT   TNHL
      .WORD   0
      .ENDR
      .LIST
      .ENDM   OBH

```

```

;-----
; The OBT macro is similar to OB except it
; generates NPL+NSL+NDL+NIOL entries
; instead of NLINES entries.
;

```

```

      .MACRO  OBT     NAME
      .NLIST
NAME   =      .-2
      .GLOBL  NAME
      .REPT   NPL+NSL+NDL+NIOL
      .WORD   0

```

```

58          . ENDR
59          . LIST
60          . ENDM   OBT
61
62          ;-----
63          ; The TBLDEF macro is called once to define table
64          ; space needed by all of the lines.
65          ; It has four arguments:
66          ; Argument 1 is the number of primary (real) lines.
67          ; Argument 2 is the number of subprocesses.
68          ; Argument 3 is the number of detached lines.
69          ; Argument 4 is the number of dedicated CL lines.
70          ;
71          . MACRO TBLDEF ANPL, ANSL, ANDL, ANIOL
72          . NLIST MD
73          NPL   =      ANPL           ;# of primary lines
74          NSL   =      ANSL           ;Number of subprocesses
75          NDL   =      ANDL           ;Number of detached lines
76          NIO L =      ANIOL          ;Number of dedicated CL lines
77          NLINES =      NPL+NSL+NDL   ;Total number of jobs
78          ;
79          ; Make sure the total number of CL units does not exceed 16
80          ;
81          . IF      GT <NIO L-16.>
82          . ERROR ;You cannot have more than 16 CL units
83          NIO L   =      16.           ;Reduce number to 16.
84          . ENDC
85          . IF      GT <<NIO L+CLXTRA>-16.>
86          . ERROR ;You cannot have more than 16 CL units
87          CLXTRA  =      16.-NIO L    ;Reduce extra units if total > 16
88          . ENDC
89          CLTOTL  =      NIO L+CLXTRA
90          ;
91          ; Set up number of lines variables.
92          ; The lines are numbered in the following order:
93          ; Primary lines.
94          ; Detached job lines.
95          ; Subprocesses.
96          ; Dedicated CL lines.
97          ;
98          LSTPL  =      2*NPL           ;Last primary line index
99          FSTD L =      LSTPL+2         ;First detached line
100         LSTD L =      LSTPL+<2*NDL>   ;Last detached line
101         FSTSL  =      LSTD L+2        ;Index to first subprocess
102         NLIN2  =      2*NLINES        ;Index to last time-sharing line
103         LSTSL  =      NLIN2           ;Index to last subprocess
104         FSTIOL =      LSTSL+2         ;Index to first CL line
105         LSTIOL =      FSTIOL+<2*<NIO L-1>> ;Index to last CL line
106         LSTLIN =      2*<NPL+NSL+NDL+NIO L> ;Index number of last line
107         . IF      EQ, NIO L           ;If there are no CL lines
108         TNHL   =      NPL             ;Total number of lines with hardware-ctrl tbls
109         . IFF
110         TNHL   =      NPL+NSL+NDL+NIO L ;Total number of lines with hardware-ctrl tbls
111         . ENDC
112         LSTHL  =      TNHL*2          ;Index # of last hardware line
113         ;
114         ; Define number of slots in job swap file if SWPSLT=0

```

```

115 ;
116 . IF EQ, SWPSLT
117 SWPSLT = NLINES ; Default to one slot for each job
118 . ENDC ; EQ, SWPSLT
119 . IF GT, <SWPSLT-NLINES> ; Never need more slots than lines
120 SWPSLT = NLINES
121 . ENDC ; GT, <SWPSLT-NLINES>
122 . IF EQ, SWAPFL ; If this is a non-swapping system
123 SWPSLT = 0 ; No swap slots needed
124 . ENDC ; EQ, SWAPFL
125 ;
126 ; Define line tables.
127 ;
128 OB LQLINK ; Link for execution queues
129 OB LSTATE ; Current execution state
130 OB LBSPRI ; Job base priority value (byte)
131 LPRI = LBSPRI+1 ; Current job priority (byte)
132 OB LPARNT ; Index number of parent job
133 OBT LSW ; Line status word
134 OB LSW2 ; Additional line status
135 OB LSW2S ; Copy of LSW2 used for reset on prog exit
136 OBH ILSW2 ; Initial values for LSW2
137 OBT LSW3 ; Additional line status flags
138 OB LSW4 ; Additional line status flags
139 OBT LSW5 ; More line status flags
140 OBT LSW6 ; Line status table # 6
141 OB LSW7 ; Line status table # 7
142 OB LSW8 ; Line status table # 8
143 OB LSW9 ; Line status table # 9
144 OBT LSW10 ; Line status table # 10
145 OB LSW11 ; Line status table # 11
146 OBT LCLUNT ; CL unit index number if connected as CL line
147 OBP ITRMTP ; Initial terminal type code
148 OBT LTRMTP ; Current terminal type code
149 OBH LNAME ; Descriptive name for line
150 OB LMEMIN ; # pages of memory needed to inswap job
151 OB LPARBS ; PAR base address for job
152 OB LCXPAR ; Value for KPAR6 to map to job context block
153 OB LNBLKS ; # pages of memory currently assigned to job
154 OB LNSBLK ; # pages of memory used by PLAS regions
155 OB LTPAR ; Physical memory PAR value for terminal buffer
156 OB LQUAN ; Job's execution quantum
157 OB LITIME ; Time job is held in "interactive" state
158 OB LIOHLD ; Hold time for I/O starts while starting swap
159 OB LMINQ ; Minimum core-residency time
160 OB LHIPCT ; Controls # high-prio quantum periods job gets
161 OB LIOCNT ; # active i/o operations for job
162 OB LBASE ; Base page # assigned to job
163 OBH LHIRBB ; Start of silo input ring buffer
164 OBH LHIRBE ; End of silo input ring buffer
165 OBH LHIRBA ; Allocated size of silo input ring buf
166 OBH LHIRBS ; Free space in silo input ring buffer
167 OBH LHIRBP ; Pointer where to store next char in buffer
168 OBH LHIRBG ; Pointer where to get next char from buffer
169 OBH LHIRBC ; Autoflow control stop/start char count limits
170 OBT LINSIZ ; Size of input character buffer
171 OB LINBUF ; Start of input buffer

```

172	OB	LINEND	;End of input buffer
173	OB	LINNXT	;Where next input char goes
174	OB	LINPNT	;Where to get next char read
175	OB	LINCNT	;# of chars in input buffer
176	OB	LINSPC	;# free bytes in input buffer
177	OB	LACTIV	;# of activation chars pending
178	OB	LAFSIZ	;Field width for activation condition
179	OB	LFWLIM	;Field width limit
180	OB	LSTACT	;Position of last activation char
181	OB	LINCUR	;Pos of cursor at start of line
182	OBT	LOTSIZ	;Size of output buffer
183	OB	LOTBUF	;Start of output buffer
184	OB	LOTEND	;End of output buffer
185	OB	LOTNXT	;Place to put next output char
186	OB	LOTPNT	;Place to get next output char
187	OB	LOTSPC	;Space left in output buffer
188	OB	LWINDO	;Pointer to current display window block
189	OBH	LCDTYP	;Type of communications device (CDX\$xxx)
190	OBH	LINIR	;Terminal input service routine
191	OBH	LOUTIR	;Terminal output service routine
192	OBH	INVEC	;Input interrupt vector loc
193	OBH	RSR	;Receiver status register address
194	OBH	RBR	;Receiver buffer register
195	OBH	TSR	;Transmitter status register
196	OBH	TBR	;Transmitter buffer register
197	OBH	LDHB1B	;Base of DMA buffer 1
198	OBH	LDHB1P	;Pointer into DMA buffer 1
199	OBH	LDHB2B	;Base of DMA buffer 2
200	OBH	LDHB2R	;Remaining byte count for buffer 2
201	OBH	LDHB2S	;Suspended pointer for buffer 2
202	OBH	LCXTBL	;Pointer to character translation table
203	OBP	LSECPT	;Pointer to secondary line # table
204	OBP	LXCL	;CL unit to which line is cross connected
205	OB	LCMPL	;Head of chain of completion requests for job
206	OB	LCMQHD	;Queue head for completed message requests
207	OB	LMONHD	;Queue head for job monitor blocks
208	OB	LSUCF	;Start-up command file
209	OB	LSWPBK	;Block # in swap file
210	OB	LJSW	;User's JSW
211	OB	LEMTPC	;PC of last user-mode emt
212	OB	LSCCA	;SCCA control word address
213	OB	LSPND	;SPND counter for job
214	OB	LBRKCQ	;Break character completion queue entry
215	OB	LTTCR	;Completion routine for TT input activation
216	OB	LBRKCH	;Break character for line
217	OB	LCOL	;Current column position
218	OBP	LMSGBF	;Send message pointer
219	OB	LSNDCH	;Last char sent
220	OB	LESRTN	;Echo suppression routine
221	OB	LESCHR	;Echo suppression char code
222	OB	LRBFIL	;Rubout filler for line
223	OB	LTSCMD	;Pending special action command
224	OB	LNSPAC	;# of special activation chars
225	OB	LSPACT	;Point to special actv char tbl
226	OB	LPROJ	;Project #
227	OB	LPROG	;Programmer #
228	OB	LCPUHI	;High-order CPU time

229		OB	LCPULO	;Low-order CPU time
230		OB	LCONTM	;Connect time
231		OBP	LCDTIM	;Lost-carrier disconnect time
232		OBP	LOFFTM	;Allowed logoff time before DTR drop for line
233		OBP	LABTIM	;Autobaud control timer
234		OB	LRDTIM	;TT read timeout
235		OB	LRTCHR	;TT read timeout activation character
236		OB	LSLEPL	;.WAIT sleep time for job (low-order)
237		OB	LSLEPH	;.WAIT sleep time for job (high-order)
238		OBH	LMXNUM	;Index # of mux controlling line
239		OBH	LMXPRM	;Line parameters (speed, parity, stop bits)
240		OB	LPRG1	;1st 3 chars of running program name (rad50)
241		OB	LPRG2	;2nd 3 chars of running program name (rad50)
242	LUNAME	=	.-12.	;Offset user name table by size of 1 entry
243		.BLKB	NLINES*12.	;Store 12 char user name here
244	LMXLN	=	RBR	;# of this line within mux group

```

1      ;
2      ; Define subprocess mapping tables
3      ;
4      LNMAP = .-2
5          .NLIST BIN
6      I = 0
7          .REPT NPL
8      I = I+2
9          .WORD I
10         .ENDR
11     ; Define LNPRIM table
12     LNPRIM = .-2
13     I = 0
14         .REPT NPL+NDL
15     I = I+2
16         .WORD I
17         .ENDR
18         .IF NE, NSL
19         .REPT NSL
20     I = I+2 ; Keep count for NIOL if any
21         .WORD 0
22         .ENDR
23         .ENDC
24         .IF NE, NIOL
25         .REPT NIOL
26     I = I+2
27         .WORD I
28         .ENDR
29         .ENDC
30         .LIST BIN
31     ;
32     ; Generate interrupt receivers
33     ;
34     ; Input interrupt vector
35         .NLIST BIN
36         .REPT TNHL
37         INCB LINNUM ; COUNT UP WHICH LINE INTERRUPTED
38         .ENDR
39     INRECV: JMP ININT ; ENTER INTERRUPT SERVICE ROUTINE
40     ; Output interrupt vector
41     LXX = 2
42     OTRECV:
43         .REPT TNHL
44         JSR R4, @DLINT
45         .WORD LXX
46     LXX = LXX+2
47         .ENDR
48         .LIST BIN
49         .LIST MD
50         .ENDM TBLDEF

```

```

1
2 ; -----
3 ; The CLDEF macro begins a line definition block for a serial communications
4 ; line that will be used as a dedicated CL line.
5 ; The CLDEF macro is similar to a LINDEF and can occur inside or outside
6 ; of a MUXDEF block.
7 ; The form of the CLDEF macro outside a MUXDEF block is:
8 ; CLDEF line_number,vector_address,RSR_address
9 ; The form of the CLDEF macro inside a MUXDEF block is:
10 ; CLDEF line_number,mux_line_number
11 ;
12 ; .MACRO CLDEF AIOLN,ARG1,ARG2
13 ;
14 ; Check to make sure the CL unit number is valid
15 ;
16 IOLN = AIOLN
17 .IF GE <IOLN-CLTOTL>
18 .ERROR ;0 CL unit number exceeds # declared CL units
19 IOLN = 0
20 .ENDC
21 ;
22 ; See if this CL unit has already been assigned to another line
23 ;
24 .IF NDF CLUD'AIOLN
25 CLUD'AIOLN = 1
26 .ENDC
27 .IF GT <CLUD'AIOLN-2>
28 .ERROR ;CL unit AIOLN used more than once
29 .ENDC
30 CLUD'AIOLN = CLUD'AIOLN+1
31 ;
32 ; Set flag saying we are doing an CLDEF definition and then invoke LINDEF.
33 ; Note, the LINEND macro will reset IOLFLG.
34 ;
35 IOLFLG = 1 ;We are inside CLDEF
36 LINDEF ARG1 ARG2
37 .ENDM CLDEF
38 ; -----
39 ; The LINDEF macro begins a line definition block.
40 ; A line definition block is required for each primary
41 ; (real) line. A line definition block begins with
42 ; a LINDEF macro call, may include other macro calls
43 ; such as LFLAGS and must end with a LINEND macro call.
44 ; there are two arguments to the LINDEF macro:
45 ; Arg 1 is the input interrupt vector address or mux line #.
46 ; Arg 2 is the address of the receiver status register.
47 ; Arg 3 is 'OPERATOR' to specify line is control terminal.
48 ;
49 000000 LN = 0 ;Current line number
50 000000 BO = 0 ;1 if inside LINDEF block
51 000000 LX = 0 ;Line index number
52 000000 IOLFLG = 0 ;1 if inside an CLDEF block
53 000000 IOLN = 0 ;Number of dedicated CL line
54 000000 NPLDF = 0 ;Number of declared primary T/S lines
55 000000 NCLDF = 0 ;Number of CL lines that have been defined
56 000000 NDLDF = 0 ;Number of declared detached jobs
57 ;

```

```

58          .MACRO  LINDEF  AINTAD, ARSR, AOPR
59          .IF      NE BO          ;SEE IF LAST BLOCK LEFT OPEN
60          .ERROR  1; Missing LINEND on last line
61          LINEND          ;CLOSE OFF PREVIOUS BLOCK
62          .ENDC
63          BO      =      1          ;SAY WE'RE INSIDE A BLOCK
64          NAMDON  = 0          ;SAY NO NAME DECLARED YET
65          CMFDON  = 0          ;SAY NO SUCF DECLARED YET
66          ;
67          ; Update current line #
68          ; and make sure we don't overflow tables
69          ;
70          .IF      EQ, IOLFLG      ;If not inside an CLDEF block
71          LN      =      LN+1      ;Line counter
72          NPLDF   =      NPLDF+1   ;Count number of primary lines
73          LX      =      LX+2      ;Line index
74          CLX     =      LX
75          .IF      GT <LN-NPL>
76          .ERROR  2; More lines than declared with TBLDEF
77          .MEXIT
78          .ENDC ;GT <LN-NPL>
79          .IFF      ;EQ, IOLFLG      ;If inside an CLDEF block
80          CLX     =      FSTIOL+<2*NCLDF>;Get line index # of this line
81          NCLDF   =      NCLDF+1   ;Count # of dedicated CL lines
82          .IF      GT <NCLDF-NIOL> ;Don't exceed # CL lines declared in TBLDEF
83          .ERROR  0; More CL lines than declared in TBLDEF
84          .MEXIT
85          .ENDC ;GT <NCLDF-NIOL>
86          S      =      .
87          .      =      LCLUNT+CLX   ;Store CL unit # into table for this line
88          .WORD  2*IOLN
89          .      =      S
90          .ENDC
91          ;
92          ; *** Do this for DL11 lines only ***
93          ;
94          .IF      EQ CURMX        ;True if not within mux definition block
95          CURMXL  =      0
96          ;
97          ; Set up interrupt vector addresses
98          ;
99          .IF      B AINTAD
100         .IF      EQ, IOLFLG
101         .ERROR  3; Missing interrupt address (arg 1)
102         .IFF
103         .ERROR  3; Missing interrupt address (arg 2)
104         .ENDC
105         .MEXIT
106         .ENDC
107         VECCHK  AINTAD, 7
108         S      =      .
109         .      =      INVEC+CLX
110         .WORD  AINTAD
111         .      =      S
112         ;
113         ; Set up DL11 register addresses
114         ;

```

```

115         .IF      B ARSR
116         .IF      EQ,IOLFLG
117     .ERROR 4 ; Missing receiver register address (arg 2)
118         .IFF
119     .ERROR 4; Missing receiver register address (arg 3)
120         .ENDC
121         .MEXIT
122         .ENDC
123     SRCHK   ARSR
124     S       =
125         =      RSR+CLX
126         .WORD  ARSR
127         =      RBR+CLX
128         .WORD  ARSR+2
129         =      TSR+CLX
130         .WORD  ARSR+4
131         =      TBR+CLX
132         .WORD  ARSR+6
133         .IF      NB AOPR           ;SEE IF THIS IS CONTROL TERMINAL
134         =      CTRLTT           ;REMBER CONTROL TERMINAL #
135         .WORD  CLX
136         .ENDC
137         =      S
138     ;
139     ;   *** Do this for DZ11 and DH11 lines only ***
140     ;
141         .IFF
142     S       =
143         =      LMXNUM+CLX       ;MUX UNIT NUMBER
144         .WORD  CURMX
145         .IF      B AINTAD
146     .ERROR 0;Missing multiplexer line number
147     CURMXL =      0
148         .IFF
149     CURMXL =      AINTAD
150         .ENDC
151         =      LMXLN+CLX       ;LINE WITHIN MUX
152         .WORD  CURMXL
153         .IF      NB ARSR
154         =      CTRLTT
155         .WORD  CLX
156         .ENDC
157         =      S
158         .ENDC
159     ;
160     ;   *** Do this for all lines ***
161     ;
162     S       =
163         =      LCDTYP+CLX       ;Communications device type index
164         .WORD  CURCDX
165         =      S
166     ;
167     ;   Establish default values in case user doesn't specify
168     ;   them inside line definition block.
169     ;
170     DFLAGS =      NRMFLG       ;DEFAULT LINE CONTROL FLAGS
171         .IF      EQ,IOLFLG       ;If this is not an CLDEF

```

```

172     DIS      =      DINSPC      ; INPUT BUFFER SIZE
173     DOS      =      DOTSPC      ; OUTPUT BUFFER SIZE
174     . IFF
175     DIS      =      0            ; Input ring buffer size
176     DOS      =      CLORSZ      ; Output ring buffer size
177     . IIF    LE,DOS  DOS = 32.   ; Don't allow <= 0 size
178     . ENDC
179
180     ; Establish default values for character silos
181
182     SILSIZ   =      0
183     SILXOF   =      0
184     SILXON   =      0
185
186     . ENDM   LINDEF

```

```

1      ; -----
2      ; The FLAGS macro is used to set flags in the ILSW2 table.
3      ; The one argument to flags is the value to be stored
4      ; in ILSW2.
5      ;
6      .MACRO  FLAGS  AFLG
7  DFLAGS =      AFLG      ;SAVE FOR LINEND
8      .ENDM   FLAGS

```

```

9      ; -----
10     ; The TRMTYP macro is used to declare the terminal type.
11     ;
12     .MACRO  TRMTYP  ATYP
13     .IF    EQ,IOLFLG      ;Do not do for CL lines
14     S      =
15     .      =      ITRMTP+CLX
16     .WORD  ATYP
17     =      S
18     .ENDC
19     .ENDM   TRMTYP

```

```

20     ; -----
21     ; The NAME macro is used within a line definition block to declare
22     ; a commentary name for the line which is displayed with the
23     ; SHOW TERMINALS keyboard command.
24     ;
25     ;

```

```

26 000000 .CSECT  NAMSEC
27 000000 NAMSEC:
28 002424 .CSECT  TSGEN
29          .MACRO  NAME  NAMSTR
30          .CSECT  NAMSEC
31  NAMPTR =
32          .ASCIZ  \NAMSTR\
33  LNMTOP =
34          ;
35          .CSECT  TSGEN
36  S      =
37          =      LNAME+CLX
38          .WORD  NAMPTR
39          =      S
40  NAMDON = 1
41          .ENDM   NAME

```

```

42     ; -----
43     ; The BUFSIZ macro is used to set the size of
44     ; the input and output character buffers.
45     ; Arg 1 = Input buffer size (# of characters)
46     ; Arg 2 = output buffer size (# of characters)
47     ;
48     ;

```

```

49     .MACRO  BUFSIZ  AIS,ADS
50  DIS    =      AIS
51     .IF    NB ADS
52  DOS    =      ADS      ;SET OUTPUT BUFFER SIZE
53     .ENDC
54     .ENDM   BUFSIZ

```

```

55     ; -----
56     ; The SILO macro is used to set up information about the
57     ;

```

```

58      ; terminal input character silo.
59      ; Arg 1 = Size of the silo buffer.
60      ; Arg 2 = Free space remaining when XOFF is to be sent.
61      ; Arg 3 = Number of chars remaining when XON is to be sent.
62      ;
63      .MACRO  SILO      ASIZ, AXOF, AXON
64      SILSIZ =          ASIZ
65      SILXOF =          AXOF
66      SILXON =          AXON
67      .ENDM   SILO
68      ;
69      ;-----
70      ; The PAGE macro is used to establish the number
71      ; of lines on a page.
72      ;
73      .MACRO  PAGE  ALINE
74      .ENDM   PAGE
75      ;
76      ;-----
77      ; The CMDFIL macro is used to declare a command file which
78      ; is to be executed when the line is started.
79      ;
80      .CSECT  CMFSEC
81      CMFSEC:
82      .CSECT  TSGEN
83      .MACRO  CMDFIL  ARG
84      .IF    EQ, IOLFLG      ; Only do for non-CL lines
85      .CSECT  CMFSEC
86      NAMPTR = .
87      .ASCIZ  /ARG/
88      CMFTOP = .
89      .CSECT  TSGEN
90      S      = .
91      .      =          LSUCF+CLX
92      .WORD  NAMPTR
93      .      =          S
94      .ENDC
95      CMFDON = 1
96      .ENDM   CMDFIL
97      ;
98      ;-----
99      ; The LINPRM macro is used to specify parameters
100     ; for lines.
101     ; There are three parameters:
102     ; 1. Speed select code.
103     ; 2. Even (0) / Odd (1) parity (No longer used).
104     ; 3. One (0) or two (1) stop bits
105     ;
106     .MACRO  LINPRM  ASPD, APAR, ASTOP
107     PERR   =          0
108     .IIF   GT, ASPD-17      PERR=1
109     .IIF   GT, APAR-2       PERR=1
110     .IIF   LT, ASTOP-1      PERR=1
111     .IIF   GT, ASTOP-2      PERR=1
112     .IF    NE, PERR
113     .ERROR ;Invalid speed, parity or stop-bits parameter in LINPRM
114     .ENDC

```

```

115 XPAR = 100
116 CARLEN = 20
117 .IF GT,<APAR-1>
118 XPAR = 0
119 CARLEN = 30
120 .ENDC
121 LSTPRM = <ASPD*400>
122 .ENDM LINPRM
123 ;
124 ;-----
125 ; The SPEED macro is used to specify baud rates for lines
126 ; as well as number of data bits and parity selection.
127 ; The default is 9600 baud with 8. data bits and no parity.
128 ; The form of the macro is:
129 ;
130 ; SPEED speedcode,data_bits,parity
131 ;
132 ; where speedcode is selected from the speed code table
133 ; and is of the form S9600, for example
134 ; and data_bits = 7 or 8.
135 ; and parity = EVEN, ODD or NONE
136 ;
137 007000 LSTPRM = <S9600*400>!<20000*0>!NONE ;Default to 4800,8,N
138 ;
139 .MACRO SPEED SPDCOD,NBITS,PARCOD
140 .IF DF,S'SPDCOD
141 SPDVAL = S'SPDCOD
142 .IFF ;DF,S'SPDCOD
143 .IF DF,SPDCOD
144 SPDVAL = SPDCOD
145 .IFF ;DF,SPDCOD
146 .ERROR 0;Invalid speed specified with SPEED macro
147 SPDVAL = 14.
148 .ENDC ;NDF,S'SPDCOD
149 .ENDC ;DF,S'SPDCOD
150 .IF B,NBITS
151 NDBITS = 8.
152 .IFF ;B,NBITS
153 NDBITS = NBITS
154 .IF LT,<NDBITS-7>
155 .ERROR ;SPEED macro only accepts 7 or 8. data bits
156 NDBITS = 8.
157 .ENDC ;LT,<NDBITS-7>
158 .IF GT,<NDBITS-8.>
159 .ERROR ;SPEED macro only accepts 7 or 8. data bits
160 NDBITS = 8.
161 .ENDC ;GT,<NDBITS-8.>
162 .ENDC ;B,NBITS
163 .IF B,PARCOD
164 PARITY = NONE
165 .IFF ;B,PARCOD
166 PARITY = PARCOD
167 .IF NE,PARITY ;NOT NONE?
168 .IF NE,<PARITY-EVEN> ;NOT EVEN?
169 .IF NE,<PARITY-ODD> ;NOR ODD?
170 .ERROR ;Parity must be EVEN, ODD or NONE in SPEED macro
171 PARITY = NONE

```

172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205

```

        . ENDC      ; NOT ODD
        . ENDC      ; NOT EVEN
        . ENDC      ; EVEN OR ODD
        . ENDC      ; B, PARCOD
LSTPRM = <SPDVAL*400>!<20000*<B. -NDBITS>>!PARITY
        . ENDM      SPEED
;
;-----
; The VECCHK macro is called to see if a line vector
; address is reasoanble.
;
        . MACRO    VECCHK  VA, MASK
VERR=0
        . IIF      LT, VA-60          VERR=1
; Although RT-11 V5.3 reserves 470 and 474, we do not (for now).
        . IIF      GE, VA-500        VERR=1
        . IIF      NE, VA&MASK       VERR=1
        . IF       NE, VERR
        . ERROR    ; Invalid vector address for this line
        . ENDC
        . ENDM      VECCHK
;
;-----
; SRCHK macro checks the validity of a receiver status
; register address
;
        . MACRO    SRCHK  SR
SRERR=0
        . IIF      LT, SR-160000     SRERR=1
        . IIF      NE, SR&7          SRERR=1
        . IF       NE, SRERR
        . ERROR    ; Invalid status register address for this line
        . ENDC
        . ENDM      SRCHK
    
```

```

1      ; -----
2      ; The LINEND macro is used to close out a line
3      ; definition block.
4      ;
5      .CSECT  SASECT
6      SASECT:
7      .CSECT  TSGEN
8      .MACRO  LINEND
9      ; Make sure we're inside a line def block.
10     .IF     EQ BO
11     .ERROR 6 ; Missing LINDEF for this line
12     BO     = 0
13     .MEXIT
14     .ENDC
15     BO     = 0 ;END LINDEF BLOCK
16     ;
17     ; Make sure NAME and CMDFIL reserve at least 1 byte
18     .IF     EQ,NAMDON
19     NAME    <>
20     .ENDC
21     .IF     EQ,CMFDON
22     CMDFIL  <>
23     .ENDC
24     ;
25     ; Define input and output character buffer sizes for line
26     ; Define input buffer
27     S       =
28     .       = LINSIZ+CLX
29     .WORD   DIS-1 ;DEFINE BUFFER SIZE
30     =       S
31     ; Define output buffer
32     S       =
33     .       = LOTSIZ+CLX
34     .WORD   DOS ;DEFINE BUFFER SIZE
35     =       S
36     ;
37     ; Define table for user defined activation characters.
38     ;
39     .IF     EQ,IOLFLG ;Only do for non-CL lines
40     S       =
41     .CSECT  SASECT
42     T       =
43     .BLKB   MXSPAC
44     .CSECT  TSGEN
45     .       = LSPACT+CLX
46     .WORD   T
47     =       S
48     .ENDC ;End conditional (EQ,IOLFLG)
49     ;
50     ; Items for primary lines only.
51     ;
52     LF      = 0 ;Assume this is not a primary or CL line
53     .IF     NE,IOLFLG ;If doing a CL line definition
54     LF      = 1 ;Treat like primary line
55     .IFF    ;If not doing a CL line definition
56     .IF     LE <LN-NPL> ;Do only if this is a primary line
57     LF      = 1 ;Gen code

```

```

58          .ENDC          ;End conditional (LE <LN-NPL>)
59          .ENDC          ;End conditional (NE, IOLFLG)
60          ;
61          .IF      NE,LF      ;Do if primary line or CL line
62          ;
63          ; Define line control flags
64          ;
65          S      =      .
66          .      =      ILSW2+CLX
67          .WORD  DFLAGS      ;SET THE FLAGS
68          .      =      S
69          ;
70          ; Defile silo buffer size information
71          ;
72          S      =      .
73          .      =      LHIRBA+CLX      ;Silo size
74          .WORD  SILSIZ
75          .      =      LHIRBC+CLX      ;XOFF/XON control info
76          .BYTE  SILXOF      ;XOFF point
77          .BYTE  SILXON      ;XON point
78          .      =      S
79          ;
80          ; Define line parameters (Required for DZ11 & DH11 lines, optional for DL11)
81          ;
82          .IF      NDF,LSTPRM
83          LSTPRM =      0
84          .IF      NE CURMX
85          .ERROR  0;Missing SPEED macro call
86          .ENDC          ;End conditional (NE CURMX)
87          .ENDC          ;End conditional (NDF,LSTPRM)
88          S      =      .
89          .      =      LMXPRM+CLX
90          .WORD  <LSTPRM>!<CURMX>
91          .      =      S
92          ;
93          ; Define subprocess table.
94          ;
95          .IF      EQ,IOLFLG      ;Do if not inside CLDEF block
96          .IF      NE MAXSEC
97          S      =      .
98          .      =      LSECPT+CLX
99          .WORD  S
100         .      =      S
101         .REPT  MAXSEC
102         .BYTE  0
103         .ENDR
104         .EVEN
105         .ENDC          ;End conditional (NE MAXSEC)
106         .ENDC          ;End conditional (EQ, IOLFLG)
107         ;
108         ; Define chracter translation table
109         ;
110         .IF      NE MXTTCT
111         S      =      .
112         .      =      LCXTBL+CLX
113         .WORD  S
114         .      =      S

```

```

115             .REPT  MXTTCT+1
116             .WORD  0
117             .ENDR
118             .EVEN
119             .ENDC             ;End conditional (NE MXTTCT)
120             ;
121             .ENDC             ;End conditional (NE,LF)
122             ;
123             ; Reset flag that says we are inside an CLDEF block
124             ;
125             IOLFLG =      0      ;No longer inside an CLDEF block
126             .ENDM  LINEND
127             ;
128             ;-----
129             ; The CLEND macro is like the LINEND macro except it is used to
130             ; terminate a communication line definition started with a CLDEF macro.
131             ;
132             .MACRO  CLEND
133             LINEND
134             .ENDM  CLEND
135             ;
136             ;-----
137             ; The DETACH macro is used to define a start-up command file
138             ; To be run on a detached line when TSX-Plus is started.
139             ; The one argument to DETACH is the name of the command file.
140             ;
141             .MACRO  DETACH  NAME
142             .IF    NE      BO
143             .ERROR 1; Missing LINEND on last line
144             LINEND
145             .ENDC
146             NDLDf =      NDLDf+1      ;Count number of detached jobs
147             LN    =      LN+1
148             LX    =      LX+2
149             CLX   =      LX
150             .IF    LE, <NDL>
151             .ERROR 2; DETACH macro declared with no detached lines
152             .MEXIT
153             .ENDC
154             .IF    GT, <LN-NPL-NDL>
155             .ERROR 2; More lines than declared with TBLDEF
156             .MEXIT
157             .ENDC
158             ; Store startup command file name
159             S     =
160             .     =      LSUCF+CLX
161             .WORD S
162             .     =      S
163             .IF    NB, <NAME>
164             .ASCIZ /NAME/
165             .ENDC
166             .REPT <DETCBS+1-<-S>>
167             .BYTE 0
168             .ENDR
169             .EVEN
170             .ENDM  DETACH
171             ;

```

172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214

000000

```

-----
; The SYSPS macro is used to define a system password which may be
; required to be entered for some lines before the normal logon
; sequence begins.
;
;       SYSPSS = 0
;       .MACRO SYSPS STRING
;       SYSPSS = 1
SYPSWD: .ASCIZ \STRING\
;       .IF GT,<21.-<.-SYPSWD>>
;       .REPT <21.-<.-SYPSWD>>
;       .BYTE 0
;       .ENDR
;       .ENDC ;GT,<21.-<.-SYPSWD>>
;       .EVEN
;       .ENDM SYSPS
-----
; The RTDEF macro is used to declare information about shared
; run-time systems.
;
; The 3 arguments to RTDEF are
; 1. 12 character name of run-time system file.
; 2. R or RW indicating Read-only or Read-Write access.
; 3. Number of blocks to skip at the front of the file.
;
;       .MACRO RTDEF NAME,RFLAG,SKIP
;       .CSECT RDBSEC
T =
;       .RAD50 /'NAME'/
;       .IF NE,<<.-T>-B.>
; .ERROR 0;Run-time system name was not correctly specified
;       .ENDC
;       .WORD 0,0
;       .IF IDN,RFLAG,RW
;       .BYTE RF#WRT
;       .IFF
;       .BYTE 0
;       .ENDC
;       .BYTE SKIP
;       .CSECT TSGEN
NUMRDB = NUMRDB+1
;       .ENDM RTDEF
    
```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
;
;-----
; The DHDEF macro is used to declare the beginning
; of a block of lines which are attached to a DH11
; multiplexer. All line definition blocks up to
; the next MUXEND macro call will be connected to
; the DH11.
; There are four arguments to DHDEF:
; 1. The interrupt vector address of the mux receiver.
; 2. The address of the mux control and status register.
; 3. The interrupt vector address of the associated DM11.
; 4. The CSR address of the associated DM11.
;
; .MACRO DHDEF AVEC, ACSR, ADMVEC, ADMADR
; .IF NE CURMX
; .ERROR 1; Missing MUXEND macro
; MUXEND
; .ENDC
LSTMX = LSTMX+2
CURMX = LSTMX
CURCDX = CDX$DH ;Lines within this block are connected to DH11
DHUSE = 1 ;Set flag saying DH11 support is needed
VECCHK AVEC, 7
SRCHK ACSR
; .IF NE, ADMVEC
VECCHK ADMVEC, 3
; .ENDC
; .IF NE, ADMADR
SRCHK ADMADR
; .ENDC
S =
; = MXTYPE+CURMX ;Type of multiplexor
; .WORD CDX$DH ;Type = DH11
; = MH$SCR+CURMX ;Status and control register address
; .WORD ACSR
; = MH$RCR+CURMX ;Received character register
; .WORD ACSR+2
; = MH$LPR+CURMX ;Line parameter register
; .WORD ACSR+4
; = MH$CAR+CURMX ;Current address register
; .WORD ACSR+6
; = MH$BCR+CURMX ;Byte count register
; .WORD ACSR+10
; = MH$BAR+CURMX ;Buffer active register
; .WORD ACSR+12
; = MH$BRK+CURMX ;Break control register
; .WORD ACSR+14
; = MH$SSR+CURMX ;Silo status register
; .WORD ACSR+16
; = DM$CSR+CURMX ;DM11 Control Status register
; .WORD ADMADR
; = DM$LSR+CURMX ;DM11 Line status register
; .WORD ADMADR+2
; = MXVEC+CURMX ;DH11 Interrupt vector address
; .WORD AVEC
; = DM$VEC+CURMX ;DM11 Interrupt vector address
; .WORD ADMVEC

```

58
59

= S
.ENDM DHDEF

```

1
2
3 ; -----
4 ; The DHVDEF macro is used to declare the beginning
5 ; of a block of lines which are attached to a DHV11
6 ; multiplexer. All line definition blocks up to
7 ; the next MUXEND macro call will be connected to
8 ; the DHV11.
9 ; There are two arguments to DHVDEF:
10 ; 1. The interrupt vector address of the mux receiver.
11 ; 2. The address of the mux control and status register.
12 ;
13 ; .MACRO DHVDEF AVEC,ACSR
14 .IF NE CURMX
15 .ERROR 1; Missing MUXEND macro
16 MUXEND
17 .ENDC
18 LSTMX = LSTMX+2
19 CURMX = LSTMX
20 CURCDX = CDX$VH ;Lines within this block connected to DHV11
21 DHUSE = 1 ;Set flag saying DHV11 support is needed
22 VECCHK AVEC,7
23 SRCHK ACSR
24 S =
25 = MXTYPE+CURMX ;Type of multiplexor
26 .WORD CDX$VH ;Type = DHV11
27 = VH$CSR+CURMX ;Status and control register address
28 .WORD ACSR
29 = VH$DBR+CURMX ;Data buffer register
30 .WORD ACSR+2
31 = VH$LPR+CURMX ;Line parameter register
32 .WORD ACSR+4
33 = VH$LSR+CURMX ;Line Status Register
34 .WORD ACSR+6
35 = VH$LCR+CURMX ;Line Control Register
36 .WORD ACSR+10
37 = VH$BA1+CURMX ;Buffer Address register 1
38 .WORD ACSR+12
39 = VH$BA2+CURMX ;Buffer Address register 2
40 .WORD ACSR+14
41 = VH$BCR+CURMX ;Byte Count Register
42 .WORD ACSR+16
43 = MXVEC+CURMX ;Interrupt vector address
44 .WORD AVEC
45 = S
46 .ENDM DHVDEF
47 ; -----
48 ; The DHUDEF macro is used to declare the beginning
49 ; of a block of lines which are attached to a DHU11
50 ; multiplexer. All line definition blocks up to
51 ; the next MUXEND macro call will be connected to
52 ; the DHU11.
53 ; There are two arguments to DHUDEF:
54 ; 1. The interrupt vector address of the mux receiver.
55 ; 2. The address of the mux control and status register.
56 ;
57 ; .MACRO DHUDEF AVEC,ACSR

```

58
59

DHVDEF AVEC,ACSR ;Handle same as DHV11
.ENDM DHUDEF

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

```

;-----
; The MUXDEF macro is used to declare the beginning
; of a block of lines which are attached to a DZ11
; multiplexer. All line definition blocks up to
; the next MUXEND macro call will be connected to
; the DZ11.
; There are two arguments to MUXDEF:
; 1. The interrupt vector address of the mux receiver.
; 2. The address of the mux control and status register.
;
; .MACRO MUXDEF AVEC,ACSR
; .IF NE CURMX
; .ERROR 1; Missing MUXEND macro
; MUXEND
; .ENDC
LSTMX = LSTMX+2
CURMX = LSTMX
CURCDX = CDX%DZ ;Lines within this block are connected to DZ11
; VECCHK AVEC,7
; SRCHK ACSR
S =
; = MXTYPE+CURMX ;Multiplexor type
; .WORD CDX%DZ ;Type = DZ11
; = MXCSR+CURMX
; .WORD ACSR
; = MXLPR+CURMX
; .WORD ACSR+2
; = MXTCR+CURMX
; .WORD ACSR+4
; = MXDTR+CURMX
; .WORD ACSR+5
; = MXTBUF+CURMX
; .WORD ACSR+6
; = MXCAR+CURMX
; .WORD ACSR+7
; = MXVEC+CURMX
; .WORD AVEC
; = S
; .ENDM MUXDEF
;
; Alternate name for MUXDEF macro
;
; .MACRO DZDEF AVEC,ACSR
; MUXDEF AVEC,ACSR
; .ENDM DZDEF
;-----
; The MUXEND macro is called to declare the end of
; a set of lines connected to a DZ11 or DH11.
;
; .MACRO MUXEND
; .IF EQ CURMX
; .ERROR 6; Missing earlier MUXDEF
; .ENDC
CURMX = 0
CURCDX = CDX%DL ;Following lines are connected to DL11's

```

58

.ENDM MUXEND

```

1
2 ; -----
3 ; Symbolic equates for F (FLAGPAGE), N (NOFLAGPAGE,NOHOLD,NARROW),
4 ; W (WIDE) and H (HOLD) In SPFLAG, SPWIDE, and SPHOLD macros.
5 ;
6 ; F = 1000
7 ; N = 0
8 ; W = 2000
9 ; H = 40
10 ;
11 ; Init the values to hold flag page, wide, and hold flags.
12 ;
13 ; F1 = 0
14 ; F2 = 0
15 ; F3 = 0
16 ; F4 = 0
17 ; F5 = 0
18 ; F6 = 0
19 ; F7 = 0
20 ; F8 = 0
21 ; F9 = 0
22 ; F10 = 0
23 ; F11 = 0
24 ; F12 = 0
25 ; F13 = 0
26 ; F14 = 0
27 ; F15 = 0
28 ; F16 = 0
29 ; F17 = 0
30 ; F18 = 0
31 ; F19 = 0
32 ; F20 = 0
33 ; F21 = 0
34 ; F22 = 0
35 ; F23 = 0
36 ; F24 = 0
37 ; F25 = 0
38 ; F26 = 0
39 ; F27 = 0
40 ; F28 = 0
41 ; F29 = 0
42 ; F30 = 0
43 ;
44 ; Define the default for each spooled device as to whether or not
45 ; flag pages will be the default for that device. This macro must
46 ; be executed prior to the execution of the SPWIDE macro, the SPHOLD
47 ; macro, and the SPOOL macro.
48 ;
49 ; .MACRO SPFLAG A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, AA, BB, CC, DD
50 ; .IIF NB, <A> F1 = <A>
51 ; .IIF NB, <B> F2 = <B>
52 ; .IIF NB, <C> F3 = <C>
53 ; .IIF NB, <D> F4 = <D>
54 ; .IIF NB, <E> F5 = <E>
55 ; .IIF NB, <F> F6 = <F>
56 ; .IIF NB, <G> F7 = <G>
57 ; .IIF NB, <H> F8 = <H>
58 ; .IIF NB, <I> F9 = <I>

```

```

58 . IIF NB, <J> F10 = <J>
59 . IIF NB, <K> F11 = <K>
60 . IIF NB, <L> F12 = <L>
61 . IIF NB, <M> F13 = <M>
62 . IIF NB, <N> F14 = <N>
63 . IIF NB, <O> F15 = <O>
64 . IIF NB, <P> F16 = <P>
65 . IIF NB, <Q> F17 = <Q>
66 . IIF NB, <R> F18 = <R>
67 . IIF NB, <S> F19 = <S>
68 . IIF NB, <T> F20 = <T>
69 . IIF NB, <U> F21 = <U>
70 . IIF NB, <V> F22 = <V>
71 . IIF NB, <W> F23 = <W>
72 . IIF NB, <X> F24 = <X>
73 . IIF NB, <Y> F25 = <Y>
74 . IIF NB, <Z> F26 = <Z>
75 . IIF NB, <AA> F27 = <AA>
76 . IIF NB, <BB> F28 = <BB>
77 . IIF NB, <CC> F29 = <CC>
78 . IIF NB, <DD> F30 = <DD>
79 . ENDM SPFLAG

```

```

;
; Define the default for each spooled device as to whether or not
; flag pages will be centered based on a 132 columned or 80 columned
; page. This macro must be executed after the execution of the SPFLAG
; macro, and prior to the execution of the SPOOL macro.
;

```

```

86 . MACRO SPWIDE A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, AA, BB, CC, DD
87 . IIF NB, <A> F1 = F1 ! <A>
88 . IIF NB, <B> F2 = F2 ! <B>
89 . IIF NB, <C> F3 = F3 ! <C>
90 . IIF NB, <D> F4 = F4 ! <D>
91 . IIF NB, <E> F5 = F5 ! <E>
92 . IIF NB, <F> F6 = F6 ! <F>
93 . IIF NB, <G> F7 = F7 ! <G>
94 . IIF NB, <H> F8 = F8 ! <H>
95 . IIF NB, <I> F9 = F9 ! <I>
96 . IIF NB, <J> F10 = F10 ! <J>
97 . IIF NB, <K> F11 = F11 ! <K>
98 . IIF NB, <L> F12 = F12 ! <L>
99 . IIF NB, <M> F13 = F13 ! <M>
100 . IIF NB, <N> F14 = F14 ! <N>
101 . IIF NB, <O> F15 = F15 ! <O>
102 . IIF NB, <P> F16 = F16 ! <P>
103 . IIF NB, <Q> F17 = F17 ! <Q>
104 . IIF NB, <R> F18 = F18 ! <R>
105 . IIF NB, <S> F19 = F19 ! <S>
106 . IIF NB, <T> F20 = F20 ! <T>
107 . IIF NB, <U> F21 = F21 ! <U>
108 . IIF NB, <V> F22 = F22 ! <V>
109 . IIF NB, <W> F23 = F23 ! <W>
110 . IIF NB, <X> F24 = F24 ! <X>
111 . IIF NB, <Y> F25 = F25 ! <Y>
112 . IIF NB, <Z> F26 = F26 ! <Z>
113 . IIF NB, <AA> F27 = F27 ! <AA>
114 . IIF NB, <BB> F28 = F28 ! <BB>

```

```

115             . IIF NB, <CC>      F29 = F29 ! <CC>
116             . IIF NB, <DD>      F30 = F30 ! <DD>
117             . ENDM      SPWIDE
118
119 ; Define the default for each spooled device as to whether or not
120 ; spool files are to be eligible to be started as soon as they are
121 ; created. This macro must be executed after the execution of the
122 ; SPFLAG macro, and prior to the execution of the SPOOL macro.
123 ;
124             . MACRO      SPHOLD  A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, AA, BB, CC, DD
125             . IIF NB, <A>      F1 = F1 ! <A>
126             . IIF NB, <B>      F2 = F2 ! <B>
127             . IIF NB, <C>      F3 = F3 ! <C>
128             . IIF NB, <D>      F4 = F4 ! <D>
129             . IIF NB, <E>      F5 = F5 ! <E>
130             . IIF NB, <F>      F6 = F6 ! <F>
131             . IIF NB, <G>      F7 = F7 ! <G>
132             . IIF NB, <H>      F8 = F8 ! <H>
133             . IIF NB, <I>      F9 = F9 ! <I>
134             . IIF NB, <J>      F10 = F10 ! <J>
135             . IIF NB, <K>      F11 = F11 ! <K>
136             . IIF NB, <L>      F12 = F12 ! <L>
137             . IIF NB, <M>      F13 = F13 ! <M>
138             . IIF NB, <N>      F14 = F14 ! <N>
139             . IIF NB, <O>      F15 = F15 ! <O>
140             . IIF NB, <P>      F16 = F16 ! <P>
141             . IIF NB, <Q>      F17 = F17 ! <Q>
142             . IIF NB, <R>      F18 = F18 ! <R>
143             . IIF NB, <S>      F19 = F19 ! <S>
144             . IIF NB, <T>      F20 = F20 ! <T>
145             . IIF NB, <U>      F21 = F21 ! <U>
146             . IIF NB, <V>      F22 = F22 ! <V>
147             . IIF NB, <W>      F23 = F23 ! <W>
148             . IIF NB, <X>      F24 = F24 ! <X>
149             . IIF NB, <Y>      F25 = F25 ! <Y>
150             . IIF NB, <Z>      F26 = F26 ! <Z>
151             . IIF NB, <AA>     F27 = F27 ! <AA>
152             . IIF NB, <BB>     F28 = F28 ! <BB>
153             . IIF NB, <CC>     F29 = F29 ! <CC>
154             . IIF NB, <DD>     F30 = F30 ! <DD>
155             . ENDM      SPHOLD
156
157 ; The following 3 macros are used within the SPOOL macro to make it
158 ; possible for the user to set or not to set the flag page, flag page
159 ; centering, and spool hold flags on a per device basis.
160 ;
161             . MACRO      BLANK   NUM
162             . WORD      F'NUM          ; SDFLAG
163             . ENDM      BLANK
164
165             . MACRO      NOT     NUM
166             . WORD      F'NUM & ^CH  ; SDFLAG
167             . ENDM      NOT
168
169             . MACRO      OR      NUM
170             . WORD      F'NUM ! H     ; SDFLAG
171             . ENDM

```

```

172 ; -----
173 ; The SPOOL macro is used to declare those devices which
174 ; are to be spooled by TSX-Plus (such as line printers).
175 ; There are seven arguments to spool:
176 ; 1) Number of devices to be spooled (may be zero)
177 ; 2) Number of spool files allowed to be open.
178 ; 3) Number of buffers for spooler to use.
179 ; 4) Number of blocks in spool disk file.
180 ; 5) List of 3 character names of devices to be spooled.
181 ; 6) 0 for 'nohold' mode, 1 for 'hold' mode.
182 ; 7) # of blocks which will be remembered for back up.
183 ;
184 ; .MACRO SPOOL SND, SNF, SNB, SNDB, SNAM, SHLD, SNBU
185 ; Define number of spooled devices
186 SPLND = SND
187 SNBUX = SNBU
188 .IF EQ, SNBU
189 SNBUX = 1
190 .ENDC
191 PVSPBL = SNBUX+10. ;# PRIVATE BLOCKS PER DEV
192 SNDBX = SNDB ;TOTAL # OF SPOOL BLOCKS
193 .IF LT <SNDB-<SND*PVSPBL>-2>
194 SNDBX = <SND*PVSPBL>+2
195 .ENDC
196 .IF GT, SPLND
197 ;**
198 ;** Assemble this code if there are spooled devices.
199 ;**
200 SPLNF = SNF ;DEFINE # OF SPOOL FILES
201 SPLNB = SNB ;DEFINE # OF SPOOL BUFFERS
202 NESB: .WORD SPLNB
203 ; DEFINE SPOOL BUFFERS
204 .IF EQ, SPLNB ;THERE MUST BE AT LEAST 1 BUFFER
205 .ERROR ;There must be at least 1 buffer for spooler
206 SPLNB = 1 ;FORCE 1 BUFFER
207 .ENDC
208 SPLBHD: .WORD 0 ;HEAD OF FREE BUFFER CHAIN
209 .IF EQ, SPLNF ;Make sure we have at least 1 file
210 .ERROR ;There must be at least 1 spool file
211 SPLNF = 1 ;FORCE 1 FILE
212 .ENDC
213 ;
214 ; Define spool device control blocks (SDCB)
215 ;
216 SDCB:
217 C = 0
218 .REPT SPLND
219 .WORD 0,0,0,0,0 ;SDCHAN
220 .WORD 0,0,0,0,0,0
221 .WORD 0,0,0,0
222 ; INITIAL FORM NAME
223 .ASCII /STD / ;SDFORM
224 .WORD 0,0 ;SDANAM
225 ; GEN INIT FLAGS
226 C = C + 1
227 .IF B, SHLD
228 BLANK \C

```

```

229             . IFF
230             . IF    EQ, SHLD
231             NOT \C
232             . IFF
233             OR \C
234             . ENDC
235             . ENDC
236             . WORD  0                ; SDSKIP
237             . WORD  PVSPBL           ; SDFRBL
238 ; GEN BACKUP CELLS
239             . REPT  SNBUX
240             . WORD  0
241             . ENDR
242             . WORD  0                ; SDBULS = END OF SDBU
243             . ENDR
244 SDCBND:                ; END OF SDCB AREA
245 ; DEFINE SIZE OF SDCB
246 SDCBSZ = 48. + <2*SNBUX>
247 SDBULS = SDCBSZ-4.
248 ;
249 ; Define table of device names.
250 ;
251 SPLDEV: .RAD50 // 'SNAM'//          ; DEFINE TABLE OF NAMES
252         . EVEN
253 SPLDVN:                ; END OF TABLE
254         . IF    NE, <<SPLDVN-SPLDEV>-<2*SND>>
255         . ERROR ; Number of spooled devices not equal to number of names
256         . ENDC
257 SPLANM: .ASCII // 'SNAM'//
258         . EVEN
259 ; Reserve space for spool file channel block
260 SPLCHN: .BLKW 5
261 ;
262         . IFF
263 ; **
264 ; ** This code is assembled if there are no spooled devices.
265 ; **
266 SPLND = 0
267 SPLNF = 0
268 SPLNB = 0
269 SDCBSZ = 1
270 SDBULS = 1
271 ;
272 SPLDEV:
273 SPLDVN:
274 SPLANM:
275 SPLBHD:
276 SPLCHN:
277 SDCB:
278 SDCBND:
279 NESB:
280         . WORD  0                ; SAY ALL LISTS ARE EMPTY
281 ;
282         . ENDC
283         . ENDM  SPOOL

```

```

1          ; =====
2          ; The TSX-Plus system manager alters values in the following
3          ; section to customize the system for a particular configuration.
4          ;
5          ; System parameters:
6          ;
7          ; Swap file device-file specification (do not place on VM).
8          ;
9 002424 075250 100020 075150 SWDBLK: .RAD50 /SY TSXSWPTSX/
10 002432 100020
11          ;
12          ; Spool file device-file specification (do not place on VM).
13          ;
14 002434 075250 100020 074514 SPLBLK: .RAD50 /SY TSXSPLTSX/
15 002442 100020
16          ;
17          ; PLAS region swap file specification (do not place on VM).
18          ;
19 002444 075250 100020 071576 RSFBLK: .RAD50 /SY TSXRSFTSX/
20 002452 100020
21          ;
22          ; File spec for file used to hold user defined command definitions (UCL)
23          ;
24 002454 075250 100020 101704 UCLDAT: .RAD50 /SY TSXUCLTSX/
25 002462 100020
26          ;
27          ; File spec for temp file used while processing IND command files
28          ;
29 002464 075250 100020 035164 INDFIL: .RAD50 /SY TSXINDTSX/
30 002472 100020
31          ;
32          ; Maximum amount of memory that can be used by any job (# K bytes).
33          ; This value must not exceed 64. (Kb)
34          ;
35          ; HIMEM = 64. ;Max memory that any job may use
36          ;
37          ; Default memory size for jobs that will be in effect when the job
38          ; logs on. (Specify in # K bytes).
39          ;
40          ; DFLMEM = 64. ;Default memory limit for jobs
41          ;
42          ; SWAPFL controls whether TSX-Plus is allowed to swap jobs to disk if
43          ; insufficient memory is available to hold all active users.
44          ; The normal case (SWAPFL=1) allows TSX-Plus to do job swapping.
45          ; SWAPFL can be set to 0 (zero) in special situations such as when a
46          ; small number of lines are being supported on a floppy disk based system
47          ; that does not have room for a swap file.
48          ; If SWAPFL is set to zero the following actions occur:
49          ; 1. No disk swap file is created.
50          ; 2. A line will not be allowed to log on if there is insufficient
51          ; free memory space to support it.
52          ; 3. Each job is allocated a memory size equal to DFLMEM (default job
53          ; memory size).
54          ; 4. The MEMORY command cannot be used to change the job size.
55          ;
56          ; SWAPFL = 1 ;1==>Allow job swapping; 0==>Do not swap.
57          ;

```

```

53      ; If the system is generated with job swapping enabled (SWAPFL=1), then
54      ; the SWPSLT parameter controls the number of job slots allocated
55      ; in the swap file. SWPSLT should be in the range 0 up to the
56      ; total number of jobs. If SWPSLT is set to zero, TSX-Plus will
57      ; automatically allocate one job slot in the swap file for each job.
58      ; SWPSLT may be set to a value less than the total number of jobs if
59      ; a small amount of job swapping is anticipated; however, a system
60      ; crash will occur if the system needs to swap a job out of memory
61      ; and no free slot is available in the swap file.
62      ; The SWPSLT parameter has no effect on non-swapping systems (SWAPFL=0).
63      ; The recommended setting for this parameter is 0 (zero).
64      ;
65      000012 SWPSLT =      10.      ;Number of job slots in swap file
66      ;
67      ; Number of 512-byte blocks to allocate for swap file that is used
68      ; for extended memory PLAS (Program's Logical Address Space) regions
69      ; that are used by jobs that have virtual overlays or virtual arrays.
70      ; Note that this is the total space in the PLAS swap file for all
71      ; extended memory regions in use at any time by all jobs.
72      ; Note: In a non-swapping system (SWAPFL=0), SEGBLK must be non-zero
73      ; if PLAS support is wanted, but its value does not matter.
74      ;
75      000764 SEGBLK =      500.     ;# blocks for PLAS swap file
76      ;
77      ; Number of shared global PLAS regions that can be created by all jobs.
78      ;
79      000014 NGR      =      12.     ;Number of global PLAS regions
80      ;
81      ; BUSTYP defines the machine bus structure for TSX-Plus. There are two
82      ; possible machine bus structures supported by TSX-Plus - the QBUS (LSI)
83      ; and the UNIBUS. Select one of these parameters below to specify the
84      ; bus support desired. Use the following information for choosing the
85      ; correct bus structure.
86      ;
87      ; QBUS   - 11/23, 11/23-Plus, 11/73, and Professional.
88      ; UNIBUS - 11/24, 11/34a, 11/44, and 11/60.
89      ;
90      000001 BUSTYP =      QBUS   ;Specify machine bus structure (UNIBUS/QBUS)
91      ;
92      ; Memory upper limit size specification expressed in number of k-bytes.
93      ; This parameter controls the maximum memory available for TSX-Plus
94      ; system use. Memory above this upper limit will not be used by the
95      ; operating system.
96      ; If the MEMSIZ parameter is set to 0 (zero), TSX-Plus will use all
97      ; available memory on the machine. To disable the use of extended
98      ; memory, set MEMSIZ to 248 or less.
99      ;
100     000000 MEMSIZ =      0.      ;Upper memory limit
101     ;
102     ; The INIABT parameter controls the action taken by TSX-Plus when
103     ; certain errors are detected during system initialization.
104     ; If INIABT=0, TSX-Plus ignores the error and continues running.
105     ; If INIABT=1, TSX-Plus aborts initialization and prints an error message.
106     ;
107     ; *****
108     ; ** The normal and recommended setting for      **
109     ; ** this parameter is INIABT=1. It is cleared **

```

```

110      ; ** for default installation.                **
111      ; *****
112      ;
113      ; The following initialization errors are controlled by the INIABT flag:
114      ; 1. A device that was specified in TSGEN does not have a
115      ;    TSX-Plus handler on the system disk.
116      ; 2. A time sharing line that was generated into TSX-Plus is not
117      ;    installed on the machine.
118      ; 3. A shared run-time system file could not be found during startup.
119      ;
120      000000 INIABT = 0 ;0==>Continue on error, 1==>Abort on error
121      ;
122      ; The UXIFLG parameter controls the action taken by TSX-Plus when
123      ; an interrupt occurs at an unexpected location. Unexpected interrupts
124      ; may occur if the interrupt vector address specified in a device
125      ; handler does not match the actual interrupt address for which the
126      ; device has been set. Unexpected interrupts can also occur if real-time
127      ; interrupts occur and no connection has been established between the
128      ; real-time interrupt and a TSX-Plus real-time program.
129      ;
130      ; If UXIFLG is set to 1 (one) then unexpected interrupts cause a system
131      ; crash with the error message:
132      ;   ?TSX-F-UEI-Interrupt occurred at unexpected location
133      ;   Argument value = xxxx
134      ; Where "xxxx" is the address at which the interrupt occurred.
135      ;
136      ; If UXIFLG is set to 0 (zero) then unexpected interrupts are ignored
137      ; by the system and do not cause a crash or print an error message.
138      ;
139      ; The recommended setting for UXIFLG is 1 (one).
140      ;
141      000000 UXIFLG = 0 ;Unexpected interrupt control flag
142      ;
143      ; Parameters related to the TSX-Plus system crash dump facility.
144      ; This optional facility will print some useful internal system
145      ; data if a system crash occurs. The dump information can be printed
146      ; on any terminal connected to a DL-11 type line (including DLV-11)
147      ; or on a parallel printer port.
148      ; It is recommended that this facility not be included in the system
149      ; unless you are experiencing system crashes.
150      ;
151      ; Set SYSDMP to 1 if you want the crash dump facility, 0 if not.
152      ;
153      000000 SYSDMP = 0 ;1==>Enable crash dump, 0==>No crash dump
154      ;
155      ; Address of transmitter control register for device to which crash
156      ; dump is to be written. This must be a DL-11 type device controller
157      ; or a parallel printer controller. It is valid to use either the
158      ; transmitter or receiver CSR.
159      ; Specify 177560 or 177564 to dump on the console terminal.
160      ; Specify 177510 or 177514 to dump to line printer connected to standard
161      ; parallel port.
162      ;
163      177564 DMPTCR = 177564 ;Transmitter control reg for dump device
164      ;
165      ; Set DMPKTP to 1 if you want a system crash to occur any time a trap
166      ; occurs within the system. Set it to 0 (zero) if you want recoverable

```

```

167      ; traps within the system to abort the job but continue execution of the
168      ; system.
169      ;
170      000000      DMPKTP =      0      ;1==>Always crash on traps within system
171      ;
172      ; The IOABT parameter controls the action taken by TSX-Plus when
173      ; a job terminates execution. If IOABT=0, TSX-Plus will wait for
174      ; all outstanding I/O pending for the job to complete before the job
175      ; is actually terminated. If IOABT=1, TSX-Plus will call the handler
176      ; abort entry point for all outstanding I/O pending for the job.
177      ; Note, the "SET IO [NO] ABORT" keyboard command may be used to
178      ; change the value of this parameter.
179      ;
180      000001      IOABT =      1      ;0==>I/O rundown, 1==>I/O abort
181      ;
182      ; U$CL is a flag that controls whether the User Command Linkage is to
183      ; be used to allow users to define their own commands.
184      ; If U$CL is non-zero the UCL facility is enabled and users may define
185      ; their own system commands. If U$CL is zero, user defined commands
186      ; will not be supported by the system. Note: if the UCL facility is
187      ; enabled, the TSXUCL.SAV file must be placed on the system disk.
188      ;
189      000001      U$CL =      1      ;0==>No UCL program, 1==>UCL program
190      ;
191      ; Number of user-defined commands that can be stored by TSXUCL
192      ; for each job. (The number of blocks required in the SY:TSXUCL.DAT file
193      ; is approximately equal to the number of commands per job times the
194      ; total number of time-sharing lines divided by 5).
195      ;
196      000024      UCLMNC =      20.      ;Maximum user-defined commands per job
197      ;
198      ; The UCLORD parameter selects the default call order for checking
199      ; to see if a command is a user-defined command.
200      ; FIRST ==> Check for user-defined commands before system commands.
201      ; MIDDLE ==> Check after system commands but before command files.
202      ; LAST ==> Check after system commands and command files.
203      ;
204      ; Note that the SET UCL FIRST/LAST keyboard command can be used to
205      ; alter this order on a line-by-line basis.
206      ;
207      000002      UCLORD =      MIDDLE ;Select FIRST / MIDDLE / LAST
208      ;
209      ; The LDSYS flag controls whether the standard system support for
210      ; logical disks (LD) is to be provided.
211      ; If LDSYS is set to 1, system support for logical disks is included.
212      ; If LDSYS is set to 0, system support for logical disks is excluded.
213      ;
214      000001      LDSYS =      1      ;1==>Include LD support, 0==>Exclude LD.
215      ;
216      ; The SLEDIT flag controls whether the Single Line Editor (SL) facility
217      ; is to be made available to the system.
218      ; If SLEDIT is set to 1, Single Line Editor support is included.
219      ; If SLEDIT is set to 0, Single Line Editor support is omitted.
220      ; Single Line Editor support adds approximately 2Kb to the size of the
221      ; mapped portion of the system.
222      ;
223      000001      SLEDIT =      1      ;1==>Include SL support, 0==>Exclude SL

```

```

224 ;
225 ; The KEYMAX parameter specifies the number of user-defined keys supported
226 ; by the single line editor. The DEFINE/KEY command is used to associate
227 ; a user-specified text string with a function key. The maximum number
228 ; of such key definitions that may be in effect at one time for each user
229 ; is controlled by the KEYMAX parameter.
230 ; The maximum supported value for KEYMAX is 60.
231 ;
232 000007 KEYMAX = 7 ; Maximum number of user-defined keys for SL
233 ;
234 ; The MAXWIN parameter specifies the maximum number of terminal display
235 ; windows that may be in use by all jobs on the system.
236 ; If MAXWIN is set to 0 (zero), the display window feature is not included
237 ; in the system. Display windows are useful if you frequently utilize
238 ; subprocesses in that they preserve the screen context when you switch
239 ; between processes.
240 ;
241 000012 MAXWIN = 10 ; Total number of display windows for all jobs
242 ;
243 ; Set DBGFLG to 1 to cause the TSX-Plus program debugging facility
244 ; to be included with the system.
245 ; Set DBGFLG to 0 if the debugging facility is not wanted.
246 ;
247 000000 DBGFLG = 0 ; 1==>Include debugger; 0==>Exclude debugger
248 ;
249 ; Number of slots in INSTALL table to reserve for user programs.
250 ;
251 000004 NUIP = 4 ; Number of INSTALL slots for user programs
252 ;
253 ; The following time-slice values are used to schedule jobs for execution.
254 ; Each time value must be specified in 0.1 second units.
255 ;
256 ; QUANO -- Time slice for round-robin scheduling of high-priority
257 ; real-time jobs. That is, jobs with execution priorities
258 ; greater than or equal to PRIHI.
259 ;
260 000002 QUANO = 2 ; Time slice for real-time jobs
261 ;
262 ; QUAN1 -- Time that jobs will remain in a high-priority state after
263 ; they receive an activation character from the terminal.
264 ; A job is classified as "interactive" from the time when an
265 ; activation character is received until the job consumes
266 ; QUAN1 units of time, then the job is classified as "compute
267 ; bound".
268 ;
269 000024 QUAN1 = 20 ; High-priority time for interactive jobs
270 ;
271 ; QUAN1A -- Time that jobs will remain in a high-priority state after
272 ; they are activated because of I/O completion or they are
273 ; restarted following other wait states.
274 ;
275 000002 QUAN1A = 2 ; High-priority time for wait-reactivation
276 ;
277 ; QUAN1B -- Time slice used to switch between "interactive" jobs.
278 ;
279 000002 QUAN1B = 2 ; Time slice for "interactive" jobs.
280 ;

```

```

281      ; QUAN1C -- Time job will be allowed to stay in highest execution state
282      ;         after receipt of a character from the terminal.
283      ;
284      000001      QUAN1C =      1.      ;Time at highest execution state
285      ;
286      ; QUAN2 -- Time that normal priority CPU-bound jobs are allowed to run
287      ;         if there are no high-priority jobs that want to run.
288      ;         This time-slice controls round-robin scheduling of CPU-bound jobs
289      ;         with execution priority values in the range (PRILOW+1) to
290      ;         (PRIHI-1).
291      ;
292      000012      QUAN2  =      10.     ;Normal-priority CPU-bound job time-slice
293      ;
294      ; QUAN3 -- Time slice for round-robin scheduling of very low priority
295      ;         jobs. That is, jobs with priorities less than or equal
296      ;         to PRILOW.
297      ;
298      000024      QUAN3  =      20.     ;Time slice for very low priority jobs
299      ;
300      ; INTIOC -- Number of consecutive times that a job will be allowed to
301      ;         perform I/O operations following input of an activation
302      ;         character from the terminal before the job is classified
303      ;         as non-interactive.
304      ;
305      000036      INTIOC =      30.     ;Number of I/O ops. while "interactive".
306      ;
307      ; HIPRCT -- Number of consecutive times that a job will be given a
308      ;         high-priority execution boost following wait states such
309      ;         as I/O wait before the job will be scheduled as a normal
310      ;         CPU-bound job.
311      ;
312      000050      HIPRCT =      40.     ;Number of consecutive high-priority hits
313      ;
314      ; Time that job will be held in memory after being swapped in from disk.
315      ; A job is not eligible to be swapped out of memory until CORTIM has
316      ; elapsed since it was swapped into memory. However, the job becomes
317      ; immediately eligible to be swapped if it goes into a state where it is
318      ; waiting on any resource other than non-terminal I/O.
319      ; Specify in 0.1 second units.
320      ;
321      000002      CORTIM =      2.      ;Guaranteed memory-residency time
322      ;
323      ; Job priority classes: There are three groups of job priorities,
324      ; the lowest priority group ranges from a job priority 0 up to and
325      ; including the priority equal to the PRILOW parameter. Jobs with
326      ; priorities in this range execute with lower priority than all normal
327      ; time-sharing jobs.
328      ; The second range of priorities is from (PRILOW+1) up to (PRIHI-1).
329      ; Jobs in this range are treated as normal time-sharing jobs.
330      ; The third range of priorities is from PRIHI up to 127. These priorities
331      ; are for real-time jobs which will take unconditional precedence over
332      ; all other jobs.
333      ; All priority values must be in the range 0 to 127.
334      ;
335      000023      PRILOW =      19.     ;Highest "low priority" value
336      000120      PRIHI  =      80.     ;Lowest "high priority" value
337      ;

```

```

338      ; PRIDEF -- Default job priority.
339      ;
340      000062 PRIDEF =      50.      ;Default job priority
341      ;
342      ; PRIVIR -- Amount by which a job's execution priority is reduced
343      ;          when the job is disconnected from the terminal by switching
344      ;          to a subprocess. Note: this only applies to jobs with
345      ;          base priorities in the range (PRILOW+1) to (PRIHI-1).
346      ;
347      000012 PRIVIR =      10.      ;Disconnect job priority reduction
348      ;
349      ; Maximum number of subprocesses per primary process.
350      ;
351      000003 MAXSEC =      3.      ;Max subprocesses per user
352      ;
353      ; Maximum file size (# blocks) that will be returned in response to
354      ; a .ENTER request that specifies a file size of 0 blocks.
355      ;
356      001750 MAXFIL =     1000.      ;Max # blocks for default allocation
357      ;
358      ; Number of 512 byte blocks to hold in memory in a generalized data cache.
359      ; If the CACHE parameter is set to 0 (zero), data caching is not performed.
360      ; Note: The data caching facility adds approximately 2000 bytes to the
361      ; size of the unmapped portion of the system and 528*CACHE bytes to
362      ; the mapped portion of the system.
363      ; The maximum number of blocks that may be held in the cache is 4095. (2MB)
364      ;
365      000454 CACHE =      300.      ;Number of blocks in data cache
366      ;
367      ; The following parameters relate to the cache of file directory entries
368      ; maintained by TSX-Plus. This cache is used to reduce the number of disk
369      ; accesses required to do lookups on frequently accessed files.
370      ; The system disk (SY:) is automatically cached.
371      ; Other devices are only cached if they are introduced to the system
372      ; by use of the MOUNT command.
373      ;
374      ; Maximum number of units that may be cached.
375      ; This includes all logical disks (LD) and all physical disks for which
376      ; directory caching is enabled by use of the MOUNT command.
377      ; (Space required is 18 bytes per unit).
378      ;
379      000036 MAXCSH =      30.      ;Max # device units whose directories to cache
380      ;
381      ; Maximum number of file entries to be held in directory cache.
382      ; (Space required is 18 bytes per entry)
383      ;
384      000074 NMFCSH =      60.      ;Max # file entries to be cached
385      ;
386      ; Maximum number of device units that can be allocated to jobs for exclusive
387      ; use by use of the ALLOCATE command.
388      ;
389      000005 MAXALC =      5.      ;Max # units that can be allocated
390      ;
391      ; Maximum number of simultaneous requests by jobs to monitor other jobs.
392      ;
393      000005 MAXMON =      5.      ;Max # job monitoring requests
394      ;

```

```

395 ; The system password is a global password which must be entered
396 ; when a line is initiated before the normal logon sequence begins.
397 ; The use of a system password is optional and may be enabled on a
398 ; line-by-line basis by specifying the $SYSPS flag with the
399 ; FLAGS macro within the line definition blocks for the lines
400 ; for which the password will be required. If a system password is
401 ; required for a line, an exclamation point prompt is printed as the
402 ; first thing when the line is initiated. The idea is to force the
403 ; calling person to provide a password before printing the normal
404 ; logon greeting which identifies the nature and identity of the site.
405 ;
406 002474 ; SYSPS <TSX> ;System password for all lines with $SYSPS
407 ;
408 ; Amount of time a phone job is allowed to be active without
409 ; establishing carrier after its ring has been answered (by raising
410 ; DTR). After this time interval the job is automatically logged off.
411 ; Specify in 0.5 second units. This timer is inoperative if set to 0.
412 ;
413 000170 TIMIN = 120. ;Sixty seconds for normal modem connection
414 ;
415 ; Amount of time that carrier may be lost during a normal phone
416 ; connection. If carrier is re-established within the time interval
417 ; then the counter is reset. If carrier is continuously absent for
418 ; this interval, then the job is automatically logged off.
419 ; Specify in 0.5 second units. This timer is inoperative if set to 0.
420 ;
421 000170 TIMOUT = 120. ;Five seconds of lost carrier
422 ;
423 ; Amount of time that a phone job may be connected as an apparent local
424 ; job (never generated the ring signal) without establishing carrier.
425 ; After this time interval, the job is automatically logged off. (cf. PHONE)
426 ; Specify in 0.5 second units. This timer is inoperative if set to 0.
427 ;
428 000001 TIMLOC = 1. ;Normally disallow this case
429 ;
430 ; Amount of time that a phone connection will be maintained after a job
431 ; has logged off. After this time the DTR signal will be dropped,
432 ; causing the phone to be hung up.
433 ; Specify in 0.5 second units. This timer is inoperative if set to 0.
434 ;
435 000074 OFFTIM = 60. ;Allow two seconds to log back on
436 ;
437 ; Amount of time a after a ring is answered (by raising DTR) that a
438 ; phone connection will be maintained if carrier is not detected. After
439 ; this time the DTR signal will be dropped, causing the phone to be hung up.
440 ; Specify in 0.5 second units. This timer is inoperative if set to 0.
441 ;
442 000170 ONTIM = 120. ;Allow up to 1 min for modems to connect
443 ;
444 ; Modem lines ($PHONE in the LINDEF FLAGS macro) are normally
445 ; treated as phone lines if the DCD signal (carrier) is present
446 ; when the lines are started and optionally treated as local lines
447 ; if the signal is not present. The TIMLOC parameter determines how
448 ; long a phone line may be active without establishing carrier.
449 ; If you want a phone job which does not establish carrier to be
450 ; treated as a local line, set PHONE to 0. If you want an active
451 ; phone job which does not establish carrier within TIMLOC*0.5 sec

```

```

452          ; to be killed, set PHONE to 1.
453          ;
454          000000 PHONE = 0. ;$PHONE lines may be local if carrier absent
455          ;
456          ; Define Lead-in character that tells TSX-Plus that a special
457          ; terminal control sequence is coming from the program.
458          ;
459          000035 TSLICH = 035 ;Octal 35 = decimal 29.
460          ;
461          ; Define the keyboard control character that will be used to
462          ; switch to a subprocess.
463          ; (Specify the octal value of the ASCII control character)
464          ;
465          000027 VLSWCH = 027 ;Octal 27 = control-W
466          ;
467          ; Define keyboard control character used to cause the current screen
468          ; window contents to be printed.
469          ; (Specify the octal value of the ASCII control character)
470          ;
471          000002 PWCH = 002 ;Octal 02 = control-B
472          ;
473          ; Define keyboard control character that is used to generate a line
474          ; of status information on the terminal. Note that the information
475          ; is displayed like the SEND command and is not managed by the
476          ; Process Windowing system -- it goes away on screen refresh.
477          ;
478          000024 STATCH = 024 ;Octal 24 = control-T
479          ;
480          ; Define keyboard control character that is used to terminate
481          ; a cross-connection between a time-sharing line and a CL line.
482          ; (Specify the octal value of the ASCII control character)
483          ;
484          000034 CCXTRM = 034 ;Octal 34 = control-\ (control backslash)
485          ;
486          ; Define keyboard control character that is used to signal
487          ; special control functions for a time-sharing line cross-connected
488          ; to a CL line.
489          ; (Specify the octal value of the ASCII control character)
490          ;
491          000001 CCXCTL = 001 ;Octal 001 = control-A
492          ;
493          ; Define the version number to be associated with the CL handler when
494          ; being used with VTCOM. If CLVRSN is defined as 0 then an appropriate
495          ; value will be selected via an internal table. Zero is the suggested
496          ; setting.
497          ;
498          000000 CLVRSN = 0. ;CL version number
499          ;
500          ; Define maximum number of user defined activation characters
501          ; that each line may define during execution.
502          ;
503          000020 MXSPAC = 16. ;Max # user defined activation chars per job
504          ;
505          ; Define maximum number of characters that can be translated by
506          ; the terminal handler. This translation consists of replacing
507          ; a received character by a substitution character on input and replacing
508          ; the substitution character by the original character on output.

```

```

509      ; This parameter must be non-zero to use the SET TT TRANSLATE=( ) command.
510      ;
511      000005      MXTTCT =      5.      ;Max # chars that terminal handler can translate
512      ;
513      ; Select default system editor.
514      ; The choices are
515      ;   EDIT
516      ;   TECO
517      ;   KED
518      ;   K52
519      ;
520      000003      EDITOR =      KED      ;Default system editor
521      ;
522      ; Select system default implicit or explicit wildcards for CCL commands.
523      ;   If WILDFL = 0 then explicit wildcards are selected.
524      ;   If WILDFL = 1 then implicit wildcards are selected.
525      ;
526      000001      WILDFL =      1      ;1==>Implicit wildcard, 0==>Explicit wildcard
527      ;
528      ;-----
529      ; The DEVDEF macro must be used to define the names and characteristics
530      ; of all devices which are to be available to TSX-Plus users.
531      ; The form of a device definition is:
532      ;
533      ;       DEVDEF <device>[,option,...,option]
534      ;
535      ; For each device to be available to the system an entry must be made
536      ; using the DEVDEF macro. This macro requires at least one argument
537      ; but may have several optional arguments as described below:
538      ;
539      ; 1. The first parameter is the two character device name enclosed
540      ;    in angle brackets.
541      ; 2. The optional parameters specify the device characteristics.
542      ;    There are nine allowable device attributes which may be
543      ;    specified in any order. They are as follows:
544      ;
545      ;       DMA      Device performs Direct Memory Access (DMA).
546      ;       MAPIO   Perform I/O mapping (18-bit controllers on 22-bit QBUS).
547      ;       EVNBUF  Require even byte buffer address for I/O transfers.
548      ;       NOCACHE Do not use generalized data cache for this device.
549      ;       NOMOUNT Do not allow mounts (i.e., use directory cache) for
550      ;               this device.
551      ;       REGALC  Require device allocation before use.
552      ;       MAPH    Load the device handler outside the low memory 40K
553      ;               byte region and into a mapped handler region.
554      ;       NOMAPH  Do not load the handler into a mapped handler region
555      ;               instead load it into the low memory 40k byte region.
556      ;       HANBUF  Handler contains an internal I/O buffer.
557      ;
558      ; For standard device drivers, it is important to choose MAPIO when
559      ; 18-bit controllers or handlers will be used on a 22-bit LSI system.
560      ; It is not necessary to specify other device attributes for standard
561      ; TSX-Plus supplied device drivers since TSX-Plus will automatically
562      ; make default selections.
563      ;
564      ; *****
565      ; ** When performing a TSX-Plus **

```

```

566 ; ** system generation, remove the **
567 ; ** devices in this list which are **
568 ; ** not present on your system, **
569 ; ** and include those which are. **
570 ; *****
571 ;
572 002522 DEVBEG ;Beginning of device definitions
573
574 002522 DEVDEF <DW> ;Pro hard disk
575 002522 DEVDEF <DZ> ;Pro floppy disk
576 002522 DEVDEF <NL> ;Null handler
577 002522 DEVDEF <ZZ>,MAPH ;DBL security handler
578 002522 DEVEND ;End of device definitions
579
580 ; -----
581 ; Parameters related to system I/O buffers used when DMA devices
582 ; with 18-bit controllers are used on Q-bus systems with
583 ; 22-bit addressing (e.g., 11/23-Plus and 11/73).
584 ;
585 ; Number of system buffers allocated for I/O buffering.
586 ; (The recommended number is one per active device that requires buffering.)
587 ;
588 000000 MIONBF = 0. ;Number of system I/O buffers
589 ;
590 ; Size of each system I/O buffer, in units of 512 bytes.
591 ; The maximum allowed value for this parameter is 15.
592 ;
593 000017 MIOBSZ = 15. ;I/O buffer size in units of 512 bytes
594 ;
595 ; -----
596 ; Some device handlers allocate extended memory (PLAS) regions for
597 ; their use. For example, the DU and MU handlers each require one
598 ; PLAS region. If you are using any other handlers which require
599 ; extended memory regions, include the number of regions required.
600 ;
601 000000 DEVXMR = 0. ;Number of XM regions for device handlers
602 ;
603 ; -----
604 ; SPFLAG allows you to specify whether or not you wish individual
605 ; spooled devices to print a flag page before each file. This macro
606 ; defines the initial settings, which may be later changed with the
607 ; SPOOL dev,[NO]FLAGPAGE keyboard command. The initial setting is
608 ; specified by selecting "F" for FLAGPAGE or "N" for NOFLAGPAGE.
609 ; Each "F" or "N" must correspond to the respective device in the
610 ; list of device names in the SPOOL macro. For example:
611 ;
612 ; SPFLAG F,N
613 ;
614 ; would enable flag pages for the first device and disable them for the
615 ; second device specified in the SPOOL macro. As an example, with the
616 ; following SPOOL macro:
617 ;
618 ; SPOOL 2,20.,3,2000.,<LP CL2>,0,5.
619 ;
620 ; flag pages would be initially enabled for LP and disabled for CL2.
621 ;
622 002522 SPFLAG F,N

```

623
 624
 625
 626
 627
 628
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 640
 641
 642
 643 002522
 644
 645
 646
 647
 648
 649
 650
 651
 652
 653
 654
 655
 656
 657
 658
 659
 660
 661
 662
 663
 664
 665 002522
 666
 667
 668
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679

```

-----
; SPWIDE allows you to specify the default width to be used for centering
; flagpages on individual spooled devices. This is done by specifying
; "W" for WIDE (132 column centering), or "N" for NARROW (80 column
; centering). Each "W" or "N" must correspond to the respective device
; in the list of device names in the SPOOL macro. For example in:
;
;       SPWIDE  W,N
;
; where the SPOOL macro looks like:
;
;       SPOOL   2,20.,3,2000.,<LP CL2>,0,5.
;
; flag pages printed on LP will be centered on 132 columns, whereas flag
; pages printed on CL2 will be centered on 80 columns. Note that the
; only choice is between 80 and 132 column centering, no other widths
; are supported. The initial setting may be changed later with the
; SPOOL dev,WIDE and SPOOL dev,NARROW keyboard commands.
;
;       SPWIDE  W,N
;
-----
; SPHOLD allows you to specify whether or not spool files may be started
; printing as soon as they are created or should be held until the output
; channel is closed. This is specified by selecting "H" for HOLD if they
; are to be held until the channel is closed, or "N" for NOHOLD if spool
; files may be started printing as soon as they are created. Each "H" or
; "N" must correspond to the respective device in the list of device
; names in the SPOOL macro. For example in:
;
;       SPHOLD  H,N
;
; where the SPOOL macro looks like:
;
;       SPOOL   2,20.,3,2000.,<LP CL2>.,5.
;
; spooled device LP will hold files from being printed until the channel
; is closed, whereas spooled device CL2 will allow files to start printing
; as soon as they are created. The SPOOL dev,[NO]HOLD keyboard command
; may be used later to change the initial settings.
;
;       SPHOLD  N,H
;
-----
; Define those devices which are to be spooled by TSX-Plus
; (such as line printers).
; There are seven arguments to the SPOOL macro:
; 1. Number of devices to be spooled (may be zero).
; 2. Number of spool files which may be open by all users.
; 3. Number of spool buffers (512. bytes each).
; 4. Number of blocks in spool disk file.
; 5. List of 3 character names of devices to be spooled.
; 6. Specify 0 if spool files are to be eligible to be started
;    as soon as they are created, specify 1 if they are to be held
;    until the channel is closed. This parameter specifies for all
;    devices the same characteristic as selected by the SPHOLD macro.
    
```

```

680 ;           If you wish to specify hold/nohold on an individual device basis,
681 ;           using the SPHOLD macro, then this parameter should be blank (.,)
682 ;           If a value is specified, it will override all the SPHOLD selections.
683 ;           7. Number of blocks which are to be backed up
684 ;           when the "SPOOL xx,BACK" command is given.
685 ;
686 ;           Note: The SPOOL macro must be present even if
687 ;           there are no spooled devices. However, if the first
688 ;           argument (number of spooled devices) is zero, no spool
689 ;           tables are generated and arguments 2-7 are ignored.
690 ;
691 002522 ;           SPOOL  2,30.,3,250.,<CL2CL3>,0,5.
692 ;
693 ;-----
694 ; Define parameters pertaining to record (block) locking
695 ; for shared files.  If the shared file block locking
696 ; facility is not wanted, set all of these parameters to
697 ; 0 (zero).
698 ;
699 ; Maximum number of shared files which may be open
700 ; simultaneously.  Note that several users accessing the same
701 ; file count as 1.
702 ;
703 000062 MAXSF  =      50.      ;Max number of shared files
704 ;
705 ; Maximum number of I/O channels which all users may
706 ; simultaneously have open to shared files.
707 ; Note, this is the total number for all users not
708 ; for each user.
709 ;
710 000144 MAXSFC =      100.     ;Max # shared file channels
711 ;
712 ; Maximum number of blocks which may be simultaneously
713 ; held locked by any channel.  That is, max blocks
714 ; locked per channel.
715 ;
716 000003 MXLBLK =       3.      ;Max blocks locked per channel
717 ;
718 ; Number of 512-byte blocks to be held in the in-memory data
719 ; cache for shared files.
720 ; (Note that the MAXSF, MAXSFC, and MXLBLK parameters must be
721 ; non-zero to enable shared file data caching.)
722 ;
723 000000 NUMDC  =       0.      ;Number of blocks in shared file data cache
724 ;
725 ;-----
726 ; Define parameters pertaining to the inter-program
727 ; message communication feature.  If this feature is
728 ; not wanted, set all four parameters to 0 (zero).
729 ;
730 ; Maximum number of message communication channels
731 ; which may be simultaneously in use.
732 ;
733 000005 MAXMC  =       5.      ;Max message channels
734 ;
735 ; Maximum message length (bytes).
736 ;

```

```

737      000310      MSCHRS =      200.      ;Max message length (bytes)
738      ;
739      ; Maximum number of messages which may be held in queue.
740      ;
741      000006      MAXMSG =      6.      ;Max queued messages
742      ;
743      ; Maximum number of requests for messages that may be held in queue
744      ;
745      000005      MAXMRB =      5.      ;Max # pending message requests
746      ;
747      ;-----
748      ; The RTVECT parameter specifies the number of real-time interrupt vectors
749      ; that can be connected to TSX-Plus jobs. Set RTVECT to the maximum number
750      ; of interrupt vectors that all running real-time programs may be connected
751      ; to at the same time.
752      ; (Note: The basic real-time support facility is now a standard part of
753      ; TSX-Plus and it is no longer necessary to set RTVECT to 1 to include
754      ; real-time facilities such as locking a job in memory or accessing the
755      ; I/O page. It is also no longer necessary to set RTVECT to 1 to allow
756      ; use of the SYSMON program. RTVECT should be set to 0 (zero) unless some
757      ; real-time interrupts are going to be connected to TSX-Plus jobs.)
758      ;
759      000000      RTVECT =      0.      ;Max # interrupt vectors that may be connected
760      ;
761      ;-----
762      ; Define the size of the table within TSX-Plus used to hold information
763      ; when the performance monitoring feature is being used.
764      ; Each word in this table corresponds to one cell in the histogram.
765      ; Specify the size as number of bytes for the table.
766      ; (Note: The maximum allowed size is 8192 bytes)
767      ;
768      000000      PMSIZE =      0.      ;Size of performance monitor table (bytes)
769      ;
770      ;-----
771      ; Use the RTDEF macro at this point to specify information about
772      ; any shared run-time systems to be loaded when TSX-Plus is started.
773      ;
774      ; The form of the RTDEF macro is
775      ; RTDEF <name>,r-flag,skip-count
776      ;
777      ; Where
778      ; - Name is the 12 character name of the file containing the run-time system
779      ; which must be specified in the form DevFilnamExt -- that is, three
780      ; character device name, six character file name and three character
781      ; extension.
782      ; - R-flag is either R if user programs are to have read-only access to
783      ; the run-time system, or RW if read-write access is to be granted.
784      ; - Skip-count is the number of blocks to be skipped over at the front
785      ; of the file when loading it.
786      ;
787      ; Example:
788      ; RTDEF <SY CBR064SHR>,R,1.      ;COBOL-Plus shared run-time
789      ; RTDEF <SY DBLSHRRTS>,R,1.      ;DBL shared run-time
790      ; RTDEF <SY DB4RTSSHR>,R,0.      ;DBL V4 shared run-time
791      ;

```

```

1          ; -----
2          ; Time-sharing line parameters:
3          ;
4          ; Default input and output character buffer sizes.
5          ; These buffer sizes will be used for lines that don't use
6          ; the BUFSIZ macro within their line definitions to declare
7          ; their character buffer sizes.
8          ; These buffer sizes are also used for all subprocesses.
9          ;
10         000276 DINSPC =      190.    ;Default input char buffer size
11         000360 DOTSPC =      240.    ;Default output char buffer size
12         ;
13         ; When the terminal-output character buffer is filled a job is suspended.
14         ; The job is restarted after characters are printed from the buffer and
15         ; there are OTRASZ characters remaining in the buffer.
16         ;
17         000031 OTRASZ =      25.    ;Reactivation character count
18         ;
19         ; A software character "silo" is used to hold characters received
20         ; from time-sharing lines until they can be processed by the system.
21         ; The silo is used to prevent the loss of characters during high
22         ; speed input. Each time-sharing line and CL line has its own silo.
23         ; If the input to the line is coming from a terminal, the silo can be
24         ; quite small. On the other hand, if the input is coming from another
25         ; computer or other high speed device, the silo size should be increased.
26         ; The NCSILO, NCXOFF, and NCXON parameters set default values pertaining
27         ; to the silos. The SILO macro can be used within a line definition
28         ; to specify silo parameters for a specific line.
29         ;
30         ; Default size of input character silos.
31         ;
32         000062 NCSILO =      50.    ;Default silo size
33         ;
34         ; The system will transmit a control-S (XOFF) character when an input
35         ; silo is filled to the point where there are only NCXOFF free
36         ; character positions remaining.
37         ;
38         000014 NCXOFF =      12.    ;Default XOFF point for silos
39         ;
40         ; If the system sends an XOFF because a silo becomes nearly full,
41         ; it will send an XON to restart transmission when there are only
42         ; NCXON characters remaining in the silo.
43         ;
44         000004 NCXON  =      4.    ;Default XON point for silos
45         ;
46         ; Number of "extra" CL (communication line) units to be genned into
47         ; system. These CL units are not initially assigned to any line but
48         ; may be used "take over" a time-sharing line to use it as a CL unit.
49         ; The total number of CL units (those defined using CLDEF blocks plus
50         ; the extra units) may not exceed 16. The first 8 CL units are
51         ; named CLO to CL7, the second 8 are named C10 through C17.
52         ;
53         000006 CLXTRA =      6.    ;Number of extra CL units.
54         ;
55         ; Default output ring buffer size for I/O communication lines defined
56         ; with the CLDEF macro and accessed as "CL" devices.
57         ; The recommended value is ((3*baud_rate)/1000+2).

```

```

58      ;
59      000040      CLORSZ =      32.      ; Size of CL output ring buffers
60      ;
61      ; -----
62      ; Flags which can be used with the FLAGS macro within
63      ; a line definition block to define line characteristics.
64      ;
65      100000      $SCOPE =      100000 ; ON==>CRT type terminal
66      040000      $ECHO  =      40000  ; ON==>Echo characters to terminal
67      020000      $TAPE  =      20000  ; ON==>"Paper-tape" mode (do x-on/x-off control, etc.)
68      010000      $BBIT  =      10000  ; ON==>Support 8 bit (rather than 7 bit) characters.
69      004000      $START =      4000   ; ON==>Automatically start line during initialization
70      002000      $ALTER =      2000   ; ON==>Allow .GTLIN to receive ! and activate on ^C
71      001000      $TAB   =      1000   ; ON==>Do not simulate tabs (Terminal handles tab char)
72      000400      $FORM  =      400    ; ON==>Do not simulate form-feeds (Terminal handles FF)
73      000200      $AUTO  =      200    ; ON==>Do autobaud speed selection for line
74      000100      $PAGE  =      100    ; ON==>Enable ctrl-S/ctrl-Q input processing
75      000040      $LC    =      40     ; ON==>Enable lower-case input
76      000020      $NOSUB =      20     ; ON==>Disallow use of subprocesses
77      000010      $DEFER =      10     ; ON==>Do deferred character echoing (recommended)
78      000004      $QTSET =      4      ; ON==>Set tt quiet (Don't list command files)
79      000002      $SYSPP =      2      ; ON==>Require system password before logon
80      000001      $PHONE =      1      ; ON==>Dial-up, modem connected line
81      ;
82      ; Default line flags that will be used for each line that does
83      ; not explicitly specify flags using a FLAGS macro.
84      ;
85      040150      NRMFLG =      $ECHO!$DEFER!$PAGE!$LC
86      ;
87      ; -----
88      ; Terminal type names that are legal to used with the TRMTYP macro
89      ; within a line definition block to define the terminal type.
90      ;
91      ; VT100 ==> DEC VT100
92      ; VT200 ==> DEC VT200 with 7 bit control codes
93      ; VT52  ==> DEC VT52
94      ; LA36  ==> DEC LA36
95      ; LA120 ==> DEC LA120
96      ; HAZEL ==> Hazeltine brand terminals
97      ; ADM3A ==> Lear Siegler ADM3A
98      ; DIABLO==> Diablo brand terminals (with X-ON/X-OFF protocol)
99      ; QUME  ==> Qume brand terminals (with X-ON/X-OFF protocol)
100     ;

```

```

1
2 ; -----
3 ; Line definitions
4 ;
5 ; The TBLDEF macro call requires four arguments:
6 ; 1. The number of real (physical) time-sharing lines on machine.
7 ; 2. The number of subprocess jobs.
8 ; 3. The number of detached jobs.
9 ; 4. The number of dedicated CL lines.
10 002736 ; TBLDEF 7.,3.,2.,0. ;# Real, # Subprocess, # Detached, # CL lines
11
12 ;
13 ; Define primary (real) time-sharing lines
14 ;
15 010066 ; LINDEF 60,177560,OPER ;Use console terminal as t/s term
16 010066 ; NAME <Console>
17 010066 ; CMDFIL LINE1.TSX
18 010066 ; TRMTYP VT100
19 010066 ; FLAGS NRMFLG!$START!$SCOPE!$TAB
20 010066 ; LINEND
21
22 010106 ; LINDEF 220,173400 ;Printer port
23 010106 ; NAME <Printer port>
24 010106 ; FLAGS NRMFLG!$AUTO
25 010106 ; CMDFIL LINE2.TSX
26 010106 ; TRMTYP VT100
27 010106 ; LINEND
28
29 010126 ; LINDEF 210,173300 ;Communications port
30 010126 ; NAME <Com. port>
31 010126 ; FLAGS NRMFLG!$AUTO!$PHONE
32 010126 ; CMDFIL LINE3.TSX
33 010126 ; TRMTYP VT100
34 010126 ; LINEND
35 ;
36 ; Define lines on Pro quad serial line unit
37 ;
38 010146 ; MUXDEF 300,160000 ;Quad serial line unit
39 ;
40 ; Line 0 on quad serial line unit
41 ;
42 010146 ; LINDEF 0
43 010146 ; NAME <QSL 0>
44 010146 ; TRMTYP VT100
45 010146 ; CMDFIL LINE3.TSX
46 010146 ; FLAGS NRMFLG!$AUTO!$PHONE!$SCOPE!$TAB
47 010146 ; LINEND
48 ;
49 ; Line 1 on quad serial line unit
50 ;
51 010166 ; LINDEF 1
52 010166 ; NAME <QSL 1>
53 010166 ; TRMTYP VT100
54 010166 ; CMDFIL LINE3.TSX
55 010166 ; FLAGS NRMFLG!$AUTO!$PHONE!$SCOPE!$TAB
56 010166 ; LINEND
57 ;

```

```
58 ; Line 2 on quad serial line unit
59 ;
60 010206 LINDEF 2
61 010206 NAME <QSL 2>
62 010206 TRMTYP VT100
63 010206 CMDFIL LINE3.TSX
64 010206 FLAGS NRMFLG!$AUTO!$PHONE!$SCOPE!$TAB
65 010206 LINEND
66 ;
67 ; Line 3 on quad serial line unit
68 ;
69 010226 LINDEF 3
70 010226 NAME <QSL 3>
71 010226 TRMTYP VT100
72 010226 CMDFIL LINE3.TSX
73 010226 FLAGS NRMFLG!$AUTO!$PHONE!$SCOPE!$TAB
74 010226 LINEND
75 ;
76 010246 MUXEND
77 ;
78 ;
79 ; Use the "DETACH" macro here to declare any start-up command
80 ; files to be run as detached jobs.
81 ;
82 010246 DETACH SY:DET1.TSX ;Start-up detach job command file
83 010370 DETACH SY:DET2.TSX ;Start-up detach job command file
84 ;
85 ;=====
86 ; END OF SECTION OF TSGEN TO BE ALTERED BY USER
87 ;=====
```

3
4
5

.LIST MD
.LIST CND
.ENDC

```

1          ; -----
2          ;   Finish building tables
3          ;
4          ;   Make sure memory size parameters are reasonable.
5          ;
6          .IF      GT,HIMEM-64.
7          .ERROR   ;HIMEM may not exceed 64.
8          HIMEM    =      64.
9          .ENDC
10         .IF      GT,DFLMEM-HIMEM
11         DFLMEM   =      HIMEM
12         .ENDC
13         ;
14         ;   Make sure silo parameters are reasonable
15         ;   Actual silo limit is specified by MAXSLO
16         ;
17         .IIF     GT,<NCSILO-255.> NCSILO = 255.
18         S = <NCSILO/2>-2
19         .IIF     GT,<NCXOFF-S>      NCXOFF = S
20         NCXOFF   = NCXOFF&377      ;MUST BE BYTE SIZE
21         .IIF     GT,<NCXON-S>      NCXON  = S
22         NCXON    = NCXON&377      ;MUST BE BYTE SIZE
23         ;
24         ;   Make sure last line definitions were properly terminated
25         ;
26         .IF      NE BO
27         .ERROR   1 ; Missing LINEND for last line
28         LINEND
29         .ENDC
30         .IF      NE CURMX
31         .ERROR   1 ; Missing MUXEND
32         MUXEND
33         .ENDC
34         ;
35         ;   Make sure the right # of lines were defined.
36         ;
37         .IF      NE <NPLDF-NPL>
38         .ERROR   2; Wrong number of primary lines defined
39         .ENDC
40         .IF      NE <NCLDF-NIOL>
41         .ERROR   2; Wrong number of CL lines defined
42         .ENDC
43         .IF      GT <NDLDF-NDL>
44         .ERROR   2; Wrong number of detached jobs defined
45         .ENDC
46         ;
47         ;   Define any additional detached job lines.
48         ;
49         .REPT    <NPL+NDL-LN>
50         DETACH
51         .ENDR
52         .IF      NE <LN-NPL-NDL>
53         .ERROR   2 ; Wrong number of lines defined
54         .ENDC
55         .IF      NE <NCLDF-NIOL>
56         .ERROR   2; Wrong number of CL lines defined
57         .ENDC

```

000027

000014

000004

000000

```

58      ;
59      ; Define tables for subprocesses (if any)
60      ;
61      .IF      NE NSL
62      .REPT    000003
63      BO      =      1          ;BEGIN BLOCK
64      LN      =      LN+1
65      LX      =      LX+2
66      CLX     =      LX
67      BUFSIZ  DINSPC, DOTSPC
68      LINEND
69      .ENDR
70      .ENDC
71 010512     .CSECT  TSGEN
72      ;
73      ; Define mux interrupt entry vectors
74      ;
75      ; Input interrupt entry points
76      .REPT    000001
77      INCB    MUXNUM
78      .ENDR
79 010516     000167  000000G  INMXV:  JMP      ININT
80      ; Output interrupt entry points (set up by TSINIT)
81 010522     OTMXV:  .BLKW  3*<LSTMX/2>
82      ;
83      ; Generate tables for Unibus map registers
84      ;
85      .GLOBL  UMRBAS, UMREND, UMRWHD
86 010530     000000  UMRWHD: .WORD  0
87      .MACRO  UMRDEF  NUM
88      .BYTE  CURUMR          ;UM$UMR
89      .BYTE  NUM             ;UM$NMR
90      .WORD  NUM*4096.       ;UM$WDS
91      .WORD  0               ;UM$IOQ
92      CURUMR =      CURUMR+NUM
93      .ENDM  UMRDEF
94      ; Define UMR sets in order of size -- small to large.
95      CURUMR =      5          ;Map regs 0-4 always mapped by init code.
96 010532     UMRBAS:
97      .IF      EQ, <BUSTYP-UNIBUS> ;Generate only for UNIBUS machines
98      UMRDEF  1.
99      UMRDEF  1.
100     UMRDEF  4.
101     UMRDEF  4.
102     UMRDEF  8.
103     UMRDEF  8.
104     .ENDC
105 010532     UMREND:
106      ;
107      ; Check file and device caching parameters
108      ;
109      .IF      LT, <MAXCSH-1>      ;MAKE SURE WE CACHE AT LEAST 1 DEVICE
110     MAXCSH  =      1
111     .ENDC
112     .IF      LT, <NMFCSH-4>      ;MINIMUM NUMBER OF CASHED FILES
113     NMFCSH  =      4
114     .ENDC

```

```

115 ;
116 ; Generate tables to keep track of allocated devices
117 ;
118 .IF LT <MAXALC-1> ;Make sure we have at least 1 entry
119 MAXALC = 1
120 .ENDC
121 010532 ALCTBL: ;Base of device allocation table
122 000005 .REPT MAXALC
123 .WORD 0 ;AD$DVU
124 .BYTE 0 ;AD$JOB
125 .BYTE 0 ;AD$FLG
126 .ENDR
127 010556 ALCEND: ;End of device allocaton table
128 ;
129 ; Check validity of PMSIZE
130 ;
131 .IF GT <PMSIZE-8192.> ;PMSIZE MAY NOT EXCEED 8192.
132 .ERROR 0;PMSIZE may not exceed 8192.
133 PMSIZE = 8192.
134 .ENDC
135 ;
136 ; Check validity of MIOBSZ
137 ;
138 .IF GT <MIOBSZ-15.>
139 .ERROR 0;MIOBSZ may not exceed 15.
140 MIOBSZ = 15.
141 .ENDC
142 ;
143 ; Check validity of CACHE parameter
144 ;
145 .IF GT <CACHE-4095.>
146 CACHE = 4095.
147 .ENDC

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

000002

```
-----  
; Generate tables for shared file i/o control  
;  
; .IF NE,MAXSF ;ANY SHARED FILES?  
; **  
; ** Assemble if there are shared files **  
; **  
; .IFF  
; **  
; ** Assemble this code if there are no shared files **  
; **  
MAXSFC = 0  
MXLBLK = 0  
NUMDC = 0  
;  
; .ENDC  
;  
;  
-----  
; Generate dummy shared run-time definitions  
;  
; .REPT NDRTDF  
RTDEF <$$ >,R,0  
; .ENDR
```

```

1          ; -----
2          ;   Generate tables for real-time support facility.
3          ;
4          ;   Generate the vector control blocks
5          ;
6          . GLOBL  VCBBAS, VCBEND
7 010556   VCBBAS:
8          . REPT  RTVECT
9          JSR    R2, @#RTINT
10         . WORD  0, 0, 0          ; Must match size defined in TSDEFS
11         . ENDR
12 010556   VCBEND:
13         NUMIOQ =    NUMIOQ+RTVECT ; ADD I/O QUEUE ELEMENTS FOR INT COMPL ROUTINES
14         NUMSYQ =    NUMSYQ+RTVECT
15         ;
16         . IF    NE, RTVECT
17         ; **
18         ; ** Assemble this code if real-time support is wanted
19         ; **
20         . GLOBL  RTINT
21         . ENDC          ; End of real-time conditional
22         ;
23         ; -----
24         ;   Conditional code for different types of terminals.
25         ;
26         . IF    NE, DHUSE
27         ; **
28         ; ** Assemble this code if DH11 support is needed
29         ; **
30         . IFF
31         ; **
32         ; ** Assemble this code if DH11 support is not needed
33         ; **
34         . GLOBL  TSDHIO, DHSTRT, DHSTOP, DHOINT
35         . GLOBL  VHSTRT, VHOINT, DHTIMR, VHSTOP
36         . GLOBL  DHXON, DHXOFF, VHXOFF, VHXON
37 010556   TSDHIO =    0
38 010556   DHTIMR:
39 010556   DHSTRT:
40 010556   DHSTOP:
41 010556   DHXON:
42 010556   DHXOFF:
43 010556   DHOINT:
44 010556   VHSTRT:
45 010556   VHSTOP:
46 010556   VHXON:
47 010556   VHXOFF:
48 010556   VHOINT:
49 010556   000207   RETURN
50         . ENDC          ; End of DH11 conditional code
51         ;
52         ; -----
53         ;   Conditional code for CL units
54         ;
55         . GLOBL  CL$OPT, CL$STA, CL$COL, CL$RQH, CL$WQH
56         . GLOBL  CL$ORB, CL$ORA, CL$ORS, CL$ORG, CL$ORP, CL$ORE
57         . GLOBL  CL$LEN, CL$LIN, CL$WID, CL$SKP, CL$LIX

```

```

58          .GLOBL  CL$EPN,CL$EPS,CL$EPP,CL$XLN,CLSTS
59          ;
60          ; Define CL device status word. See the RT-11 .DSTATUS or .DRDEF macros
61          ; for status bit definitions. The device type code is the same as XL.
62          ; (To make file names appear in SHOW QUEUE for spooled CL device,
63          ; add the SPECL$ flag (10000).)
64          ;
65          006057 CLSTS = 4000+2000+57          ;HNDLR$+SPFUN$+XL$COD
66          ;
67          .IF      NE,CLVRSN          ; Test to see if CLVRSN is defined as 0
68          .IF      LE,<CLVRSN-14.> ; or greater than 14.
69          .ERROR 0;Minimum CL version number is 15. ;If not report an error
70          CLVRSN = 15.          ; and set to a reasonable number
71          .ENDC          ; End conditional LE,<CLVRSN-14.>
72          .ENDC          ; End conditional NE,CLVRSN
73          ;
74          ; Define table that tells which line is associated with each CL unit.
75          ; Note, this table is always generated. Even if there are no CL lines
76          ; genned in.
77          ;
78          010560 000000 000000 000000 CL$LIX: .WORD 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ;Line index for each CL unit
79          010566 000000 000000 000000
80          010574 000000 000000 000000
81          010602 000000 000000 000000
82          010610 000000 000000 000000
83          010616 000000 000000 000000
84          ;
85          ; Macro to define CL tables which have one entry per CL unit
86          ;
87          .MACRO  CLTABL NAME
88          NAME:
89          .REPT  CLTOTL
90          .WORD  0
91          .ENDR
92          .ENDM  CLTABL
93          ;
94          .IF      NE,CLTOTL          ; Assemble if there are CL units
95          ;**
96          ;** Assemble this code if there are CL units
97          ;**
98          CLTABL CL$OPT          ;Option flags (CO$xxx)
99          CLTABL CL$STA          ;Status flags (CM$xxx)
100         CLTABL CL$COL          ;Current column position
101         CLTABL CL$LEN          ;Number of lines per page
102         CLTABL CL$LIN          ;Current line number
103         CLTABL CL$XLN          ;Number of line CL unit is cross connected to
104         CLTABL CL$SKP          ;Number of lines to skip at bottom of page
105         CLTABL CL$WID          ;Maximum allowed line width
106         CLTABL CL$RQH          ;Internal queue head for read requests
107         CLTABL CL$WQH          ;Internal queue head for write requests
108         CLTABL CL$ORB          ;Start of output ring buffer
109         CLTABL CL$ORE          ;End of output ring buffer
110         CLTABL CL$ORP          ;Pointer where next char goes in output ring
111         CLTABL CL$ORG          ;Pointer to next char to get from output ring
112         CLTABL CL$ORA          ;Allocated size of output ring buffer
113         CLTABL CL$ORS          ;Free space in output ring buffer
114         CLTABL CL$EPN          ;Number of Form-feeds for end page

```

```

110 011134          CLTABL CL$EPS          ;Pointer to end-of-file string buffer
111 011150          CLTABL CL$EPP          ;Pointer to next char within EOF string buffer
112                ;
113                . IFF                    ;Assemble if no CL units
114                ;**
115                ;** Assemble this code if there are no CL units
116                ;**
117                . GLOBL CLOTIR, CLSIZE, CLHEAD, CLLQE, CLCQE, CLINIR, CLABF
118                . GLOBL TSCLR
119                TSCLR = 0
120                CLSIZE = 0
121                CLHEAD = 0
122                ;
123                CLCQE:
124                CLLQE:
125                CL$OPT:
126                CL$STA:
127                CL$COL:
128                CL$LEN:
129                CL$LIN:
130                CL$XLN:
131                CL$SKP:
132                CL$WID:
133                CL$RQH:
134                CL$WQH:
135                CL$ORB:
136                CL$ORE:
137                CL$ORP:
138                CL$ORG:
139                CL$ORA:
140                CL$ORS:
141                CL$EPN:
142                CL$EPS:
143                CL$EPP:
144                . WORD 0
145                ;
146                CLINIR:
147                CLOTIR:
148                CLABF:
149                RETURN
150                ;
151                . ENDC                    ;End conditional (NE, CLTOTL)
152                ;
153                ;-----
154                ; Define some misc data cells
155                ;
156 011164 000012   NMFREQ: . WORD NUMIQ-NUMSYQ ;# I/O queue elements available for user jobs
157                ;
158                ; Invoke dummy SYSPS if user commented it out
159                ;
160                . IIF EQ, SYSPSS SYSPS <>
161                ;
162                ; Make sure PHONE parameter is 0 or 1
163                ;
164                . IIF NE, PHONE PHONE = 1
165                ;
166                ; Close out some CSECTS.

```

```

167
168 000034
169 000034
170 000240
171
172 000000
173 000000 123456
174
176
177
178
179
180
181 000000
182 000050
183 000050 000000G
184 000014
185 000014 000000G
186 000016 000000G
187
188 000001

```

```

;
; .CSECT RDBSEC
RDBEND:
; .CSECT SASECT
; EVEN
; .CSECT GENTOP
GENTOP: .WORD 123456 ;Flag word for top of tsgen
; IF NE FULLST
; LIST BIN
; ENDC ;NE FULLST
;
; Define address of top of resident portion of TSX-Plus
;
; .ASECT
; = 50
; .WORD PROITP
; = 14
; .WORD SCHED
; .WORD DSKBUF
;
; .END

```

Errors detected: 0

*** Assembler statistics

Work file reads: 173
 Work file writes: 159
 Size of work file: 23718 Words (93 Pages)
 Size of core pool: 18176 Words (71 Pages)
 Operating system: RT-11

Elapsed time: 00:01:44.23
 ,LP: PROGEN=DK: PROGEN/C/N: SYM

IOLFLG	7-52#	17-15	17-15	17-17	17-18	17-20	17-20	17-20	17-20#	17-22	17-22	17-25
	17-26	17-27	17-27	17-27	17-27#	17-29	17-29	17-32	17-33	17-34	17-34	17-34
	17-34#	17-42	17-42	17-44	17-45	17-47	17-47	17-47	17-47#	17-51	17-51	17-53
	17-54	17-56	17-56	17-56	17-56#	17-60	17-60	17-62	17-63	17-65	17-65	17-65
	17-65#	17-69	17-69	17-71	17-72	17-74	17-74	17-74	17-74#	19-69	19-69	19-69
	19-69	19-69	19-69	19-69#	19-69#	19-69#						
IOLN	7-53#											
IDQSIZ	1-139	2-22										
IDSTRT	1-138	2-14										
ITRMTP	17-10	17-10#	17-18	17-26	17-33	17-44	17-53	17-62	17-71			
JCXPGS	1-111	3-342#										
K52	3-381#											
KED	3-380#	15-520										
KEYMAX	3-38	15-232#										
KMNCHN	1-127	3-356#										
KMNHI	1-53	3-64#										
KMNPGS	1-127	3-355#										
KMNSTK	1-127	3-353#										
KMNSTR	1-127	3-354#										
KMNTOP	1-53	3-63#										
KMONCE	1-46	3-215#										
KUSECK	1-119	1-169#										
L	15-578	15-578	15-578#									
LA120	1-136											
LA36	1-136											
LABTIM	17-10	17-10#										
LACTIV	17-10	17-10#										
LAFSIZ	17-10	17-10#										
LAST	3-387#											
LBASE	17-10	17-10#										
LBRKCH	17-10	17-10#										
LBRKCQ	17-10	17-10#										
LBSPRI	17-10	17-10	17-10#									
LCDTIM	17-10	17-10#										
LCDTYP	17-10	17-10#	17-15	17-22	17-29	17-42	17-51	17-60	17-69			
LCLUNT	17-10	17-10#										
LCMPL	17-10	17-10#										
LCMQHD	17-10	17-10#										
LCOL	17-10	17-10#										
LCONTM	17-10	17-10#										
LCPUHI	17-10	17-10#										
LCPULO	17-10	17-10#										
LCXPAR	17-10	17-10#										
LCXTBL	17-10	17-10#	17-20	17-27	17-34	17-47	17-56	17-65	17-74			
LDDEVX	1-105	3-102#										
LDHB1B	17-10	17-10#										
LDHB1P	17-10	17-10#										
LDHB2B	17-10	17-10#										
LDHB2R	17-10	17-10#										
LDHB2S	17-10	17-10#										
LDSYS	3-183	15-214#										
LDVERS	1-61	3-219#										
LEMTPC	17-10	17-10#										
LESCHR	17-10	17-10#										
LESRTN	17-10	17-10#										
LF	17-20	17-20#	17-20#	17-27	17-27#	17-27#	17-34	17-34#	17-34#	17-47	17-47#	17-47#

	17-56	17-56#	17-56#	17-65	17-65#	17-65#	17-74	17-74#	17-74#	19-69	19-69	19-69
	19-69#	19-69#	19-69#									
LFWLIM	17-10	17-10#										
LHIPCT	17-10	17-10#										
LHIRBA	17-10	17-10#	17-20	17-27	17-34	17-47	17-56	17-65	17-74			
LHIRBB	17-10	17-10#										
LHIRBC	17-10	17-10#	17-20	17-27	17-34	17-47	17-56	17-65	17-74			
LHIRBE	17-10	17-10#										
LHIRBG	17-10	17-10#										
LHIRBP	17-10	17-10#										
LHIRBS	17-10	17-10#										
LINBUF	17-10	17-10#										
LINCNT	17-10	17-10#										
LINCUR	17-10	17-10#										
LINEND	17-10	17-10#										
LINIR	17-10	17-10#										
LINNUM	1-93	1-133	3-202#	17-10*	17-10*	17-10*	17-10*	17-10*	17-10*	17-10*		
LINNXT	17-10	17-10#										
LINPNT	17-10	17-10#										
LINSIZ	17-10	17-10#	17-20	17-27	17-34	17-47	17-56	17-65	17-74	19-69	19-69	19-69
LINSPC	17-10	17-10#										
LIDCNT	17-10	17-10#										
LIDHLD	17-10	17-10#										
LITIME	17-10	17-10#										
LJSW	17-10	17-10#										
LMEMIN	17-10	17-10#										
LMIHQ	17-10	17-10#										
LMONHD	17-10	17-10#										
LMSGBF	17-10	17-10#										
LMXLN	1-59	17-10#	17-42	17-51	17-60	17-69						
LMXNUM	17-10	17-10#	17-42	17-51	17-60	17-69						
LMXPRM	1-59	17-10	17-10#	17-20	17-27	17-34	17-47	17-56	17-65	17-74		
LN	7-49#	17-15	17-15	17-15#	17-20	17-22	17-22	17-22#	17-27	17-29	17-29	17-29#
	17-34	17-42	17-42	17-42#	17-47	17-51	17-51	17-51#	17-56	17-60	17-60	17-60#
	17-65	17-69	17-69	17-69#	17-74	17-82	17-82	17-82#	17-83	17-83	17-83#	17-83#
	19-52	19-69	19-69	19-69	19-69	19-69	19-69	19-69#	19-69#	19-69#		
LNAME	17-10	17-10#	17-16	17-23	17-30	17-43	17-52	17-61	17-70			
LNBLKS	17-10	17-10#										
LNMAP	1-60	17-10#										
LNMTOP	1-48	17-16#	17-23#	17-30#	17-43#	17-52#	17-61#	17-70#				
LNPRIM	1-60	17-10#										
LNSBLK	17-10	17-10#										
LNSPAC	17-10	17-10#										
LOFFTM	17-10	17-10#										
LOGCHN	1-68	1-224#										
LOKBAS	1-82	3-72#										
LOKCSH	1-82	3-76#										
LOKINI	1-124	3-156#										
LOKMEM	1-82	3-75#										
LOKVEC	1-124	1-126	3-155#									
LOMAP	1-110	3-338#										
LOTBUF	17-10	17-10#										
LOTEND	17-10	17-10#										
LOTNXT	17-10	17-10#										
LOTPNT	17-10	17-10#										
LOTSIZ	17-10	17-10#	17-20	17-27	17-34	17-47	17-56	17-65	17-74	19-69	19-69	19-69

PRILOW	3-186	15-335#										
PRIVIR	3-189	15-347#										
PROBRK	1-118	1-164#										
PROFLG	1-117	3-207#										
PROITP	1-133	21-183										
PROODC	1-118	1-165#										
PROSLT	1-100	3-117#										
PSW	1-135	2-133										
PVON	1-93	3-205#										
PVSPBL	1-59	15-691	15-691	15-691	15-691#							
PWCH	3-192	15-471#										
QBUS	1-114	3-312#	15-90									
QCOMP	1-98	2-69#										
QCOMPL	1-138	2-15										
QFREE	1-138	2-12										
QIO	1-138	2-13										
QUANO	3-4	15-260#										
QUAN1	1-58	3-5	15-269#									
QUAN1A	1-68	3-6	15-275#									
QUAN1B	1-68	3-7	15-279#									
QUAN1C	3-8	15-284#										
QUAN2	1-58	3-9	15-292#									
QUAN3	3-10	15-298#										
QUME	1-137											
R\$CFST	1-58	2-217#										
R\$CH17	1-112	2-210#										
R\$CHN	1-112	2-209#										
R\$DATE	1-112	2-212#										
R\$INST	1-58	2-216#										
R\$INTC	1-104	2-218#										
R\$JOB	1-112	2-213#										
R\$MFMV	1-92	2-220#										
R\$SWPC	1-119	2-219#										
R\$TTOP	1-58	2-215#										
R\$UBAS	1-112	2-214#										
R\$XCHN	1-113	2-211#										
RBR	1-56	17-10	17-10	17-10#	17-15	17-22	17-29					
RDB	1-84	3-372#										
RDBEND	1-84	21-169#										
RDBSEC	1-48	3-371#										
REQALC	4-24#											
RF\$WRT	1-109											
RMNBAS	1-134	2-110	2-111									
RPRCSR	1-79	2-114#										
RPRVEC	1-79	2-115#										
RSFBLK	1-90	15-17#										
RSR	1-56	17-10	17-10#	17-15	17-22	17-29						
RTVECT	1-94	15-759#	21-8	21-13	21-14	21-16						
RUNCHN	1-76	1-226#										
S	17-15	17-15	17-15	17-15#	17-15#	17-15#	17-16	17-16#	17-17	17-17#	17-18	17-18#
	17-20	17-20	17-20	17-20	17-20	17-20	17-20	17-20	17-20	17-20#	17-20#	17-20#
	17-20#	17-20#	17-20#	17-20#	17-20#	17-20#	17-22	17-22	17-22	17-22#	17-22#	17-22#
	17-23	17-23#	17-25	17-25#	17-26	17-26#	17-27	17-27	17-27	17-27	17-27	17-27
	17-27	17-27	17-27	17-27	17-27#	17-27#	17-27#	17-27#	17-27#	17-27#	17-27#	17-27#
	17-29	17-29	17-29	17-29#	17-29#	17-29#	17-30	17-30#	17-32	17-32#	17-33	17-33#
	17-34	17-34	17-34	17-34	17-34	17-34	17-34	17-34	17-34	17-34#	17-34#	17-34#

	17-34#	17-34#	17-34#	17-34#	17-34#	17-34#	17-38	17-38#	17-42	17-42	17-42#	17-42#
	17-43	17-43#	17-44	17-44#	17-45	17-45#	17-47	17-47	17-47	17-47	17-47	17-47
	17-47	17-47	17-47	17-47	17-47#	17-47#	17-47#	17-47#	17-47#	17-47#	17-47#	17-47#
	17-51	17-51	17-51#	17-51#	17-52	17-52#	17-53	17-53#	17-54	17-54#	17-56	17-56
	17-56	17-56	17-56	17-56	17-56	17-56	17-56	17-56	17-56#	17-56#	17-56#	17-56#
	17-56#	17-56#	17-56#	17-56#	17-60	17-60	17-60#	17-60#	17-61	17-61#	17-62	17-62#
	17-63	17-63#	17-65	17-65	17-65	17-65	17-65	17-65	17-65	17-65	17-65	17-65
	17-65#	17-65#	17-65#	17-65#	17-65#	17-65#	17-65#	17-65#	17-69	17-69	17-69#	17-69#
	17-70	17-70#	17-71	17-71#	17-72	17-72#	17-74	17-74	17-74	17-74	17-74	17-74
	17-74	17-74	17-74	17-74	17-74#	17-74#	17-74#	17-74#	17-74#	17-74#	17-74#	17-74#
	17-82	17-82	17-82	17-82#	17-83	17-83	17-83	17-83#	19-18#	19-19	19-21	19-69
	19-69	19-69	19-69	19-69	19-69	19-69	19-69	19-69	19-69#	19-69#	19-69#	19-69#
	19-69#	19-69#	19-69#	19-69#	19-69#	19-69#						
S110	3-393#											
S1200	3-398#											
S134.5	3-394#											
S150	3-395#											
S1800	3-399#											
S19200	3-406#											
S2000	3-400#											
S2400	3-401#											
S300	3-396#											
S3600	3-402#											
S4800	3-403#											
S50	3-391#											
S600	3-397#											
S7200	3-404#											
S75	3-392#											
S9600	3-405#	8-137										
SASECT	1-48	9-6#										
SCHED	1-139	3-44	21-185									
SCPFHD	1-51	3-101#										
SD#HLD	1-134											
SDBULS	1-91	15-691#										
SDCB	1-54	15-691#										
SDCBND	1-54	15-691#										
SDCBSZ	1-91	15-691	15-691#									
SEGBLK	3-34	15-75#										
SEGCHN	1-90	3-62#										
SETERR	1-134											
SFCB	1-68	3-87#										
SFCBFH	1-53	3-89#										
SFCBND	1-68	3-88#										
SFCLS	1-125	3-166#										
SFRSST	1-124	3-160#										
SFSVST	1-124	3-159#										
SFWRIT	1-125	3-167#										
SHRRCB	1-51	3-93#										
SHRRCN	1-51	3-94#										
SILSIZ	17-15#	17-20	17-22#	17-27	17-29#	17-34	17-42#	17-47	17-51#	17-56	17-60#	17-65
	17-69#	17-74										
SILXOF	17-15#	17-20	17-22#	17-27	17-29#	17-34	17-42#	17-47	17-51#	17-56	17-60#	17-65
	17-69#	17-74										
SILXON	17-15#	17-20	17-22#	17-27	17-29#	17-34	17-42#	17-47	17-51#	17-56	17-60#	17-65
	17-69#	17-74										
SIZMEM	3-336	3-336	3-336	3-336	3-336#	3-336#						

SLEDIT	3-184	15-223#						
SMONHD	1-48	3-86#						
SMRSIZ	1-116	3-346#						
SNBUX	1-89	15-691	15-691	15-691	15-691	15-691#		
SNDBX	1-86	3-91	15-691#					
SNMSHD	1-108	3-79#						
SPLANM	1-91	15-691#						
SPLBHD	1-53	15-691#						
SPLBLK	1-55	15-13#						
SPLCHN	1-101	15-691#						
SPLDEV	1-54	15-691	15-691#					
SPLDVN	1-54	15-691	15-691#					
SPLNB	1-52	15-691	15-691	15-691#				
SPLND	1-52	15-691	15-691	15-691#				
SPLNF	1-52	3-90	15-691	15-691#				
SPOLID	1-118	3-108#						
SPSTAT	1-61	2-105#						
SPUSR	1-99	2-70#						
SR3FLG	1-115	3-210#						
SRERR	17-15	17-15#	17-22	17-22#	17-29	17-29#	17-38	17-38#
SRTSIZ	1-116	3-348#						
STATCH	3-195	15-478#						
STPFLG	1-94	3-208#						
SWAPFL	3-177	15-51#	17-10					
SWDBLK	1-65	15-9#						
SWPCHN	1-102	2-200#	2-219					
SWPSLT	3-33	15-65#	17-10	17-10				
SYCHO	1-95	2-43#						
SYCH1	1-95	2-44#						
SYCH10	1-96	2-51#						
SYCH11	1-96	2-52#						
SYCH12	1-96	2-53#						
SYCH13	1-96	2-54#						
SYCH14	1-97	2-55#						
SYCH15	1-97	2-56#						
SYCH16	1-97	2-57#						
SYCH17	1-97	2-58#	2-210					
SYCH2	1-95	2-45#						
SYCH20	1-97	2-59#						
SYCH3	1-95	2-46#						
SYCH4	1-95	2-47#						
SYCH5	1-95	2-48#						
SYCH6	1-96	2-49#						
SYCH7	1-96	2-50#						
SYINDX	1-105	2-93#						
SYNAME	1-106	3-112#						
SYNCH	1-135	2-89						
SYPSPR	1-47	3-220#						
SYPSWD	1-47	15-406	15-406	15-406#				
SYSCHN	1-95	2-42#	2-209					
SYSDAT	1-98	2-63#	2-212					
SYSDMP	3-175	15-153#						
SYSGEN	1-105	2-96#						
SYSPSS	9-177#	15-406#	21-160					
SYSUPD	1-99	2-80#						
SYSVER	1-99	2-79#						

VMXMRB	1-80	3-30#								
VMXMSG	1-80	3-29#								
VMXSF	1-90	3-14#								
VMXSFC	1-90	3-15#								
VMXWIN	1-55	3-37#								
VNCSLO	1-71	3-46#								
VNCXOF	1-71	3-47#								
VNCXON	1-71	3-48#								
VNFCSH	1-85	3-20#								
VNGR	1-51	3-35#								
VNRFLG	1-69	3-42#								
VNUIP	1-51	3-39#								
VNUMDC	1-81	3-16#								
VOFFTM	1-87	3-26#								
VONTM	1-87	3-25#								
VPLAS	1-90	3-34#								
VPMSIZ	1-73	3-43#								
VPRIDF	1-99	3-188#								
VPRIHI	1-98	3-187#								
VPRILO	1-98	3-186#								
VPRIVR	1-64	3-189#								
VQUANO	1-73	3-4#								
VQUAN1	1-83	3-5#								
VQUAN2	1-83	3-9#								
VQUAN3	1-73	3-10#								
VQUN1A	1-83	3-6#								
VQUN1B	1-83	3-7#								
VQUN1C	1-69	3-8#								
VSCHED	1-72	3-44#								
VSLEDT	1-56	3-184#								
VSWPFL	1-92	3-177#								
VSWPSL	1-55	3-33#								
VSYDMP	1-70	3-175#								
VT100	1-136	17-18	17-26	17-33	17-44	17-53	17-62	17-71		
VT200	1-136									
VT52	1-136									
VTMIN	1-87	3-22#								
VTMLDC	1-87	3-23#								
VTMOUT	1-87	3-24#								
VTSLCH	1-73	3-190#								
VU*CL	1-52	3-181#								
VUCLMC	1-115	3-18#								
VUCLOR	1-115	3-182#								
VUSPHN	1-126	3-199#								
VUXIFL	1-62	3-180#								
VVLSCH	1-73	3-191#								
VVPWCH	1-73	3-192#								
W	13-7#	15-643								
WILDFL	1-64	15-526#								
WINBAS	1-80	3-71#								
X	15-574	15-574#	15-575	15-575#	15-576	15-576#	15-577	15-577#	15-578	15-578#

