

6.31

```

1          .TITLE  TSGEN -- System Generation Parameters
2          .IDENT  /V6.31/
3 000000   .CSECT  TSGEN
4          .ENABL  LC
5          .DSABL  GBL
6          .NLIST  CND
7          ;-----
8          ; TSGEN version 6.31
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.
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
47         .GLOBL  IOHANG, GETUMR, FREUMR, SYPSWD, SYPSPR, TSXVRS
48         .GLOBL  TSGEN, RDBSEC, DTYPE, NAMSEC, LNMTOP, SASECT, SMONHD
49         .GLOBL  NPL, NSL, NLIN2, LSTHL, TNHL, NIOL, FSTIOL
50         .GLOBL  NLINES, NLCHN, $MEMSZ, MEMPTR, PHYMEM, CFSTS, CFABLV
51         .GLOBL  VNGR, SHRCB, SHRCN, SCPFHD, VNUIP, INSTBL, INSTBN
52         .GLOBL  SPLND, SPLNF, SPLNB, VU$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, CLEQFS, 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, LNMAR, LNPRIM, LPRI, VLDSYS, NSPLFL, NSPLBL
61	. GLOBL	CLVERS, CLORSZ, VMXMON, MONFQH, 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, TMIDL, 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
72	. GLOBL	CMFSEC, CMFTOP, VKEYMX, VSCHED, VDSKBU, MODTIM
73	. GLOBL	VTSLCH, MXSPAC, PMSIZE, 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, VOFFTM
87	. GLOBL	VTMOUT, SNDBX, USPLCH, UBUSMP, MEM256
88	. GLOBL	FREFRK, BOTDEV, BOTUNI, BOTCSR, MIOFLG
89	. GLOBL	AUTHAN, AHEND, SNBUX, UMSYTP, VINTIO
90	. GLOBL	VMXSFC, VMXSFC, VMLBLK, RSFBLK, VPLAS, SEGCHN
91	. GLOBL	SDCBSZ, SDBULS, SPLANM, MIONWB, INDFIL
92	. GLOBL	NDL, LSTDLD, FSTDLD, VSWPFL, MAPUSR, R\$MFMV
93	. GLOBL	CORUSR, LINNUM, MUXNUM, NUMON, PVON, TOTON
94	. GLOBL	STPFLG, MINTIM, RTVECT, VHIPCT, FRKDLY
95	. GLOBL	SYSCHN, SYCH0, 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	INDDBL, INDTSV, INDDBS, R\$INTC
105	. GLOBL	SYINDX, 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	JCXPQS, 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

```

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$WCT
121 . GLOBL CSHBFP, CSHLRU, CSHMRU, CSHFHD, CCBHD
122 . GLOBL CASTBR, CASCBR, CASTBW, CASCUP, CASTRO, CASTWO
123 . GLOBL CSHIQ, CSHINI, CSHCLN, CSHFIN, CSHBAS, CSHVEC
124 . GLOBL LOKVEC, LOKINI, DOOPAP, DOCOPN, SFSVST, SFRSST, DORLK
125 . GLOBL DOTLK, DOCULK, DOULK1, DOSFCK, SFCLS, SFWRIT, CLSCDB
126 . GLOBL DCRD1, DCRD2, LOKVEC, CLKVEC, VUSPHN, ABRTOV
127 ;
128 ; Global references
129 ;
130 . GLOBL ININT, DLINT, LINNUM, PROITP, FNDHRB
131 . GLOBL SD$HLD, MUXNUM, RMNBAS, SETERR
132 . GLOBL IOFIN, INTEN, PSW, SYNCH
133 . GLOBL LA36, LA120, VT52, VT100, ADM3A, VT200
134 . GLOBL DIABLO, QUME, HAZEL, TSDEFS, EXTP1, HANXMR, BLKMV, CVTPHY
135 . GLOBL GETRTQ, QFREE, IOSTRT, FRKGET, FORKQ, QIO, QCOMPL
136 . GLOBL SCHED, DSKBUF, IOQSIZ

```

```

1      ; -----
2      ;   Internal parameters
3      ;
4      001167   TSXVRS   =      631.      ;TSX-Plus version number
5      000017   MAXDEV   =      15.      ;Max number of devices that can be supported
6      000036   NUMFRK   =      30.      ;# Fork request blocks
7      000000   NXIVMH   =      0.      ;# extra interrupt vectors for mapped handlers
8      000120   DETCBS   =      80.      ;# characters for detached job startup cmd.
9      000004   FRKGEN   =      4.      ;# Fork blocks in TSGEN
10     000004   NMSNMB   =      4.      ;# System message buffers
11     000002   NMSYMB   =      2.      ;# message buffer reserved for system use
12     000024   NUCHN    =      20.      ;# I/O channels user may use
13     000010   MAXMUX   =      8.      ;Max # of DZ11's, DH11's, and DHV11's
14     000040   DHBFSZ   =      32.      ;# bytes for DH11 and DHV11 DMA output buffers
15     000012   TTCSCH   =      10.      ;# characters printed per scheduler check
16     000050   NUMIOQ   =      40.      ;# I/O queue elements
17     000017   NUMSYQ   =      15.      ;# I/O queue elements reserved for system I/O
18     000017   NUMCCB   =      15.      ;# data cache control blocks
19     000010   NSCP     =      8.      ;# swapper command packets
20     000714   INTSSZ   =      460.     ;# bytes for system interrupt stack
21     000006   MIONWB   =      6.      ;Number of mapped I/O wait queue elements
22     000000   MIDDBG   =      0.      ;1=Force I/O mapping (debugging use only)
23     000000   MPARFL   =      0.      ;1==>Enable mem parity traps, 0==>disable.
24     000000   PROBRK   =      0.      ;1==>Enable ODT break on PRO printer port
25     000003   PROODC   =      3.      ;Clock ticks per PI driver call
26     000000   FASTIN   =      0.      ;1==>Clock driven input character processing
27     000100   CLKVEC   =      100     ;Clock interrupt vector
28     000006   DCAGE    =      6.      ;Shared file data cache ageing factor
29     000001   KUSECK   =      1.      ;0==>Don't check for usage on INIT & SQUEEZE.
30     000024   IOHMTM   =      20.     ;# 0.1 secs I/O can be held for job swapping
31     000007   CLEOFS   =      7.      ;Max num of chars in CL ENDSTRING parameter
32     000002   NDRTDF   =      2.      ;Number of dummy shared run-time definitions
33     000000   $PRIV    =      0       ;Obsolete line-def privilege flag
34     000020   $NOVLN   =      20     ;Obsolete no-virtual-line privilege flag
35     ;
36     ; -----
37     ;   Fork priority values.
38     ;   Unlike RT-11, TSX-Plus assigns priority values to its fork requests
39     ;   and allows higher priority fork requests to interrupt lower priority ones.
40     ;   The priority values range from 1 to 127.  The higher the numerical value,
41     ;   the higher the priority.
42     ;
43     . GLOBL   FP$RT, FP$CKT, FP$DEF, FP$IOS, FP$IOF, FP$IOA, FP$MOV
44     . GLOBL   FP$CDI, FP$CDO, FP$CK1, FP$MAX, FP$FLG, FP$PIO
45     ;
46     100000   FP$FLG   =      100000   ;Flag saying this is a priority value
47     100177   FP$MAX   =      FP$FLG+127. ;Max legal fork priority
48     100144   FP$RT    =      FP$FLG+100. ;Real-time interrupts
49     100106   FP$CKT   =      FP$FLG+70.  ;50/60 Hz clock interrupt processing
50     100074   FP$CDI   =      FP$FLG+60.  ;Terminal character input processing
51     100067   FP$CDO   =      FP$FLG+55.  ;Terminal character output processing
52     100062   FP$DEF   =      FP$FLG+50.  ;Default fork priority
53     100062   FP$IOF   =      FP$FLG+50.  ;I/O finish
54     100062   FP$IOA   =      FP$FLG+50.  ;I/O abort entry
55     100062   FP$PIO   =      FP$FLG+50.  ;PI output interrupt processing
56     100036   FP$CK1   =      FP$FLG+30.  ;0.1 second clock processing
57     ; The following fork priorities are entered from a non-interrupt state.

```

```

58      100014      FP$IOS  =      FP$FLG+12.      ; I/O initiation
59      100012      FP$MOV  =      FP$FLG+10.      ; Move data to/from cache buffer
60
61      ; -----
62      ; Completion routine class priorities.
63      ; A completion routine with a higher (numerically larger) class priority
64      ; is allowed to interrupt a lower class priority completion routine.
65      ;
66      . GLOBL  CP$STD, CP$RT, CP$SYN
67      ;
68      000001      CP$STD  =      1              ; Standard -- I/O completion, .TIMIO, etc.
69      000002      CP$RT   =      2              ; Real-time completion routine
70      000003      CP$SYN  =      3              ; .SYNCH completion routine
71      ;
72      ; Type codes used to identify communication device controllers
73      ;
74      000000      CDX$DL  =      0              ; DL11
75      000002      CDX$DZ  =      2              ; DZ11
76      000004      CDX$DH  =      4              ; DH11
77      000006      CDX$VH  =      6              ; DHV11
78      000010      CDX$PI  =      10             ; Console terminal on Professional
79      000012      CDX$PC  =      12             ; Communications port on Professional
80      000014      CDX$PP  =      14             ; Printer port on Professional
81      000016      CDX$QP  =      16             ; 4 line Multiplexer on Professional
82      ; -----
83      000024      CFCHAN  =      NUCHN          ; Channel to use for command file input
84      000025      LOGCHN  =      NUCHN+1        ; Channel for log file
85      000026      USPLCH  =      NUCHN+2        ; Channel to use to write to spool file
86      000027      RUNCHN  =      NUCHN+3        ; Channel to use when loading a SAV file
87      000030      USWPCH  =      NUCHN+4        ; Channel to use to access swap file
88      000031      NLCHN   =      NUCHN+5        ; Total # channels allocated per job
89      000000      LSTMX   =      0              ; Index to last mux
90      000000      CURMX   =      0              ; Current mux #
91      000000      NUMRDB  =      0              ; Count number of shared run-times declared
92      000000      CURCDX  =      CDX$DL        ; Comm device type for current line
93      000000      DHUSE   =      0              ; Set to 1 if DH11 or DHV11 support needed

```

```

1
2 ; -----
3 ; Monitor fixed-offset value vector
4 ;
5 ; Table of addresses of TSX-Plus routines. The pointer to this vector is
6 ; stored at simulated RMON offset -2.
7 ;
8 000000 002260' TSXVEC: .WORD NUMDEV ; Negative offsets from TSXVEC reserved to users
9 000002 000576' .WORD HANENT ; 0 Ptr to word with # devices in system
10 000004 001026' .WORD HANPAR ; 2 Vector with handler entry points
11 000006 000000G .WORD GETRTQ ; 4 64-byte phys mem base of mapped handlers
12 000010 000000G .WORD QFREE ; 6 Routine to get a free I/O queue element
13 000012 000000G .WORD QIO ; 10 Routine to free an I/O queue element
14 000014 000000G .WORD IOSTRT ; 12 Routine to queue an I/O request
15 000016 000000G .WORD QCOMPL ; 14 Routine to requeue an I/O request
16 000020 000000G .WORD FRKGET ; 16 Routine to queue a completion request
17 000022 000000G .WORD FORKQ ; 20 Routine to get a fork request block
18 000024 000000G .WORD IOHANG ; 22 Routine to queue a fork request
19 000026 000000G .WORD GETUMR ; 24 Place I/O queue element on handler list
20 000030 000000G .WORD FREUMR ; 26 Allocate a Unibus map register
21 000032 001167 .WORD TSXVRS ; 30 Free a Unibus map register
22 000034 000000G .WORD IOGSIZ ; 32 TSX-Plus version number
23 ; ; 34 Size of an I/O queue element (bytes)
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  QCOMP: .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 000336      004   SYSVER: .BYTE  4          ;276 - System version number
73 000337      000   SYSUPD: .BYTE  0          ;277 - Release #
74 000340 000000   CONFIG: .WORD  0          ;300 - System configuration word
75 000342          .BLKW  5          ;302 - 313 (unused)
76 000354 001750   MAXBLK: .WORD  MAXFIL          ;314 - Largest output file size
77                ; Word 316 in TSX-Plus is reserved for specific use. It must
78                ; be initialized to zero and not used by the operating system.
79                ; See the note in TSDEFS for more specific information.
80 000356 000000 000000 .WORD  0,0          ;316 - 321 unused
81 000362 000000   CORUSR: .WORD  0          ;322 - Current job number
82 000364 000000G  .WORD  SYNCH          ;324 - Address of .SYNCH request routine
83 000366          .BLKW  13          ;326 - 357 unused
84 000420 000445   BR      MTPS          ;360 - Move to PS routine
85 000422 000432   BR      MFPS          ;362 - Move from PS routine
86 000424 000000   SYINDX: .WORD  0          ;364 - Device number of system device
87 000426 000000   CFSTS:  .WORD  0          ;366 - Command file status flags
88 000430 000000   CONFQ2: .WORD  0          ;370 - Extended configuration word
89 000432 000000   SYSGEN: .WORD  0          ;372 - System generation options
90 000434 000002   .WORD  2          ;374 - Size of USR
91 000436      014   CFABLV: .BYTE  14          ;376 - Error abort severity level
92 000437      003   .BYTE  3          ;377 - Max @file nesting level
93 000440 000000   EMTRTN: .WORD  0          ;400 - EMT return point
94 000442 000000   FRKADR: .WORD  0          ;402 - Fork routine
95 000444 000500   PNPTR:  .WORD  PNAME-MONVEC ;404 - Offset to permanent dev name table
96 000446 071677 142615 MONAME: .RAD50 /RT11XM/ ;406 - 410 - System name
97 000452 000000   HSUFFX: .WORD  0          ;412
98 000454 000000   SPSTAT: .WORD  0          ;414 - Spooler status flags
99 000456      000   .BYTE  0          ;416 - Error byte for IND
100 000457      000   INSTA:  .BYTE  0          ;417 - IND status byte
101 000460 000000   $MEMSZ: .WORD  0          ;420 - Total 32-word mem blocks avail
102 000462 000000   .WORD  0
103 000464 000442G  $TCFIG: .WORD  TTOPTS+RMNBAS ;424 - Address of TT config word
104 000466 000444G  $INDDV: .WORD  INDOFF+RMNBAS ;426 - Pointer to IND device name word
105 000470 001300   MEMPTR: .WORD  HNMEPT-MONVEC ;430 - Offset to memory control blocks
106 000472 001330'  P1EXT:  .WORD  P1XPTR          ;432 - Kernel PAR1 routine
107 000474 000000   RPRCSR: .WORD  0          ;434 - Get CSR address of PRO devices
108 000476 000000   RPRVEC: .WORD  0          ;436 - Get vector address of PRO devices
109 000500 000000   DWTYPE: .WORD  0          ;440 - Type of DW disk
110                ; Dummy cell corresponding to cell in RT-11 with TT option flags.
111                TTOPTS =      .-MONVEC          ;Offset to TTOPTS cell
112 000502 000000   TTOP:  .WORD  0          ;TTOPTS cell
113                ; Cell with name of IND RUN device
114                INDOFF =      .-MONVEC          ;Offset to INDDEV cell

```

```

115 000504      123      131      060  INDDEV: .ASCII /SYO:/          ;Default device from which IND is run
      000507      072
116
117 ; -----
118 ; MFPS is called to return on the stack the contents of the low-order
119 ; byte of the processor status word.
120 ; Note: This only works when used within handlers. If a .MFPS macro
121 ; is used in a TSX-Plus user job, a value of zero is returned.
122 ;
123 ; Outputs:
124 ; Processor status word is on top of stack.
125 000510 005046 MFPS: CLR -(SP)
126 000512 013716 000000G MFPMOV: MOV @#PSW,(SP) ;Get the psw (** patched during job init **)
127 000516 016646 000002 MOV 2(SP),-(SP) ;Now push return address on top
128 000522 016666 000002 000004 MOV 2(SP),4(SP) ;Move down PS value
129 000530 012616 MOV (SP)+,(SP) ;Move down return address
130 000532 000207 RETURN
131 ;
132 ; MTPS is called to set the value of the low-order byte of the
133 ; processor status word.
134 ;
135 ; Inputs:
136 ; Value to be moved to psw is on top of stack before call.
137 ;
138 ; Outputs:
139 ; Value is moved to psw and popped from the stack.
140 ;
141 000534 000006 MTPS: RTT ;PC&PS are on stack, let RTT set PS and return
142 ;
143 ; -----
144 ; Device and handler information tables (Do not change the order).
145 ;
146 ; *** VM depends on PHYMEM being 1 word below PNAME ***
147 ;
148 000536 000000 PHYMEM: .WORD 0 ;*** Store actual physical memory size ***
149 000540 000017 PNAME: .REPT MAXDEV ;Table of permanent device names (Rad50)
150 .WORD 0
151 .ENDR
152 000576 000017 HANENT: .REPT MAXDEV ;Handler entry point
153 .WORD 0
154 .ENDR
155 000634 177777 .WORD -1 ;Flag to mark end of HANENT table
156 000636 000017 DVSTAT: .REPT MAXDEV ;Device status flags
157 .WORD 0
158 .ENDR
159 000674 000017 HANDSK: .REPT MAXDEV ;Location of handler on the disk
160 .WORD 0
161 .ENDR
162 000732 000017 HANSIZ: .REPT MAXDEV ;Size of device handler
163 .WORD 0
164 .ENDR
165 000770 000017 DEVSIZ: .REPT MAXDEV ;# 256-word blocks on device
166 .WORD 0
167 .ENDR
168 001026 000017 HANPAR: .REPT MAXDEV ;64-byte base block for handler if mapped
169 .WORD 0
170 .ENDR

```

```

171 001064 000017 HANIIOC: .REPT MAXDEV ;# uncompleted I/O requests for handler
172 .WORD 0
173 .ENDR
174 001122 000017 DVFLAG: .REPT MAXDEV ;Table of device characteristics
175 .WORD 0
176 .ENDR
177 001120 MAXGVL = .-MONVEC ;Max offset allowed with .GVAL
178 ;-----
179 ;
180 ; Reserve space for extended channel space
181 ;
182 001160 EXTCHN:
183 000010 .REPT <NLCHN-17.>
184 CHNRES
185 .ENDR
186 ;
187 ; Reserve space for channel used to access INDTMP file
188 ;
189 001300 INDTSV: CHNRES ;Channel used for I/O to INDTMP file
190 ;
191 ; Address of channel block for swap file access
192 ;
193 001266' SWPCHN = EXTCHN+<10.*<USWPCH-17.>>
194 ;
195 ; End of MONVEC pointer table area.
196 ;
197 001254 MVSIZ = .-VECBAS
198 000526 MVWDS = MVSIZ/2 ;# words in mon vector table
199 ;
200 ; Define offsets into monitor vector area
201 ;
202 000006 R$CHN = SYSCHN-VECBAS ;Start of channel space
203 000234 R$CH17 = SYCH17-VECBAS ;Offset to channel # 17
204 001122 R$XCHN = EXTCHN-VECBAS ;Offset to extended channel space
205 000264 R$DATE = SYSDAT-VECBAS ;Offset to date word
206 000324 R$JOB = CORUSR-VECBAS ;Offset to job number cell
207 000270 R$UBAS = USROFF-VECBAS ;Offset to usr base address cell
208 000444 R$TTOP = TTOP-VECBAS ;Offset to TT option word
209 000421 R$INST = INSTA-VECBAS ;Offset to IND status byte
210 000370 R$CFST = CFSTS-VECBAS ;Offset to command file status word
211 001242 R$INTC = INDTSV-VECBAS ;Offset to INDTMP channel block
212 001230 R$SWPC = SWPCHN-VECBAS ;Offset to USWPCH channel block
213 000454 R$MFMV = MFPMOV-VECBAS ;Offset to MFPMOV instruction
214 ;
215 ;-----
216 ; Vector of entry points to handler support routines
217 ; (Do not alter the order)
218 ;
219 001312 000167 000000G JMP CVTPHY ;Routine to convert virtual to physical addr
220 001316 000167 000000G JMP FNDHRB ;Routine to search for RCB for handler
221 001322 000167 000000G JMP HANXMR ;Routine to allocate XM region for handler
222 001326 000402 BR BMJMP ;Routine to do block move
223 001330 000167 000000G PIXPTR: JMP EXTP1 ;Routine to execute mapped code
224 001334 000167 000000G BMJMP: JMP BLKMV
225 ;
226 ;-----
227 ; Memory allocation information for handlers

```

228
229 001340 000000
230 001342 000000
231 001344 000000

;
HNMEPT: .WORD 0
.WORD 0
HANRCO: .WORD 0

```

1          ; -----
2          ;   Misc data cells
3          ;
4 001346   000002   VQUANO: .WORD   QUANO
5 001350   000024   VQUAN1: .WORD   QUAN1
6 001352   000002   VQUN1A: .WORD   QUAN1A
7 001354   000002   VQUN1B: .WORD   QUAN1B
8 001356   000001   VQUN1C: .WORD   QUAN1C
9 001360   000012   VQUAN2: .WORD   QUAN2
10 001362   000024   VQUAN3: .WORD   QUAN3
11 001364   000002   VCORTM: .WORD   CORTIM
12 001366   000050   VHPRCT: .WORD   HIPRCT
13 001370   000036   VINTIO: .WORD   INTIOC
14 001372   000036   VMXSF:  .WORD   MAXSF
15 001374   000036   VMXSFC: .WORD   MAXSFC
16 001376   000000   VNUMDC: .WORD   NUMDC
17 001400   000003   VMLBLK: .WORD   MXLBLK
18 001402   000005   VUCLMC: .WORD   UCLMNC
19 001404   001750   VMXFIL: .WORD   MAXFIL
20 001406   000050   VNFCSH: .WORD   NMFCSH
21 001410   000005   VMXMON: .WORD   MAXMON
22 001412   000036   VTMOUT: .WORD   TIMOUT
23 001414   000074   VOFFTM: .WORD   OFFTIM
24 001416   000003   VMAXMC: .WORD   MAXMC
25 001420   000310   VMSCHR: .WORD   MSCHRS
26 001422   000003   VMXMSG: .WORD   MAXMSG
27 001424   000012   VMXMRB: .WORD   MAXMRB
28 001426   000100   VHIMEM: .WORD   HIMEM
29 001430   000070   VDFMEM: .WORD   DFLMEM
30 001432   000007   VSWPSL: .WORD   SWPSLT
31 001434   000000   VPLAS:  .WORD   SEGBLK
32 001436   000014   VNGR:   .WORD   NGR
33 001440   000004   NDVRCB: .WORD   DEVXMR
34 001442   000012   VMXWIN: .WORD   MAXWIN
35 001444   000007   VKEYMX: .WORD   KEYMAX
36 001446   000004   VNUIP:  .WORD   NUIP
37 001450   000000   VCSHNB: .WORD   CACHE
38 001452   000000   CSHALC: .WORD   CACHE
39 001454   040150   VNRFLG: .WORD   NRMFLG
40 001456   000000G  VSCHED: .WORD   SCHED
41 001460   000000G  VDSKBU: .WORD   DSKBUF
42 001462   000040   VNCALO: .WORD   NCSILO
43 001464   014      VNCXOF: .BYTE   NCXOFF
44 001465   004      VNCXON: .BYTE   NCXON
45 001466   000144   VDISPC: .WORD   DINSPC
46 001470   000360   VDOSPC: .WORD   DOTSPC
47 001472   000000   SYTIMH: .WORD   0
48 001474   000000   SYTIML: .WORD   0
49 001476   000000   TK1SEC: .WORD   0
50 001500   000000   TK1VAL: .WORD   0
51 001502   000000   TK1CNT: .WORD   0
52 001504   000000   TK5VAL: .WORD   0
53 001506   000000   TK3SVL: .WORD   0
54 001510   000000   TSXSIT: .WORD   0
55 001512   000000   FRKDLY: .WORD   0
56 001514   000000   CTRLTT: .WORD   0
57 001516   000000   MINTIM: .WORD   0

```

; # of job slots in swap file
; # blocks for PLAS swap file
; Number of global PLAS regions
; Number of PLAS regions for device handlers
; Maximum number of display windows
; Maximum # user-defined keys
; Number of user programs that may be INSTALLED
; # blocks in use for generalized data cache
; # blocks allocated for generalized data cache
; Default time-sharing line flags
; An entry point in TSEXEC
; A global from TSINIT
; Default #bytes for TT and CL silos
; Default XOFF when only this many free
; Default XON when this many remain
; Default line input buffer size
; Default line output buffer size
; High-order system time word
; Low-order system time word
; # clock ticks per second
; # clock ticks per 0.1 second

; # clock ticks per 0.5 seconds
; # clock ticks per 3 seconds

; Max clock ticks a fork request was delayed
; # of operator's console
; Number of minutes of system up-time

58	001520			SEGCHN: .BLKW	5		; Channel block used for PLAS region swapping
59	001532	000000		KMNTOP: .WORD	0		; Abs address of top of TSKMON
60	001534	000000		KMNHI: .WORD	0		; KMNTOP-KMNBAS
61	001536			CCLSAV: .BLKW	5		; Savestatus for CCL.SAV file info
62	001550			INDSAV: .BLKW	5		; Savestatus for IND.SAV file info
63	001562	000000		INDDBL: .WORD	0		; Lowest block in IND.SAV file of data segment
64	001564	000000		INDDBS: .WORD	0		; Number of blocks in IND.SAV data segment
65	001566	000000		USRBAS: .WORD	0		; Phys 64-byte block # of TSUSR overlay
66	001570	000000		MSGBAS: .WORD	0		; Phys 64-byte block # of TSMSG overlay
67	001572	000000		WINBAS: .WORD	0		; Phys 64-byte block # of TSWIN overlay
68	001574	000000		LOKBAS: .WORD	0		; Phys 64-byte block # of TSLOCK overlay
69	001576	000000		CSHBAS: .WORD	0		; Phys address of TSCASH code
70	001600	000000		TIOBAS: .WORD	0		; Phys address of TSTIOX code
71	001602	000000		LOKMEM: .WORD	0		; Phys 64-byte block # of rec locking data area
72	001604	000000		LOKCSH: .WORD	0		; Phys 64-byte block # of shared file cache buf
73	001606	000000		NUMDCD: .WORD	NUMDC		; Num of shared file data cache entries
74	001610	000036		NUMCDB: .WORD	MAXSFC		; Number of free shared file channels
75	001612	000000		SNMSHD: .WORD	0		; Head of free list of system message buffers
76	001614	000002		NMUMB: .WORD	<NMSNMB-NMSYMB>		; # message buffers available for user access
77	001616	000000		CSHHD: .WORD	0		; Head of directory cache list
78	001620	000000		MONFGH: .WORD	0		; Head of free list of monitor control blocks
79	001622	000000		MIOBHD: .WORD	0		; Head of mapped I/O control block list
80	001624	000000		MIOWHD: .WORD	0		; Head of mapped I/O wait block list
81	001626	000000		MIOSYQ: .WORD	0		; Pointer to 1st active mapped I/O wait block
82	001630	000000		SJMONHD: .WORD	0		; Head of job monitoring requests for all jobs
83	001632	000000		SFCB: .WORD	0		; Start of spool file control block area
84	001634	000000		SFCBND: .WORD	0		; End of spool file control block area
85	001636	000000		SFCBFH: .WORD	0		; Head of free spool file control block list
86	001640	000024		NSPLFL: .WORD	SPLNF		; Number of spool files
87	001642	000764		NSPLBL: .WORD	SNDBX		; Number of blocks in spool file
88	001644	000000		NFRESB: .WORD	0		; Number of public spool file blocks
89	001646	000000		SHRRCB: .WORD	0		; Pointer to base of global RCB area
90	001650	000000		SHRRCN: .WORD	0		; Pointer to end of global RCB area
91	001652	000000		INSTBL: .WORD	0		; Pointer to base of INSTALL table
92	001654	000000		INSTBN: .WORD	0		; Pointer past end of INSTALL table
93	001656	000000		ABRTOV: .WORD	0		; Rad50 name of overlay during trap
94	001660	000012		VMXCSH: .WORD	MAXCSH		; Max number of cached devices
95	001662	000000		CSHDEV: .WORD	0		; Start of area with device cache blocks
96	001664	000000		CSHDVN: .WORD	0		; End of area with device cache blocks
97	001666	000000		SCPFHD: .WORD	0		; Head of free list of swap command packets
98	001670	177777		LDDEVX: .WORD	-1		; Device index number of "LD" device
99	001672	177777		CLDEVX: .WORD	-1		; Device index number of "CL" device
100	001674	177777		C1DEVX: .WORD	-1		; Device index number of "C1" device
101	001676	000000	000000	BOTDEV: .WORD	0,0,0,0		; Device spec for device being booted from
	001704	000000					
102	001706	000000		BOTUNI: .WORD	0		; Unit # of device being booted from
103	001710	000000		BOTCSR: .WORD	0		; CSR of device being booted from
104	001712	000000		SPOLID: .WORD	0		; Last spool file ID number
105	001714	000000		UMSYTP: .WORD	0		; Address of top of unmapped system space
106	001716	000001		IOABFL: .WORD	IOABT		; 1==>Do I/O abort, 0==>Do I/O wait
107	001720	000000G		DEFBAS: .WORD	TSDEFS		
108	001722	114716		SYNAME: .RAD50	/XXN/		; Actual name of SY physical device
109	001724	075250	100020	UCLNAM: .RAD50	/SY TSXUCLSAV/		; Name of TSXUCL program
	001732	073376					
110	001734	000000		UCLBLK: .WORD	0		; # blocks in TSXUCL data file for each job
111	001736	075250	102405	UKMNAM: .RAD50	/SY UKMON SAV/		; Name of user-provided TSKMON command processr
	001744	073376					

```

112 001746 000000 PIDPTR: .WORD 0 ;Pointer to clock-driven PI handler routine
113 001750 PROSLT: .BLKW 9 ;ID # of device in each PRO option slot
114 001772 000000 VIDCSR: .WORD 0 ;Address of PRO video CSR
115 001774 177560 VDMTCR: .WORD DMPTCR ;Transmitter control reg addr for dump device
116 001776 000000 MODDAT: .WORD 0 ;Date last modified by TSXMOD
117 002000 000000 MODTIM: .WORD 0 ;Time (3-sec) last modified by TSXMOD
118 002002 000000 HANRCB: .WORD 0 ;Pointer to start of handler RCB area
119 ;
120 ; Data for generalized data cache
121 ;
122 002004 000000 CA$BLK: .WORD 0 ;Block number associated with cache entry
123 002006 000000 CA$DVU: .WORD 0 ;Device and unit # associated with entry
124 002010 000000 CA$WCT: .WORD 0 ;Number of words in entry
125 002012 000000 CA$UFL: .WORD 0 ;LRU chain forward link
126 002014 000000 CA$UBL: .WORD 0 ;LRU chain backward link
127 002016 000000 CA$HFL: .WORD 0 ;Hash chain forward link
128 002020 000000 CA$HBL: .WORD 0 ;Hash chain backward link
129 002022 000000 CA$HSH: .WORD 0 ;Hash chains list head vector
130 002024 000000 CSHBFP: .WORD 0 ;64-byte block number of buffer area
131 002026 000000 CSHLRU: .WORD 0 ;Pointer to least-recently-used entry
132 002030 000000 CSHMRU: .WORD 0 ;Pointer to most-recently-used entry
133 002032 000000 CSHFHD: .WORD 0 ;Head of cache block free list
134 002034 000000 CCBHD: .WORD 0 ;Head of cache control block free list
135 002036 000000 000000 CASTRO: .WORD 0,0 ;Total # reads from mounted devices
136 002042 000000 000000 CASTBR: .WORD 0,0 ;Total # blocks read from mounted devices
137 002046 000000 000000 CASCBR: .WORD 0,0 ;Number of blocks that were read from cache
138 002052 000000 000000 CASTWO: .WORD 0,0 ;Total # writes to mounted devices
139 002056 000000 000000 CASTBW: .WORD 0,0 ;Total # blocks written to mounted devices
140 002062 000000 000000 CASCUP: .WORD 0,0 ;Number of blocks moved into data cache
141 ; Entry point vector for caching module
142 002066 CSHVEC:
143 002066 000000 CSHINI: .WORD 0 ;-
144 002070 000000 CSHIO: .WORD 0 ;-
145 002072 000000 CSHCLN: .WORD 0 ;-
146 002074 000000 CSHFIN: .WORD 0 ;-
147 002076 177777 .WORD -1 ;- End of pointer vector
148 ;
149 ; Entry point vector for record locking module
150 ;
151 002100 LOKVEC:
152 002100 000000 LOKINI: .WORD 0 ;-
153 002102 000000 DOOPAP: .WORD 0 ;-
154 002104 000000 DOCOPN: .WORD 0 ;-
155 002106 000000 SFSVST: .WORD 0 ;-
156 002110 000000 SFRSST: .WORD 0 ;-
157 002112 000000 DORLK: .WORD 0 ;-
158 002114 000000 DOTLK: .WORD 0 ;-
159 002116 000000 DOCULK: .WORD 0 ;-
160 002120 000000 DOULK1: .WORD 0 ;-
161 002122 000000 DOSFCK: .WORD 0 ;-
162 002124 000000 SFCLS: .WORD 0 ;-
163 002126 000000 SFWRIT: .WORD 0 ;-
164 002130 000000 CLSCDB: .WORD 0 ;-
165 002132 000000 DCRD1: .WORD 0 ;-
166 002134 000000 DCRD2: .WORD 0 ;-
167 002136 177777 .WORD -1 ;- End of pointer vector
168 ;

```

```

169          ; Misc byte data
170          ;
171 002140    000    VSYDMP: .BYTE    SYSDMP          ;Generate dump on crash if non-zero
172 002141    000    VDMKTP: .BYTE    DMPKTP          ;Crash on any kernel trap if non-zero
173 002142    001    VSWPFL: .BYTE    SWAPFL
174 002143    001    VBUSTP: .BYTE    BUSTYP
175 002144    000    VINABT: .BYTE    INIABT
176 002145    001    VUXIFL: .BYTE    UXIFLG
177 002146    001    VU$CL:  .BYTE    U$CL
178 002147    002    VUCLOR: .BYTE    UCLORD
179 002150    001    VLDSYS: .BYTE    LDSYS
180 002151    001    VSLEDT: .BYTE    SLEDIT
181 002152    000    VDBFLG: .BYTE    DBGFLG
182 002153    023    VPRILO: .BYTE    PRILOW
183 002154    120    VPRIHI: .BYTE    PRIHI
184 002155    062    VPRIDF: .BYTE    PRIDEF
185 002156    012    VPRIVR: .BYTE    PRIVIR
186 002157    035    VTSLCH: .BYTE    TSLICH
187 002160    027    VVLSCH: .BYTE    VLSWCH
188 002161    002    VVPWCH: .BYTE    PWCH
189 002162    034    VCXTRM: .BYTE    CCXTRM
190 002163    001    VCXCTL: .BYTE    CCXCTL
191 002164    003    VEDIT:  .BYTE    EDITOR
192 002165    001    VMIOBF: .BYTE    MIONBF
193 002166    010    VMIOSZ: .BYTE    MIOBSZ
194 002167    000    VUSPHN: .BYTE    PHONE          ;0=local if no DCD;1=always mon DCD if $phone
195 002170    000    MAPUSR: .BYTE    0              ;Number of job memory mapping is set up for
196 002171    000    LINNUM: .BYTE    0
197 002172    000    MUXNUM: .BYTE    0
198 002173    000    NUMON:  .BYTE    0
199 002174    000    PVON:   .BYTE    0
200 002175    000    TONON:  .BYTE    0
201 002176    000    PROFLG: .BYTE    0              ;Non-zero ==> Running on PRO-350
202 002177    000    STPFLO: .BYTE    0
203 002200    000    UBUSMP: .BYTE    0              ;1==>Do Unibus mapping
204 002201    000    SR3FLG: .BYTE    0              ;NON-ZERO==>MEMORY MANAGEMENT REG 3 PRESENT
205 002202    000    MEM256: .BYTE    0              ;Non-zero==>machine has at least 256kb
206 002203    000    MIOFLG: .BYTE    0              ;Non-zero==>I/O mapping needed for some device
207 002204    000    NSPLDV: .BYTE    0              ;Number of installed spooled devices
208 002205    000    CLVERS: .BYTE    CLVRSN          ;CL handler version number
209 002206    000    LDVERS: .BYTE    0              ;LD translation table format (1<RTV5.4=<2)
210 002207    041    200    SYSPR:  .ASCII  /!/<200>    ;Prompt for system password
211          .EVEN
212          ; System time counters
213 002212    000000  TMTOTH: .WORD    0              ;Total uptime (0.1 second units)
214 002214    000000  TMTOTL: .WORD    0
215 002216    000000  TMUSRH: .WORD    0              ;Time spent in user jobs
216 002220    000000  TMUSRL: .WORD    0
217 002222    000000  TMSWTH: .WORD    0              ;Swap-wait time
218 002224    000000  TMSWTL: .WORD    0
219 002226    000000  TMIOH:  .WORD    0              ;Time user i/o is active
220 002230    000000  TMIOL:  .WORD    0
221 002232    000000  TMSWPH: .WORD    0              ;Time swapping is active
222 002234    000000  TMSWPL: .WORD    0
223 002236    000000  TMIOWH: .WORD    0              ;Time system is doing i/o-wait
224 002240    000000  TMIOWL: .WORD    0
225 002242    000000  TMIDLH: .WORD    0              ;Idle time

```

```

226 002244 000000 TMIDLL: .WORD 0
227 ;
228 ; Shared file data cache statistics counters
229 ;
230 002246 000000 DCTOTU: .WORD 0 ; Total number of cache hits since last divisn
231 002250 000000 DCTRD: .WORD 0 ; Total number of reads from shared files
232 002252 000000 DCCRD: .WORD 0 ; Number of reads satisfied by data in cache
233 002254 000000 DCTWR: .WORD 0 ; Total number of writes to shared files
234 002256 000000 DCCWR: .WORD 0 ; Number of writes that update cache
235 ;
236 002260 000000 NUMDEV: .WORD 0 ; Byte index to last entry in device tables
237 002262 000000 FREIOQ: .WORD 0 ; Head of i/o queue element chain
238 ;
239 ; Define mux tables for DZ11's and DH11's.
240 ;
241 . MACRO MXTBL NAME
242 . NLIST
243 NAME = . -2
244 . GLOBL NAME
245 . REPT MAXMUX
246 . WORD 0
247 . ENDR
248 . LIST
249 . ENDM MXTBL
250 ;
251 002264 MXTBL MXTYPE ; DZ11 & DH11 type of mux (CDX$DZ or CDX$DH)
252 002304 MXTBL MXCSR ; DZ11 Control Status Register
253 002324 MXTBL MXLPR ; DZ11 Line Parameter Register
254 002344 MXTBL MXTCR ; DZ11 Transmit Control Register
255 002364 MXTBL MXDTR ; DZ11 Data Terminal Ready
256 002404 MXTBL MXTBUF ; DZ11 Transmitter Buffer Register
257 002424 MXTBL MXSBRK ; DZ11 Shadow register for hardware BRK reg.
258 002444 MXTBL MXCAR ; DZ11 Carrier Detect
259 002464 MXTBL MXVEC ; DZ11 & DH11 Vector address
260 002504 MXTBL MXLNT ; DZ11 & DH11 Addr of table to map mux # to Lin
261 002524 MXTBL MH$BRK ; DH11 Break control register
262 002544 MXTBL MH$LPR ; DH11 Line Parameter Register
263 002564 MXTBL MH$PBR ; DH11 Previous value of BAR register
264 002604 MXTBL DM$CSR ; DH11(DM11) Control Status Register
265 002624 MXTBL DM$LSR ; DH11(DM11) Line Status Register
266 002644 MXTBL DM$VEC ; DH11(DM11) Address of DM11 interrupt vector
267 002322' MXRBUF = MXLPR ; DZ11 Receiver Buffer Register
268 002402' MXRING = MXTBUF ; DZ11 Ring indicator flags
269 002442' MXBRK = MXCAR ; DZ11 Break control flags
270 ; Equates for DH11 control registers
271 002302' MH$SCR = MXCSR ; DH11 System Control Register
272 002322' MH$RCR = MXRBUF ; DH11 Received Character Register
273 002342' MH$CAR = MXTCR ; DH11 Current Address Register
274 002362' MH$BCR = MXDTR ; DH11 Byte Count Register
275 002402' MH$BAR = MXTBUF ; DH11 Buffer Active Register
276 002442' MH$SSR = MXCAR ; DH11 Silo Status Register
277 ; Equates for DHV11 control registers
278 002302' VH$CSR = MH$SCR ; DHV11 Control and Status Register
279 002322' VH$DBR = MH$RCR ; DHV11 Data Buffer Register
280 002542' VH$LPR = MH$LPR ; DHV11 Line Parameter Register
281 002622' VH$LSR = DM$LSR ; DHV11 Line Status Register
282 002602' VH$LCR = DM$CSR ; DHV11 Line Control Register

```

```

283          002342'      VH$BA1 =      MH$CAR      ;DHV11 Buffer Address register 1
284          002442'      VH$BA2 =      MH$SSR      ;DHV11 Buffer Address register 2
285          002362'      VH$BCR =      MH$BCR      ;DHV11 Byte Count Register
286
287          ;          Generate FORK request blocks.
288          ;
289 002664 002670'      FREFRK: .WORD  FRKLST      ;Head of free list
290 002666 000000      FRKINI: .WORD  0          ;Pointer to fork blocks in init area
291 002670
292          000004      .REPT  FRKGEN      ;Gen in a few static fork blocks
293          .WORD  .+22.      ;Link to next block in free list
294          .WORD  0,0,0,0,0,0,0,0,0,0
295          .ENDR
296 003020 000000 000000 000000      .WORD  0,0,0,0,0,0,0,0,0,0 ;Last block with 0 forward link
      003026 000000 000000 000000
      003034 000000 000000 000000
      003042 000000 000000
297          ;
298          ; Symbolic equates for QBUS and UNIBUS machines.
299          ;
300          000001      QBUS =      1
301          000000      UNIBUS =      0
302          ;
303          ; Generate the memory size limit checking certain restrictions.
304          ; On non-extended machines, allow 256.Kb - 8.Kb I/O page
305          ; On extended machines, allow 4096.Kb - 256.Kb I/O page
306          ;
307          .MACRO  MEMORY  SIZE
308          SIZMEM = SIZE
309          .IF    LE, SIZMEM
310          SIZMEM = 3840.
311          .ENDC
312          .IF    GT, SIZMEM-3840.
313          SIZMEM = 3840.
314          .ENDC
315          .IF    LT, <SIZMEM - 96.>
316          .ERROR ;Memory size limit too small for running TSX-Plus
317          .ENDC
318          ; Allocate the memory size to examine.
319          .WORD  SIZMEM*20
320          .ENDM  MEMORY
321          ;
322          ; Memory management tables
323          ;
324 003046      MAPSIZ: MEMORY  MEMSIZ      ;PAR value of physical memory cutoff
325 003050 000000      BASMAP: .WORD  0      ;Pointer to base of memory map table
326 003052 000000      LOMAP: .WORD  0      ;Pointer to 1st user page in MEMMAP
327 003054 000000      HIMAP: .WORD  0      ;Pointer above top user page in memmap
328 003056 000000      MAPPAR: .WORD  0      ;Value to map PAR 5 to mem allocation table
329 003060 000000      FREPGS: .WORD  0      ;# free pages
330 003062 000000      JCXPGS: .WORD  0      ;# pages needed for job context block
331 003064 000000      MXJMEM: .WORD  0      ;Max # K-bytes a job may use
332 003066 000000      DFJMEM: .WORD  0      ;Default # K-bytes a job may use
333 003070 000000      MXJADR: .WORD  0      ;Address above top of largest job space
334 003072 000000      SMRSIZ: .WORD  0      ;# 64-byte blocks allocated to system overlays
335 003074 000000      MHNSIZ: .WORD  0      ;# 64-byte blocks allocated for mapped handler
336 003076 000000      SRTSIZ: .WORD  0      ;# 64-byte blocks allocated for shared run-tim

```

```

337 003100 000000      CSHSIZ: .WORD 0          ;# 64-byte blocks allocated for data cache
338                      ;
339                      ; Start a CSECT to hold shared run-time descriptor blocks
340                      ;
341 000000              .CSECT RDBSEC          ;CSECT for RDB entries
342 000000      RDBSEC:
343 000000      RDB:              ;Define base of RDB entries
344 003102              .CSECT TSGEN          ;Go back to standard TSGEN CSECT
345                      ;
346                      ; Symbolic equates for system editor names.
347                      ; Note these equates must match those in TSDEFS.
348                      ;
349                      000001      EDIT    =      1
350                      000002      TECO   =      2
351                      000003      KED    =      3
352                      000004      K52    =      4
353                      ;
354                      ; Symbolic equates for UCL order
355                      ;
356                      000001      FIRST  =      1
357                      000002      MIDDLE =      2
358                      000003      LAST   =      3
359                      ;
360                      ; Symbolic names used to define line transmit/receive speeds.
361                      ;
362                      000000      S50    =      0          ;50      baud
363                      000001      S75    =      1          ;75      baud
364                      000002      S110   =      2          ;110     baud
365                      000003      S134.5 =      3          ;134.5   baud
366                      000004      S150   =      4          ;150     baud
367                      000005      S300   =      5          ;300     baud
368                      000006      S600   =      6          ;600     baud
369                      000007      S1200  =      7          ;1200    baud
370                      000010      S1800  =      10         ;1800    baud
371                      000011      S2000  =      11         ;2000    baud
372                      000012      S2400  =      12         ;2400    baud
373                      000013      S3600  =      13         ;3600    baud
374                      000014      S4800  =      14         ;4800    baud
375                      000015      S7200  =      15         ;7200    baud
376                      000016      S9600  =      16         ;9600    baud
377                      000017      S19200 =      17         ;19200   baud
378                      ;
379                      ; Symbolic names for parity codes
380                      ;
381                      040000      EVEN   =      040000      ;Even parity
382                      140000      ODD    =      140000      ;Odd parity
383                      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      NOMDUNT     =     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       =      0          ; NONDMA
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             NIOL     =      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 <NIOL-16.>
82             . ERROR ; You cannot have more than 16 CL units
83             NIOL     =      16.           ;Reduce number to 16.
84             . ENDC
85             . IF      GT <<NIOL+CLXTRA>-16.>
86             . ERROR ; You cannot have more than 16 CL units
87             CLXTRA   =      16.-NIOL     ;Reduce extra units if total > 16
88             . ENDC
89             CLTOTL   =      NIOL+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             FSTDL    =      LSTPL+2       ;First detached line
100            LSTDL    =      LSTPL+<2*NDL> ;Last detached line
101            FSTSL    =      LSTDL+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*<NIOL-1>> ;Index to last CL line
106            LSTLIN   =      2*<NPL+NSL+NDL+NIOL> ;Index number of last line
107            . IF      EQ, NIOL           ;If there are no CL lines
108            TNHL     =      NPL           ;Total number of lines with hardware-ctrl tbls
109            . IFF     ;If there are CL lines
110            TNHL     =      NPL+NSL+NDL+NIOL ;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
119 . IF GT, <SWPSLT-NLINES> ; Never need more slots than lines
120 SWPSLT = NLINES
121 . ENDC
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 OB 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 OB LSW3 ; Additional line status flags
138 OB LSW4 ; Additional line status flags
139 OB LSW5 ; More line status flags
140 OB 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 OB LSW10 ; Line status table # 10
145 OB LSW11 ; Line status table # 11
146 OB LCLUNT ; CL unit index number if connected as CL line
147 OB ITRMTP ; Initial terminal type code
148 OB 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 OB 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      ; The CLDEF macro begins a line definition block for a serial communications
3      ; line that will be used as a dedicated CL line.
4      ; The CLDEF macro is similar to a LINDEF and can occur inside or outside
5      ; of a MUXDEF block.
6      ; The form of the CLDEF macro outside a MUXDEF block is:
7      ;     CLDEF  line_number,vector_address,RSR_address
8      ; The form of the CLDEF macro inside a MUXDEF block is:
9      ;     CLDEF  line_number,mux_line_number
10     ;
11     .MACRO  CLDEF  AIOLN,ARG1,ARG2
12     ;
13     ; Check to make sure the CL unit number is valid
14     ;
15     IOLN   =      AIOLN
16     .IF    GE <IOLN-CLTOTL>
17     .ERROR ;0 CL unit number exceeds # declared CL units
18     IOLN   =      0
19     .ENDC
20     ;
21     ; See if this CL unit has already been assigned to another line
22     ;
23     .IF    NDF      CLUD'AIOLN
24     CLUD'AIOLN =      1
25     .ENDC
26     .IF    GT <CLUD'AIOLN-2>
27     .ERROR ;CL unit AIOLN used more than once
28     .ENDC
29     CLUD'AIOLN =      CLUD'AIOLN+1
30     ;
31     ; Set flag saying we are doing an CLDEF definition and then invoke LINDEF.
32     ; Note, the LINEND macro will reset IOLFLG.
33     ;
34     IOLFLG =      1          ;We are inside CLDEF
35     LINDEF  ARG1 ARG2
36     .ENDM  CLDEF
37     ;
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     LN      =      0          ;Current line number
50     BO      =      0          ;1 if inside LINDEF block
51     LX      =      0          ;Line index number
52     IOLFLG  =      0          ;1 if inside an CLDEF block
53     IOLN    =      0          ;Number of dedicated CL line
54     NPLDF   =      0          ;Number of declared primary T/S lines
55     NCLDF   =      0          ;Number of CL lines that have been defined
56     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 ; -----
3 ; The FLAGS macro is used to set flags in the ILSW2 table.
4 ; The one argument to flags is the value to be stored
5 ; in ILSW2.
6 ;
7 ; .MACRO  FLAGS  AFLG
8 DFLAGS =      AFLG      ;SAVE FOR LINEND
9 ; .ENDM  FLAGS
10 ; -----
11 ; The TRMTYP macro is used to declare the terminal type.
12 ;
13 ; .MACRO  TRMTYP  ATYP
14 ; .IF    EQ,IOLFLG      ;Do not do for CL lines
15 S      =      .
16 ; .WORD  ATYP
17 ;      =      S
18 ; .ENDC
19 ; .ENDM  TRMTYP
20 ;
21 ; -----
22 ; The NAME macro is used within a line definition block to declare
23 ; a commentary name for the line which is displayed with the
24 ; SHOW TERMINALS keyboard command.
25 ;
26 000000 ; .CSECT  NAMSEC
27 000000 NAMSEC:
28 003102 ; .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 ; -----
44 ; The BUFSIZ macro is used to set the size of
45 ; the input and output character buffers.
46 ; Arg 1 = Input buffer size (# of characters)
47 ; Arg 2 = output buffer size (# of characters)
48 ;
49 ; .MACRO  BUFSIZ  AIS,AOS
50 DIS     =      AIS
51 ; .IF    NB AOS
52 DOS     =      AOS      ;SET OUTPUT BUFFER SIZE
53 ; .ENDC
54 ; .ENDM  BUFSIZ
55 ;
56 ; -----
57 ; The SILO macro is used to set up information about the

```

```

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 ALINES
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 000000 ; .CSECT CMFSEC
81 000000 CMFSEC:
82 003102 ; .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
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171

007000

```

XPAR      =      100
CARLEN    =      20
          .IF      GT,<APAR-1>
XPAR      =      0
CARLEN    =      30
          .ENDC
LSTPRM    =      <ASPD*400>
          .ENDM   LINPRM
;
;-----
; The SPEED macro is used to specify baud rates for lines
; as well as number of data bits and parity selection.
; The default is 9600 baud with 8. data bits and no parity.
; The form of the macro is:
;
;       SPEED      speedcode, data_bits, parity
;
; where      speedcode is selected from the speed code table
;            and is of the form S9600, for example
; and      data_bits = 7 or 8.
; and      parity = EVEN, ODD or NONE
;
LSTPRM    = <S9600*400>!<20000*0>!NONE      ; Default to 4800, 8, N
;
          .MACRO   SPEED      SPDCOD, NBITS, PARCOD
          .IF      DF, S'SPDCOD
SPDVAL    =      S'SPDCOD
          .IFF     ; DF, S'SPDCOD
          .IF      DF, SPDCOD
SPDVAL    =      SPDCOD
          .IFF     ; DF, SPDCOD
          .ERROR   0; Invalid speed specified with SPEED macro
SPDVAL    =      14.
          .ENDC   ; NDF, S'SPDCOD
          .ENDC   ; DF, S'SPDCOD
          .IF      B, NBITS
NDBITS    =      8.
          .IFF     ; B, NBITS
NDBITS    =      NBITS
          .IF      LT, <NDBITS-7>
          .ERROR   ; SPEED macro only accepts 7 or 8. data bits
NDBITS    =      8.
          .ENDC   ; LT, <NDBITS-7>
          .IF      GT, <NDBITS-8. >
          .ERROR   ; SPEED macro only accepts 7 or 8. data bits
NDBITS    =      8.
          .ENDC   ; GT, <NDBITS-8. >
          .ENDC   ; B, NBITS
          .IF      B, PARCOD
PARITY    =      NONE
          .IFF     ; B, PARCOD
PARITY    =      PARCOD
          .IF      NE, PARITY      ; NOT NONE?
          .IF      NE, <PARITY-EVEN> ; NOT EVEN?
          .IF      NE, <PARITY-ODD>  ; NOR ODD?
          .ERROR   ; Parity must be EVEN, ODD or NONE in SPEED macro
PARITY    =      NONE

```

```

172             . ENDC      ; NOT ODD
173             . ENDC      ; NOT EVEN
174             . ENDC      ; EVEN OR ODD
175             . ENDC      ; B, PARCOD
176 LSTPRM      =          <SPDVAL*400>!<20000*<B.-NDBITS>>!PARITY
177             . ENDM      SPEED
178
179 ; -----
180 ; The VECCHK macro is called to see if a line vector
181 ; address is reasonble.
182 ;
183             . MACRO     VECCHK  VA, MASK
184 VERR=0
185             . IIF      LT, VA-60          VERR=1
186 ; Although RT-11 V5.3 reserves 470 and 474, we do not (for now).
187             . IIF      GE, VA-500        VERR=1
188             . IIF      NE, VA&MASK       VERR=1
189             . IF       NE, VERR
190 . ERROR      ; Invalid vector address for this line
191             . ENDC
192             . ENDM      VECCHK
193
194 ; -----
195 ; SRCHK macro checks the validity of a receiver status
196 ; register address
197 ;
198             . MACRO     SRCHK   SR
199 SRERR=0
200             . IIF      LT, SR-160000     SRERR=1
201             . IIF      NE, SR&7         SRERR=1
202             . IF       NE, SRERR
203 . ERROR      ; Invalid status register address for this line
204             . ENDC
205             . 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
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      ; The DHDEF macro is used to declare the beginning
4      ; of a block of lines which are attached to a DH11
5      ; multiplexer. All line definition blocks up to
6      ; the next MUXEND macro call will be connected to
7      ; the DH11.
8      ; There are four arguments to DHDEF:
9      ; 1. The interrupt vector address of the mux receiver.
10     ; 2. The address of the mux control and status register.
11     ; 3. The interrupt vector address of the associated DM11.
12     ; 4. The CSR address of the associated DM11.
13     ;
14     . MACRO DHDEF AVEC, ACSR, ADMVEC, ADMADR
15     . IF NE CURMX
16     . ERROR 1; Missing MUXEND macro
17     MUXEND
18     . ENDC
19     LSTMX = LSTMX+2
20     CURMX = LSTMX
21     CURCDX = CDX$DH ;Lines within this block are connected to DH11
22     DHUSE = 1 ;Set flag saying DH11 support is needed
23     VECCHK AVEC, 7
24     SRCHK ACSR
25     . IF NE, ADMVEC
26     VECCHK ADMVEC, 3
27     . ENDC
28     . IF NE, ADMADR
29     SRCHK ADMADR
30     . ENDC
31     S =
32     = MXTYPE+CURMX ;Type of multiplexor
33     . WORD CDX$DH ;Type = DH11
34     = MH$SCR+CURMX ;Status and control register address
35     . WORD ACSR
36     = MH$RCR+CURMX ;Received character register
37     . WORD ACSR+2
38     = MH$LPR+CURMX ;Line parameter register
39     . WORD ACSR+4
40     = MH$CAR+CURMX ;Current address register
41     . WORD ACSR+6
42     = MH$BCR+CURMX ;Byte count register
43     . WORD ACSR+10
44     = MH$BAR+CURMX ;Buffer active register
45     . WORD ACSR+12
46     = MH$BRK+CURMX ;Break control register
47     . WORD ACSR+14
48     = MH$SSR+CURMX ;Silo status register
49     . WORD ACSR+16
50     = DM$CSR+CURMX ;DM11 Control Status register
51     . WORD ADMADR
52     = DM$LSR+CURMX ;DM11 Line status register
53     . WORD ADMADR+2
54     = MXVEC+CURMX ;DH11 Interrupt vector address
55     . WORD AVEC
56     = DM$VEC+CURMX ;DM11 Interrupt vector address
57     . WORD ADMVEC

```

58
59

= S
. ENDM DHDEF

```

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 DHVDEF macro is used to declare the beginning
; of a block of lines which are attached to a DHV11
; multiplexer. All line definition blocks up to
; the next MUXEND macro call will be connected to
; the DHV11.
; There are two arguments to DHVDEF:
; 1. The interrupt vector address of the mux receiver.
; 2. The address of the mux control and status register.
;
; .MACRO DHVDEF AVEC,ACSR
; .IF NE CURMX
; .ERROR 1; Missing MUXEND macro
; MUXEND
; ENDC
LSTMX = LSTMX+2
CURMX = LSTMX
CURCDX = CDX$VH ;Lines within this block connected to DHV11
DHUSE = 1 ;Set flag saying DHV11 support is needed
VECCHK AVEC,7
SRCHK ACSR
S =
; = MXTYPE+CURMX ;Type of multiplexor
; .WORD CDX$VH ;Type = DHV11
; = VH$CSR+CURMX ;Status and control register address
; .WORD ACSR
; = VH$DBR+CURMX ;Data buffer register
; .WORD ACSR+2
; = VH$LPR+CURMX ;Line parameter register
; .WORD ACSR+4
; = VH$LSR+CURMX ;Line Status Register
; .WORD ACSR+6
; = VH$LCR+CURMX ;Line Control Register
; .WORD ACSR+10
; = VH$BA1+CURMX ;Buffer Address register 1
; .WORD ACSR+12
; = VH$BA2+CURMX ;Buffer Address register 2
; .WORD ACSR+14
; = VH$BCR+CURMX ;Byte Count Register
; .WORD ACSR+16
; = MXVEC+CURMX ;Interrupt vector address
; .WORD AVEC
; = S
; .ENDM DHVDEF
;-----
; The DHUDEF macro is used to declare the beginning
; of a block of lines which are attached to a DHU11
; multiplexer. All line definition blocks up to
; the next MUXEND macro call will be connected to
; the DHU11.
; There are two arguments to DHUDEF:
; 1. The interrupt vector address of the mux receiver.
; 2. The address of the mux control and status register.
;
; .MACRO DHUDEF AVEC,ACSR

```

58
59

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

```

1      ;
2      ;-----
3      ; The MUXDEF macro is used to declare the beginning
4      ; of a block of lines which are attached to a DZ11
5      ; multiplexer. All line definition blocks up to
6      ; the next MUXEND macro call will be connected to
7      ; the DZ11.
8      ; There are two arguments to MUXDEF:
9      ; 1. The interrupt vector address of the mux receiver.
10     ; 2. The address of the mux control and status register.
11     ;
12     .MACRO MUXDEF AVEC,ACSR
13     .IF NE CURMX
14     .ERROR 1; Missing MUXEND macro
15     MUXEND
16     .ENDC
17     LSTMX = LSTMX+2
18     CURMX = LSTMX
19     CURCDX = CDX#DZ ;Lines within this block are connected to DZ11
20     VECCHK AVEC,7
21     SRCHK ACSR
22     S =
23     = MXTYPE+CURMX ;Multiplexor type
24     .WORD CDX#DZ ;Type = DZ11
25     = MXCSR+CURMX
26     .WORD ACSR
27     = MXLPR+CURMX
28     .WORD ACSR+2
29     = MXTCR+CURMX
30     .WORD ACSR+4
31     = MXDTR+CURMX
32     .WORD ACSR+5
33     = MXTBUF+CURMX
34     .WORD ACSR+6
35     = MXCAR+CURMX
36     .WORD ACSR+7
37     = MXVEC+CURMX
38     .WORD AVEC
39     = S
40     .ENDM MUXDEF
41     ;
42     ; Alternate name for MUXDEF macro
43     ;
44     .MACRO DZDEF AVEC,ACSR
45     MUXDEF AVEC,ACSR
46     .ENDM DZDEF
47     ;
48     ;-----
49     ; The MUXEND macro is called to declare the end of
50     ; a set of lines connected to a DZ11 or DH11.
51     ;
52     .MACRO MUXEND
53     .IF EQ CURMX
54     .ERROR 6; Missing earlier MUXDEF
55     .ENDC
56     CURMX = 0
57     CURCDX = CDX#DL ;Following lines are connected to DL11's

```

58

.ENDM MUXEND

```

1      ; -----
2      ; The SPOOL macro is used to declare those devices which
3      ; are to be spooled by TSX-Plus (such as line printers).
4      ; There are seven arguments to spool:
5      ; 1) Number of devices to be spooled (may be zero)
6      ; 2) Number of spool files allowed to be open.
7      ; 3) Number of buffers for spooler to use.
8      ; 4) Number of blocks in spool disk file.
9      ; 5) List of 3 character names of devices to be spooled.
10     ; 6) 0 for 'nohold' mode, 1 for 'hold' mode.
11     ; 7) # of blocks which will be remembered for back up.
12     ;
13     .MACRO SPOOL SND, SNF, SNB, SNDB, SNAM, SHLD, SNBU
14     ; Define number of spooled devices
15     SPLND = SND
16     SNBUX = SNBU
17     .IF EQ, SNBU
18     SNBUX = 1
19     .ENDC
20     PVSPBL = SNBUX+10. ;# PRIVATE BLOCKS PER DEV
21     SNDBX = SNDB ;TOTAL # OF SPOOL BLOCKS
22     .IF LT <SNDB-<SND*PVSPBL>-2>
23     SNDBX = <SND*PVSPBL>+2
24     .ENDC
25     .IF GT, SPLND
26     ;**
27     ;** Assemble this code if there are spooled devices.
28     ;**
29     SPLNF = SNF ;DEFINE # OF SPOOL FILES
30     SPLNB = SNB ;DEFINE # OF SPOOL BUFFERS
31     NESB: .WORD SPLNB
32     ; DEFINE SPOOL BUFFERS
33     .IF EQ, SPLNB ;THERE MUST BE AT LEAST 1 BUFFER
34     .ERROR ;There must be at least 1 buffer for spooler
35     SPLNB = 1 ;FORCE 1 BUFFER
36     .ENDC
37     SPLBHD: .WORD 0 ;HEAD OF FREE BUFFER CHAIN
38     .IF EQ, SPLNF ;Make sure we have at least 1 file
39     .ERROR ;There must be at least 1 spool file
40     SPLNF = 1 ;FORCE 1 FILE
41     .ENDC
42     ;
43     ; Define spool device control blocks (SDCB)
44     ;
45     SDCB:
46     .REPT SPLND
47     .WORD 0,0,0,0,0 ;SDCHAN
48     .WORD 0,0,0,0,0,0
49     .WORD 0,0,0,0
50     ; INITIAL FORM NAME
51     .ASCII /STD / ;SDFORM
52     .WORD 0,0 ;SDANAM
53     ; GEN INIT FLAGS
54     .IF NE, SHLD
55     .WORD SD#HLD ;SDFLAG
56     .IFF
57     .WORD 0 ;SDFLAG

```

```

58          . ENDC
59          . WORD 0 ; SDSKIP
60          . WORD PVSPBL ; SDFRBL
61 ; GEN BACKUP CELLS
62          . REPT SNBUX
63          . WORD 0
64          . ENDR
65          . WORD 0 ; SDBULS = END OF SDBU
66          . ENDR
67 SDCBND: ; END OF SDCB AREA
68 ; DEFINE SIZE OF SDCB
69 SDCBSZ = 48. + <2*SNBUX>
70 SDBULS = SDCBSZ-4.
71 ;
72 ; Define table of device names.
73 ;
74 SPLDEV: . RAD50 / 'SNAM' / ; DEFINE TABLE OF NAMES
75          . EVEN
76 SPLDVN: ; END OF TABLE
77          . IF NE, <<SPLDVN-SPLDEV>-<2*SND>>
78          . ERROR ; Number of spooled devices not equal to number of names
79          . ENDC
80 SPLANM: . ASCII / 'SNAM' /
81          . EVEN
82 ; Reserve space for spool file channel block
83 SPLCHN: . BLKW 5
84 ;
85          . IFF
86 ;**
87 ;** This code is assembled if there are no spooled devices.
88 ;**
89 SPLND = 0
90 SPLNF = 0
91 SPLNB = 0
92 SDCBSZ = 1
93 SDBULS = 1
94 ;
95 SPLDEV:
96 SPLDVN:
97 SPLANM:
98 SPLBHD:
99 SPLCHN:
100 SDCB:
101 SDCBND:
102 NESB:
103          . WORD 0 ; SAY ALL LISTS ARE EMPTY
104 ;
105          . ENDC
106          . 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 003102 075250 100020 075150 SWDBLK: .RAD50 /SY TSXSWPTSX/
10 003110 100020
11          ;
12          ; Spool file device-file specification (do not place on VM).
13          ;
14 003112 075250 100020 074514 SPLBLK: .RAD50 /SY TSXSPLTSX/
15 003120 100020
16          ;
17          ; PLAS region swap file specification (do not place on VM).
18          ;
19 003122 075250 100020 071576 RSFBLK: .RAD50 /SY TSXRSFTSX/
20 003130 100020
21          ;
22          ; File spec for file used to hold user defined command definitions (UCL)
23          ;
24 003132 075250 100020 101704 UCLDAT: .RAD50 /SY TSXUCLTSX/
25 003140 100020
26          ;
27          ; File spec for temp file used while processing IND command files
28          ;
29 003142 075250 100020 035164 INDFIL: .RAD50 /SY TSXINDTSX/
30 003150 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 = 56. ;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      000000      SWPSLT =      0.      ;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      000000      SEGBLK =      0.      ;# 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     000370      MEMSIZ =      248.      ;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 000001 UXIFLG = 1 ;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.
158 ; Specify 177564 to dump on the console terminal.
159 ; Specify 177514 to dump to line printer connected to standard parallel port.
160 ;
161 177560 DMPTCR = 177560 ;Transmitter control reg for dump device
162 ;
163 ; Set DMPKTP to 1 if you want a system crash to occur any time a trap
164 ; occurs within the system. Set it to 0 (zero) if you want recoverable
165 ; traps within the system to abort the job but continue execution of the
166 ; system.

```

```

167      ;
168      000000      DMPKTP =      0      ;1==>Always crash on traps within system
169      ;
170      ; The IOABT parameter controls the action taken by TSX-Plus when
171      ; a job terminates execution. If IOABT=0, TSX-Plus will wait for
172      ; all outstanding I/O pending for the job to complete before the job
173      ; is actually terminated. If IOABT=1, TSX-Plus will call the handler
174      ; abort entry point for all outstanding I/O pending for the job.
175      ; Note, the "SET IO [NO] ABORT" keyboard command may be used to
176      ; change the value of this parameter.
177      ;
178      000001      IOABT =      1      ;0==>I/O rundown, 1==>I/O abort
179      ;
180      ; U$CL is a flag that controls whether the User Command Linkage is to
181      ; be used to allow users to define their own commands.
182      ; If U$CL is non-zero the UCL facility is enabled and users may define
183      ; their own system commands. If U$CL is zero, user defined commands
184      ; will not be supported by the system. Note: if the UCL facility is
185      ; enabled, the TSXUCL.SAV file must be placed on the system disk.
186      ;
187      000001      U$CL =      1      ;0==>No UCL program, 1==>UCL program
188      ;
189      ; Number of user-defined commands that can be stored by TSXUCL
190      ; for each job. (The number of blocks required in the SY:TSXUCL.DAT file
191      ; is approximately equal to the number of commands per job times the
192      ; total number of time-sharing lines divided by 5).
193      ;
194      000005      UCLMNC =      5.      ;Maximum user-defined commands per job
195      ;
196      ; The UCLORD parameter selects the default call order for checking
197      ; to see if a command is a user-defined command.
198      ; FIRST ==> Check for user-defined commands before system commands.
199      ; MIDDLE ==> Check after system commands but before command files.
200      ; LAST ==> Check after system commands and command files.
201      ;
202      ; Note that the SET UCL FIRST/LAST keyboard command can be used to
203      ; alter this order on a line-by-line basis.
204      ;
205      000002      UCLORD =      MIDDLE ;Select FIRST / MIDDLE / LAST
206      ;
207      ; The LDSYS flag controls whether the standard system support for
208      ; logical disks (LD) is to be provided.
209      ; If LDSYS is set to 1, system support for logical disks is included.
210      ; If LDSYS is set to 0, system support for logical disks is excluded.
211      ;
212      000001      LDSYS =      1      ;1==>Include LD support, 0==>Exclude LD.
213      ;
214      ; The SLEDIT flag controls whether the Single Line Editor (SL) facility
215      ; is to be made available to the system.
216      ; If SLEDIT is set to 1, Single Line Editor support is included.
217      ; If SLEDIT is set to 0, Single Line Editor support is omitted.
218      ; Single Line Editor support adds approximately 2Kb to the size of the
219      ; mapped portion of the system.
220      ;
221      000001      SLEDIT =      1      ;1==>Include SL support, 0==>Exclude SL
222      ;
223      ; The KEYMAX parameter specifies the number of user-defined keys supported

```

```

224 ; by the single line editor. The DEFINE/KEY command is used to associate
225 ; a user-specified text string with a function key. The maximum number
226 ; of such key definitions that may be in effect at one time for each user
227 ; is controlled by the KEYMAX parameter.
228 ; The maximum supported value for KEYMAX is 60.
229 ;
230 000007 KEYMAX = 7 ;Maximum number of user-defined keys for SL
231 ;
232 ; The MAXWIN parameter specifies the maximum number of terminal display
233 ; windows that may be in use by all jobs on the system.
234 ; If MAXWIN is set to 0 (zero), the display window feature is not included
235 ; in the system. Display windows are useful if you frequently utilize
236 ; subprocesses in that they preserve the screen context when you switch
237 ; between processes.
238 ;
239 000012 MAXWIN = 10 ;Total number of display windows for all jobs
240 ;
241 ; Set DBGFLG to 1 to cause the TSX-Plus program debugging facility
242 ; to be included with the system.
243 ; Set DBGFLG to 0 if the debugging facility is not wanted.
244 ;
245 000000 DBGFLG = 0 ;1==>Include debugger; 0==>Exclude debugger
246 ;
247 ; Number of slots in INSTALL table to reserve for user programs.
248 ;
249 000004 NUIP = 4 ;Number of INSTALL slots for user programs
250 ;
251 ; The following time-slice values are used to schedule jobs for execution.
252 ; Each time value must be specified in 0.1 second units.
253 ;
254 ; QUANO -- Time slice for round-robin scheduling of high-priority
255 ; real-time jobs. That is, jobs with execution priorities
256 ; greater than or equal to PRIHI.
257 ;
258 000002 QUANO = 2 ;Time slice for real-time jobs
259 ;
260 ; QUAN1 -- Time that jobs will remain in a high-priority state after
261 ; they receive an activation character from the terminal.
262 ; A job is classified as "interactive" from the time when an
263 ; activation character is received until the job consumes
264 ; QUAN1 units of time, then the job is classified as "compute
265 ; bound".
266 ;
267 000024 QUAN1 = 20 ;High-priority time for interactive jobs
268 ;
269 ; QUAN1A -- Time that jobs will remain in a high-priority state after
270 ; they are activated because of I/O completion or they are
271 ; restarted following other wait states.
272 ;
273 000002 QUAN1A = 2 ;High-priority time for wait-reactivation
274 ;
275 ; QUAN1B -- Time slice used to switch between "interactive" jobs.
276 ;
277 000002 QUAN1B = 2 ;Time slice for "interactive" jobs.
278 ;
279 ; QUAN1C -- Time job will be allowed to stay in highest execution state
280 ; after receipt of a character from the terminal.

```

```

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

```

```

338      000062      PRIDEF =      50.      ;Default job priority
339      ;
340      ; PRIVIR -- Amount by which a job's execution priority is reduced
341      ;                when the job is disconnected from the terminal by switching
342      ;                to a subprocess. Note: this only applies to jobs with
343      ;                base priorities in the range (PRILOW+1) to (PRIHI-1).
344      ;
345      000012      PRIVIR =      10.      ;Disconnect job priority reduction
346      ;
347      ; Maximum number of subprocesses per primary process.
348      ;
349      000002      MAXSEC =      2.      ;Max subprocesses per user
350      ;
351      ; Maximum file size (# blocks) that will be returned in response to
352      ; a .ENTER request that specifies a file size of 0 blocks.
353      ;
354      001750      MAXFIL =     1000.      ;Max # blocks for default allocation
355      ;
356      ; Number of 512 byte blocks to hold in memory in a generalized data cache.
357      ; If the CACHE parameter is set to 0 (zero), data caching is not performed.
358      ; Note: The data caching facility adds approximately 2000 bytes to the
359      ; size of the unmapped portion of the system and 528*CACHE bytes to
360      ; the mapped portion of the system.
361      ; The maximum number of blocks that may be held in the cache is 4095. (2MB)
362      ;
363      000000      CACHE =      0.      ;Number of blocks in data cache
364      ;
365      ; The following parameters relate to the cache of file directory entries
366      ; maintained by TSX-Plus. This cache is used to reduce the number of disk
367      ; accesses required to do lookups on frequently accessed files.
368      ; The system disk (SY:) is automatically cached.
369      ; Other devices are only cached if they are introduced to the system
370      ; by use of the MOUNT command.
371      ;
372      ; Maximum number of units that may be cached.
373      ; This includes all logical disks (LD) and all physical disks for which
374      ; directory caching is enabled by use of the MOUNT command.
375      ; (Space required is 18 bytes per unit).
376      ;
377      000012      MAXCSH =      10.      ;Max # device units whose directories to cache
378      ;
379      ; Maximum number of file entries to be held in directory cache.
380      ; (Space required is 18 bytes per entry)
381      ;
382      000050      NMFCSH =      40.      ;Max # file entries to be cached
383      ;
384      ; Maximum number of device units that can be allocated to jobs for exclusive
385      ; use by use of the ALLOCATE command.
386      ;
387      000005      MAXALC =      5.      ;Max # units that can be allocated
388      ;
389      ; Maximum number of simultaneous requests by jobs to monitor other jobs.
390      ;
391      000005      MAXMON =      5.      ;Max # job monitoring requests
392      ;
393      ; The system password is a global password which must be entered
394      ; when a line is initiated before the normal logon sequence begins.

```

```

395 ; The use of a system password is optional and may be enabled on a
396 ; line-by-line basis by specifying the $SYSPS flag with the
397 ; FLAGS macro within the line definition blocks for the lines
398 ; for which the password will be required. If a system password is
399 ; required for a line, an exclamation point prompt is printed as the
400 ; first thing when the line is initiated. The idea is to force the
401 ; calling person to provide a password before printing the normal
402 ; logon greeting which identifies the nature and identity of the site.
403 ;
404 003152 ;          SYSPS <TSX> ;System password for all lines with $SYSPS
405 ;
406 ; Amount of time that carrier signal must be lost on dial-up
407 ; lines before we assume the connection has been broken.
408 ; This value is also used to time-out lines which ring and
409 ; do not raise carrier.
410 ; Specify in 0.5 second units.
411 ;
412 000036 TIMEOUT = 30. ;Time allowed for lost carrier
413 ;
414 ; Amount of time that a user may remain connected to a dial-up line
415 ; after logging off before Data Terminal Ready (DTR) will be
416 ; dropped causing the phone to hang up.
417 ; Specify in 0.5 second units.
418 ;
419 000074 OFFTIM = 60. ;Time allowed for job to be logged off
420 ;
421 ; Modem lines ($PHONE in the LINDEF FLAGS macro) are normally
422 ; treated as phone lines if the DCD signal (carrier) is present
423 ; when the lines are started and optionally treated as local lines
424 ; if the signal is not present. The OFFTIM and TIMEOUT parameters
425 ; are only effective if the line is recognized as a phone line when
426 ; started. Set PHONE to 0 to allow lines with the $PHONE
427 ; flag to optionally support local lines. Set PHONE to 1 to
428 ; force the OFFTIM and TIMEOUT parameters to always take effect for
429 ; lines with the $PHONE flag.
430 ;
431 000000 PHONE = 0. ;$PHONE lines may be local if carrier absent
432 ;
433 ; Define Lead-in character that tells TSX-Plus that a special
434 ; terminal control sequence is coming from the program.
435 ;
436 000035 TSLICH = 035 ;Octal 35 = decimal 29.
437 ;
438 ; Define the keyboard control character that will be used to
439 ; switch to a subprocess.
440 ; (Specify the octal value of the ASCII control character)
441 ;
442 000027 VLSWCH = 027 ;Octal 27 = control-W
443 ;
444 ; Define keyboard control character used to cause the current screen
445 ; window contents to be printed.
446 ; (Specify the octal value of the ASCII control character)
447 ;
448 000002 PWCH = 002 ;Octal 02 = control-B
449 ;
450 ; Define keyboard control character that is used to terminate
451 ; a cross-connection between a time-sharing line and a CL line.

```

```

452 ; (Specify the octal value of the ASCII control character)
453 ;
454 000034 CCXTRM = 034 ;Octal 34 = control-\ (control backslash)
455 ;
456 ; Define keyboard control character that is used to signal
457 ; special control functions for a time-sharing line cross-connected
458 ; to a CL line.
459 ; (Specify the octal value of the ASCII control character)
460 ;
461 000001 CCXCTL = 001 ;Octal 001 = control-A
462 ;
463 ; Define the version number to be associated with the CL handler when
464 ; being used with VTCOM. If CLVRSN is defined as 0 then an appropriate
465 ; value will be selected via an internal table. Zero is the suggested
466 ; setting.
467 ;
468 000000 CLVRSN = 0. ;CL version number
469 ;
470 ; Define maximum number of user defined activation characters
471 ; that each line may define during execution.
472 ;
473 000020 MXSPAC = 16. ;Max # user defined activation chars per job
474 ;
475 ; Define maximum number of characters that can be translated by
476 ; the terminal handler. This translation consists of replacing
477 ; a received character by a substitution character on input and replacing
478 ; the substitution character by the original character on output.
479 ; This parameter must be non-zero to use the SET TT TRANSLATE=( ) command.
480 ;
481 000004 MXTTCT = 4. ;Max # chars that terminal handler can translate
482 ;
483 ; Select default system editor.
484 ; The choices are
485 ; EDIT
486 ; TECO
487 ; KED
488 ; K52
489 ;
490 000003 EDITOR = KED ;Default system editor
491 ;
492 ; Select system default implicit or explicit wildcards for CCL commands.
493 ; If WILDFL = 0 then explicit wildcards are selected.
494 ; If WILDFL = 1 then implicit wildcards are selected.
495 ;
496 000001 WILDFL = 1 ;1==>Implicit wildcard, 0==>Explicit wildcard
497 ;
498 -----
499 ; The DEVDEF macro must be used to define the names and characteristics
500 ; of all devices which are to be available to TSX-Plus users.
501 ; The form of a device definition is:
502 ;
503 ; DEVDEF <device>[,option,...,option]
504 ;
505 ; For each device to be available to the system an entry must be made
506 ; using the DEVDEF macro. This macro requires at least one argument
507 ; but may have several optional arguments as described below:
508 ;

```

```

509 ; 1. The first parameter is the two character device name enclosed
510 ; in angle brackets.
511 ; 2. The optional parameters specify the device characteristics.
512 ; There are nine allowable device attributes which may be
513 ; specified in any order. They are as follows:
514 ;
515 ; DMA Device performs Direct Memory Access (DMA).
516 ; MAPIO Perform I/O mapping (18-bit controllers on 22-bit QBUS).
517 ; EVNBUF Require even byte buffer address for I/O transfers.
518 ; NOCACHE Do not use generalized data cache for this device.
519 ; NOMOUNT Do not allow mounts (i.e., use directory cache) for
520 ; this device.
521 ; REQALC Require device allocation before use.
522 ; MAPH Load the device handler outside the low memory 40K
523 ; byte region and into a mapped handler region.
524 ; NOMAPH Do not load the handler into a mapped handler region
525 ; instead load it into the low memory 40k byte region.
526 ; HANBUF Handler contains an internal I/O buffer.
527 ;
528 ; For standard device drivers, it is important to choose MAPIO when
529 ; 18-bit controllers or handlers will be used on a 22-bit LSI system.
530 ; It is not necessary to specify other device attributes for standard
531 ; TSX-Plus supplied device drivers since TSX-Plus will automatically
532 ; make default selections.
533 ;
534 ; *****
535 ; ** When performing a TSX-Plus **
536 ; ** system generation, remove the **
537 ; ** devices in this list which are **
538 ; ** not present on your system, **
539 ; ** and include those which are. **
540 ; *****
541 ;
542 003200 DEVBEG ;Beginning of device definitions
543 003200 DEVDEF <DL>
544 003200 DEVDEF <DM>
545 003200 DEVDEF <DU>,NOSET
546 003200 DEVDEF <RK>,MAPIO
547 003200 DEVDEF <DY>,MAPIO
548 003200 DEVDEF <DX>
549 003200 DEVDEF <LP>
550 003200 DEVDEF <NL>
551 003200 DEVEND ;End of device definitions
552 ;
553 ; -----
554 ; Parameters related to system I/O buffers used when DMA devices
555 ; with 18-bit controllers are used on Q-bus systems with
556 ; 22-bit addressing (e.g., 11/23-Plus and 11/73).
557 ;
558 ; Number of system buffers allocated for I/O buffering.
559 ; (The recommended number is one per active device that requires buffering.)
560 ;
561 000001 MIONBF = 1 ;Number of system I/O buffers
562 ;
563 ; Size of each system I/O buffer, in units of 512 bytes.
564 ; The maximum allowed value for this parameter is 15.
565 ;

```

```

566      000010      MIOBSZ =      8.      ; I/O buffer size in units of 512 bytes
567
568      ; -----
569      ; Some device handlers allocate extended memory (PLAS) regions for
570      ; their use. For example, the DU and MU handlers each require one
571      ; PLAS region. If you are using any other handlers which require
572      ; extended memory regions, include the number of regions required.
573      ;
574      000004      DEVXMR =      4.      ; Number of XM regions for device handlers
575
576      ; -----
577      ; Define those devices which are to be spooled by TSX-Plus
578      ; (such as line printers).
579      ; There are seven arguments to the SPOOL macro:
580      ; 1. Number of devices to be spooled (may be zero).
581      ; 2. Number of spool files which may be open by all users.
582      ; 3. Number of spool buffers (512. bytes each).
583      ; 4. Number of blocks in spool disk file.
584      ; 5. List of 3 character names of devices to be spooled.
585      ; 6. Specify 0 if spool files are to be eligible to be
586      ; started as soon as they are created,
587      ; specify 1 if they are to be held until the channel
588      ; is closed. Note: The "SPOOL xx,[NO]HOLD" keyboard
589      ; command can override this parameter.
590      ; 7. Number of blocks which are to be backed up
591      ; when the "SPOOL xx,BACK" command is given.
592      ;
593      ; Note: The SPOOL macro must be present even if
594      ; there are no spooled devices. However, if the first
595      ; argument (number of spooled devices) is zero, no spool
596      ; tables are generated and arguments 2-7 are ignored.
597      ;
598      003200      SPOOL 1,20.,2,500.,<LP >,0,10.
599
600      ; -----
601      ; Define parameters pertaining to record (block) locking
602      ; for shared files. If the shared file block locking
603      ; facility is not wanted, set all of these parameters to
604      ; 0 (zero).
605      ;
606      ; Maximum number of shared files which may be open
607      ; simultaneously. Note that several users accessing the same
608      ; file count as 1.
609      ;
610      000036      MAXSF =      30.      ; Max number of shared files
611      ;
612      ; Maximum number of I/O channels which all users may
613      ; simultaneously have open to shared files.
614      ; Note, this is the total number for all users not
615      ; for each user.
616      ;
617      000036      MAXSFC =      30.      ; Max # shared file channels
618      ;
619      ; Maximum number of blocks which may be simultaneously
620      ; held locked by any channel. That is, max blocks
621      ; locked per channel.
622      ;

```

```

623      000003      MXLBLK =      3.      ;Max blocks locked per channel
624      ;
625      ; Number of 512-byte blocks to be held in the in-memory data
626      ; cache for shared files.
627      ; (Note that the MAXSF, MAXSFC, and MXLBLK parameters must be
628      ; non-zero to enable shared file data caching.)
629      ;
630      000000      NUMDC  =      0.      ;Number of blocks in shared file data cache
631      ;
632      ;-----
633      ; Define parameters pertaining to the inter-program
634      ; message communication feature.  If this feature is
635      ; not wanted, set all four parameters to 0 (zero).
636      ;
637      ; Maximum number of message communication channels
638      ; which may be simultaneously in use.
639      ;
640      000003      MAXMC   =      3.      ;Max message channels
641      ;
642      ; Maximum message length (bytes).
643      ;
644      000310      MSCHRS =      200.    ;Max message length (bytes)
645      ;
646      ; Maximum number of messages which may be held in queue.
647      ;
648      000003      MAXMSG  =      3.      ;Max queued messages
649      ;
650      ; Maximum number of requests for messages that may be held in queue
651      ;
652      000012      MAXMRB  =      10.    ;Max # pending message requests
653      ;
654      ;-----
655      ; The RTVECT parameter specifies the number of real-time interrupt vectors
656      ; that can be connected to TSX-Plus jobs.  Set RTVECT to the maximum number
657      ; of interrupt vectors that all running real-time programs may be connected
658      ; to at the same time.
659      ; (Note: The basic real-time support facility is now a standard part of
660      ; TSX-Plus and it is no longer necessary to set RTVECT to 1 to include
661      ; real-time facilities such as locking a job in memory or accessing the
662      ; I/O page.  It is also no longer necessary to set RTVECT to 1 to allow
663      ; use of the SYSDON program.  RTVECT should be set to 0 (zero) unless some
664      ; real-time interrupts are going to be connected to TSX-Plus jobs.)
665      ;
666      000000      RTVECT  =      0.      ;Max # interrupt vectors that may be connected
667      ;
668      ;-----
669      ; Define the size of the table within TSX-Plus used to hold information
670      ; when the performance monitoring feature is being used.
671      ; Each word in this table corresponds to one cell in the histogram.
672      ; Specify the size as number of bytes for the table.
673      ; (Note: The maximum allowed size is 8192 bytes)
674      ;
675      000000      PMSIZE  =      0.      ;Size of performance monitor table (bytes)
676      ;
677      ;-----
678      ; Use the RTDEF macro at this point to specify information about
679      ; any shared run-time systems to be loaded when TSX-Plus is started.

```

```

680 ;
681 ; The form of the RTDEF macro is
682 ; RTDEF <name>,r-flag,skip-count
683 ;
684 ; Where
685 ; - Name is the 12 character name of the file containing the run-time system
686 ; which must be specified in the form DevFilnamExt -- that is, three
687 ; character device name, six character file name and three character
688 ; extension.
689 ; - R-flag is either R if user programs are to have read-only access to
690 ; the run-time system, or RW if read-write access is to be granted.
691 ; - Skip-count is the number of blocks to be skipped over at the front
692 ; of the file when loading it.
693 ;
694 ; Example:
695 ; RTDEF <SY CBRO63SHR>,R,1. ;COBOL-Plus shared run-time
696 ; RTDEF <SY DBLSHRRTS>,R,1. ;DBL shared run-time
697 ; RTDEF <SY DB4RTSSHR>,R,0. ;DBL V4 shared run-time
698 ;

```

```

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 ; 000144 DINSPC = 100. ;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 ; 000040 NCSILO = 32. ;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 ; 000001 CLXTRA = 1. ;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      001000      $TAB   =      1000   ;ON==>Do not simulate tabs (Terminal handles tab char)
71      000400      $FORM  =      400    ;ON==>Do not simulate form-feeds (Terminal handles FF)
72      000200      $AUTO  =      200    ;ON==>Do autobaud speed selection for line
73      000100      $PAGE  =      100    ;ON==>Enable ctrl-S/ctrl-Q input processing
74      000040      $LC    =      40     ;ON==>Enable lower-case input
75      000020      $NOSUB =      20     ;ON==>Disallow use of subprocesses
76      000010      $DEFER =      10     ;ON==>Do deferred character echoing (recommended)
77      000004      $QTSET =      4      ;ON==>Set tt quiet (Don't list command files)
78      000002      $SYSPS =      2      ;ON==>Require system password before logon
79      000001      $PHONE =      1      ;ON==>Dial-up, modem connected line
80      ;
81      ;  Default line flags that will be used for each line that does
82      ;  not explicitly specify flags using a FLAGS macro.
83      ;
84      040150      NRMFLG =      $ECHO!$DEFER!$PAGE!$LC
85      ;
86      ;-----
87      ;  Terminal type names that are legal to used with the TRMTYP macro
88      ;  within a line definition block to define the terminal type.
89      ;
90      ;  VT100 ==> DEC VT100
91      ;  VT200 ==> DEC VT200 with 7 bit control codes
92      ;  VT52  ==> DEC VT52
93      ;  LA36  ==> DEC LA36
94      ;  LA120 ==> DEC LA120
95      ;  HAZEL ==> Hazeltine brand terminals
96      ;  ADM3A ==> Lear Siegler ADM3A
97      ;  DIABLO==> Diablo brand terminals (with X-ON/X-OFF protocol)
98      ;  QUME  ==> Qume brand terminals (with X-ON/X-OFF protocol)
99      ;

```

```

1          ; -----
2          ; Line definitions
3          ;
4          ; The TBLDEF macro call requires four arguments:
5          ; 1. The number of real (physical) time-sharing lines on machine.
6          ; 2. The number of subprocess jobs.
7          ; 3. The number of detached jobs.
8          ; 4. The number of dedicated CL lines.
9          ;
10         003330          TBLDEF 3,2,2,0          ;# Real, # Subprocess, # Detached, # CL lines
11
12         ;
13         ; Define primary (real) time-sharing lines
14         ;
15         ;
16         ; #1 time-sharing line
17         006220          LINDEF 60,177560,OPER          ;USE CONSOLE TERMINAL AS T/S TERM
18         006220          NAME <Console>
19         ;              CMDFIL LINE1.TSX
20         ;              TRMTYP VT100
21         006220          FLAGS NRMFLG!$START
22         006220          LINEND
23
24         ; #2 time-sharing line
25         006234          LINDEF 310,176510
26         ;              CMDFIL LINE2.TSX
27         ;              TRMTYP LA120
28         006234          FLAGS NRMFLG
29         006234          LINEND
30
31         ; #3 time-sharing line
32         006250          LINDEF 320,176520
33         ;              CMDFIL LINE3.TSX
34         ;              TRMTYP VT52
35         006250          FLAGS NRMFLG
36         006250          LINEND
37
38         ; ; The following section is an example of line definitions for a
39         ; ; DHV11 type multiplexer.
40         ;
41         ;              DHVDEF 370,160020          ;DHV11 MUX VECTOR & RSR ADDRESS
42         ;
43         ; ; Mux line # 0 - first line on DHV
44         ;              LINDEF 0
45         ;              CMDFIL LINE2.TSX
46         ;              FLAGS NRMFLG!$AUTO
47         ;              LINEND
48         ;
49         ; ; Mux line # 7 - last line on DHV
50         ;              LINDEF 7
51         ;              CMDFIL LINE2.TSX
52         ;              FLAGS NRMFLG!$AUTO
53         ;              LINEND
54         ;
55         ;              MUXEND          ;END OF DHV11 MUX LINES USED
56         ;
57         ; ; The following section is an example of line definitions for a

```

```

58      ;; DZV11 type multiplexer.
59      ;
60      ;       DZDEF      360,160010           ; DZV11 MUX VECTOR & RSR ADDRESS
61      ;
62      ;; Mux line # 0 - first line on DZ
63      ;       LINDEF    0
64      ;       TRMTYP    VT100
65      ;       SPEED     S9600
66      ;       CMDFIL    LINE2.TSX
67      ;       LINEND
68      ;
69      ;; Mux line # 3 - last line on DZ
70      ;       LINDEF    3
71      ;       TRMTYP    LA120
72      ;       SPEED     S1200
73      ;       CMDFIL    LINE2.TSX
74      ;       FLAGS     NRMFLG!$FORM
75      ;       LINEND
76      ;
77      ;       MUXEND           ;End of DZ11 lines
78      ;
79      ; Use the "DETACH" macro here to declare any start-up command
80      ; files and associated parameters (up to 80 characters) to be
81      ; run as detached jobs:
82      ;
83      ;       DETACH <SY:EXAMPL.TSX PARM1 PARM2> ;Detached job with parameters
84      ;       DETACH <SY:DETACH.TSX>           ;Start-up detached job
85      ;       DETACH <SY:WINPRT.TSX>          ;Start window-print detached job
86      ;
87      ;=====
88      ; END OF SECTION OF TSGEN TO BE ALTERED BY USER
89      ;=====

```

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

```

000016

000014

000004

000002

```

58
59 ; Define tables for subprocesses (if any)
60 ;
61 ; .IF NE NSL
62 000002 .REPT NSL
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 006530 . CSECT TSGEN
72 ;
73 ; Define mux interrupt entry vectors
74 ;
75 ; Input interrupt entry points
76 000000 .REPT LSTMX/2
77 INCB MUXNUM
78 . ENDR
79 006530 000167 000000G INMXV: JMP ININT
80 ; Output interrupt entry points (set up by TSINIT)
81 006534 OTMXV: .BLKW 3*<LSTMX/2>
82 ;
83 ; Generate tables for Unibus map registers
84 ;
85 ; .GLOBL UMRBAS, UMREND, UMRWHD
86 006534 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 000005 CURUMR = 5 ; Map regs 0-4 always mapped by init code.
96 006536 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 006536 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      ALCTBL:                                ; Base of device allocation table
122      . REPT   MAXALC
123      . WORD   0                               ; AD$DVU
124      . BYTE   0                               ; AD$JOB
125      . BYTE   0                               ; AD$FLG
126      . ENDR
127      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 006562  VCBBAS:
8          . REPT  RTVECT
9          JSR    R2, @#RTINT
10         . WORD  0, 0, 0          ; Must match size defined in TSDEFS
11         . ENDR
12 006562  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 000000  TSDHIO =      0
38 006562  DHTIMR:
39 006562  DHSTRT:
40 006562  DHSTOP:
41 006562  DHXON:
42 006562  DHXOFF:
43 006562  DHOINT:
44 006562  VHSTRT:
45 006562  VHSTOP:
46 006562  VHXON:
47 006562  VHXOFF:
48 006562  VHOINT:
49 006562  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$COD
66           ;
67           . IF     NE, CLVRSN           ; Test to see if CLVRSN is defined as 0
68           . IF     LE, <CLVRSN-14.> ; or greater than 14.
69           . ERROR  0; Minimum CL version number is 15. ; If not report an error
70           CLVRSN = 15.                 ; and set to a reasonable number
71           . ENDC                          ; End conditional LE, <CLVRSN-14.>
72           . ENDC                          ; End conditional NE, CLVRSN
73           ;
74           ; Define table that tells which line is associated with each CL unit.
75           ; Note, this table is always generated. Even if there are no CL lines
76           ; genned in.
77           ;
78           006564 000000 000000 000000 CL$LIX: . WORD  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ; Line index for each CL unit
79           006572 000000 000000 000000
80           006600 000000 000000 000000
81           006606 000000 000000 000000
82           006614 000000 000000 000000
83           006622 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 006666          CLTABL  CL$EPS          ;Pointer to end-of-file string buffer
111 006670          CLTABL  CL$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 006672 000031  NMFREQ: . WORD   NUMIOQ-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 000120
171
172 000000
173 000000 123456
174
176
177
178
179
180
181 000000
182 000050 000050
183 000050 000000G
184 000014 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: 137
 Work file writes: 127
 Size of work file: 21456 Words (84 Pages)
 Size of core pool: 17920 Words (70 Pages)
 Operating system: RT-11

Elapsed time: 00:01:39.44
 DK: TSGEN, LP: TSGEN=DK: TSGEN. MAC/C/N: SYM

NSPLFL	1-60	4-86#						
NUCHN	1-69	2-12#	2-83	2-84	2-85	2-86	2-87	2-88
NUIP	4-36	16-249#						
NUMCCB	1-107	2-18#						
NUMCDB	1-81	4-74#						
NUMDC	4-16	4-73	16-630#					
NUMDCD	1-81	4-73#						
NUMDEV	1-101	3-8	4-236#					
NUMFRK	1-57	2-6#						
NUMIQG	1-102	2-16#	22-13	22-13#	22-156			
NUMON	1-93	4-198#						
NUMRDB	1-84	2-91#	21-25	21-25	21-25#	21-25#		
NUMSYG	1-102	2-17#	22-14	22-14#	22-156			
NXIVMH	1-57	2-7#						
ODD	4-382#							
OFFTIM	4-23	16-419#						
OTMXV	1-59	20-81#						
OTRASZ	1-60	17-17#						
OTRECV	1-57	18-10#						
P1EXT	3-106#							
P1XPTR	3-106	3-223#						
PHONE	4-194	16-431#	22-164					
PHYMEM	1-50	3-148#						
PIDPTR	1-117	4-112#						
PMSIZE	1-73	16-675#	20-131					
PNAME	1-100	3-95	3-149#					
PNPTR	3-95#							
PRIDEF	4-184	16-338#						
PRIHI	4-183	16-334#						
PRILOW	4-182	16-333#						
PRIVIR	4-185	16-345#						
PROBRK	1-118	2-24#						
PROFLG	1-117	4-201#						
PROITP	1-130	22-183						
PROODC	1-118	2-25#						
PROSLT	1-100	4-113#						
PSW	1-132	3-126						
PVON	1-93	4-199#						
PVSPBL	1-59	16-598	16-598	16-598#				
PWCH	4-188	16-448#						
QBUS	1-114	4-300#	16-90					
QCOMP	1-98	3-69#						
QCOMPL	1-135	3-15						
QFREE	1-135	3-12						
QIO	1-135	3-13						
QUANO	4-4	16-258#						
QUAN1	1-58	4-5	16-267#					
QUAN1A	1-68	4-6	16-273#					
QUAN1B	1-68	4-7	16-277#					
QUAN1C	4-8	16-282#						
QUAN2	1-58	4-9	16-290#					
QUAN3	4-10	16-296#						
QUME	1-134							
R\$CFST	1-58	3-210#						
R\$CH17	1-112	3-203#						
R\$CHN	1-112	3-202#						

SEGCHN	1-90	4-58#				
SETERR	1-131					
SFCB	1-68	4-83#				
SFCBFH	1-53	4-85#				
SFCBND	1-68	4-84#				
SFCLS	1-125	4-162#				
SFRSST	1-124	4-156#				
SFSVST	1-124	4-155#				
SFWRIT	1-125	4-163#				
SHRRCB	1-51	4-89#				
SHRRCN	1-51	4-90#				
SILSIZ	18-17#	18-22	18-25#	18-29	18-32#	18-36
SILXOF	18-17#	18-22	18-25#	18-29	18-32#	18-36
SILXON	18-17#	18-22	18-25#	18-29	18-32#	18-36
SIZMEM	4-324	4-324	4-324	4-324	4-324#	
SLEDIT	4-180	16-221#				
SMONHD	1-48	4-82#				
SMRSIZ	1-116	4-334#				
SNBUX	1-89	16-598	16-598	16-598	16-598#	
SNDBX	1-87	4-87	16-598#			
SNMSHD	1-108	4-75#				
SPLANM	1-91	16-598#				
SPLBHD	1-53	16-598#				
SPLBLK	1-55	16-13#				
SPLCHN	1-101	16-598#				
SPLDEV	1-54	16-598	16-598#			
SPLDVN	1-54	16-598	16-598#			
SPLNB	1-52	16-598	16-598	16-598#		
SPLND	1-52	16-598	16-598	16-598#		
SPLNF	1-52	4-86	16-598	16-598#		
SPOLID	1-118	4-104#				
SPSTAT	1-61	3-98#				
SPUSR	1-99	3-70#				
SR3FLG	1-115	4-204#				
SRERR	18-17	18-17#	18-25	18-25#	18-32	18-32#
SRTSIZ	1-116	4-336#				
STPFLG	1-94	4-202#				
SWAPFL	4-173	16-51#	18-10			
SWDBLK	1-65	16-9#				
SWPCHN	1-102	3-193#	3-212			
SWPSLT	4-30	16-65#	18-10	18-10	18-10#	
SYCHO	1-95	3-43#				
SYCH1	1-95	3-44#				
SYCH10	1-96	3-51#				
SYCH11	1-96	3-52#				
SYCH12	1-96	3-53#				
SYCH13	1-96	3-54#				
SYCH14	1-97	3-55#				
SYCH15	1-97	3-56#				
SYCH16	1-97	3-57#				
SYCH17	1-97	3-58#	3-203			
SYCH2	1-95	3-45#				
SYCH20	1-97	3-59#				
SYCH3	1-95	3-46#				
SYCH4	1-95	3-47#				
SYCH5	1-95	3-48#				

VLDSYS	1-60	4-179#										
VLSWCH	4-187	16-442#										
VMAXMC	1-80	4-24#										
VMIOBF	1-81	4-192#										
VMIOSZ	1-79	4-193#										
VMLBLK	1-90	4-17#										
VMSCHR	1-79	4-25#										
VMXCSH	1-85	4-94#										
VMXFIL	1-108	4-19#										
VMXMON	1-61	4-21#										
VMXMRB	1-80	4-27#										
VMXMSG	1-80	4-26#										
VMXSF	1-90	4-14#										
VMXSFC	1-90	4-15#										
VMXWIN	1-55	4-34#										
VNCSLO	1-71	4-42#										
VNCXOF	1-71	4-43#										
VNCXON	1-71	4-44#										
VNFCSH	1-85	4-20#										
VNGR	1-51	4-32#										
VNRFLG	1-69	4-39#										
VNUIP	1-51	4-36#										
VNUMDC	1-81	4-16#										
VOFFTM	1-86	4-23#										
VPLAS	1-90	4-31#										
VPRIDF	1-99	4-184#										
VPRIHI	1-98	4-183#										
VPRILO	1-98	4-182#										
VPRIVR	1-64	4-185#										
VQUANO	1-73	4-4#										
VQUAN1	1-83	4-5#										
VQUAN2	1-83	4-9#										
VQUAN3	1-73	4-10#										
VQUN1A	1-83	4-6#										
VQUN1B	1-83	4-7#										
VQUN1C	1-69	4-8#										
VSCHED	1-72	4-40#										
VSLEDT	1-56	4-180#										
VSWPFL	1-92	4-173#										
VSWPSL	1-55	4-30#										
VSYDMP	1-70	4-171#										
VT100	1-133											
VT200	1-133											
VT52	1-133											
VTMDUT	1-87	4-22#										
VTSLCH	1-73	4-186#										
VU\$CL	1-52	4-177#										
VUCLMC	1-115	4-18#										
VUCLOR	1-115	4-178#										
VUSPHN	1-126	4-194#										
VUXIFL	1-62	4-176#										
VVLSCH	1-73	4-187#										
VVPWCH	1-73	4-188#										
WILDFL	1-64	16-496#										
WINBAS	1-80	4-67#										
X	16-543	16-543#	16-544	16-544#	16-545	16-545#	16-546	16-546#	16-547	16-547#	16-548	16-548#
	16-549	16-549#	16-550	16-550#	16-551	16-551#	16-551	16-551#	16-551#	16-551#	16-551#	16-551#

