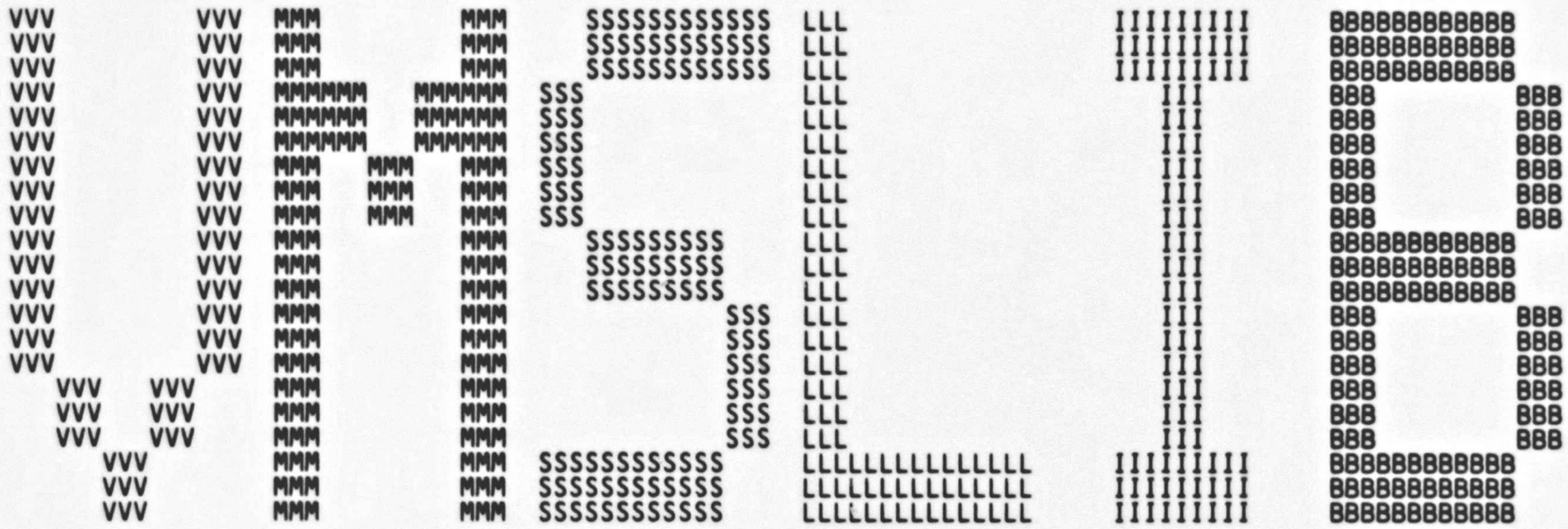
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB

```
  SSSSSSSS  TTTTTTTTTT    AAAAAA    RRRRRRRR    DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF  MM      MM  PPPPPPPP
  SSSSSSSS  TTTTTTTTTT    AAAAAA    RRRRRRRR    DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF  MM      MM  PPPPPPPP
SS              TT       AA    AA  RR      RR  DD      DD  EE          FF          MMMM  MMMM  PP      PP
SS              TT       AA    AA  RR      RR  DD      DD  EE          FF          MMMM  MMMM  PP      PP
SS              TT       AA    AA  RR      RR  DD      DD  EE          FF          MM  MM  MM  PP      PP
SS              TT       AA    AA  RR      RR  DD      DD  EE          FF          MM  MM  MM  PP      PP
  SSSSSS        TT       AA    AA  RRRRRRRR    DD      DD  EEEEEEEE    FFFFFFFF    MM      MM  PPPPPPPP
  SSSSSS        TT       AA    AA  RRRRRRRR    DD      DD  EEEEEEEE    FFFFFFFF    MM      MM  PPPPPPPP
        SS      TT       AAAAAAAAAA  RR  RR    DD      DD  EE          FF          MM      MM  PP
        SS      TT       AAAAAAAAAA  RR  RR    DD      DD  EE          FF          MM      MM  PP
        SS      TT       AA    AA  RR    RR    DD      DD  EE          FF          MM      MM  PP
        SS      TT       AA    AA  RR    RR    DD      DD  EE          FF          MM      MM  PP
SSSSSSSS        TT       AA    AA  RR      RR  DDDDDDDD    EEEEEEEEEE  FF          MM      MM  PP
SSSSSSSS        TT       AA    AA  RR      RR  DDDDDDDD    EEEEEEEEEE  FF          MM      MM  PP

  SSSSSSSS  DDDDDDDD    LL
  SSSSSSSS  DDDDDDDD    LL
SS          DD      DD  LL
SS          DD      DD  LL
SS          DD      DD  LL
SS          DD      DD  LL
  SSSSSS    DD      DD  LL
  SSSSSS    DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
SSSSSSSS    DDDDDDDD    LLLLLLLLLL
SSSSSSSS    DDDDDDDD    LLLLLLLLLL
```

```
{ STARDEFMP.MDL - system user interface definitions
{
{ Version:      'V04-000'
{
{*****************************************************************************
{*                                                                           *
{*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
{*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
{*  ALL RIGHTS RESERVED.                                                     *
{*                                                                           *
{*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
{*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
{*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
{*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
{*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
{*  TRANSFERRED.                                                             *
{*                                                                           *
{*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
{*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
{*  CORPORATION.                                                             *
{*                                                                           *
{*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
{*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
{*                                                                           *
{*****************************************************************************

{++
{
{ FACILITY:  VAX/VMS System Macro Libraries
{
{ ABSTRACT:
{
{       This file contains the SDL source for all user visible operating
{       system interfaces from M to P.
{
{ ENVIRONMENT:
{
{       n/a
{
{--


{ AUTHOR:  The VMS Group         CREATION DATE:  1-Aug-1976

{ MODIFIED BY:

{       V03-055 DAS0002         David Solomon           9-Jul-1984
{               Add MNT$V_NOREBUILD to mount flags longword.
{
{       V03-054 ACG0432         Andrew C. Goldstein,    6-Jul-1984  16:34
{               Add PRC$V_NOPASSWORD bit to $CREPRC flags
{
{       V03-053 MHB0156         Mark Bramhall           2-May-1984
{               Add SYSUAF_COP record type/id to $NSARECDEF.
{
```

```
        V03-052 RLRMVER         Robert L. Rappaport     26-Apr-1984
                Add MSG$_RC25MVER, MSG$_RDRXMVER, MSG$_TU81MVER, and
                MSG$_MAYAMVER symbols added to $MSGDEF.

        V03-051 TMH0051         Tim Halvorsen           12-Apr-1984
                Remove $M_ symbols from PRVDEF (V03-049) for bits in
                the second longword, since SDL cannot generate masks
                for bit offsets greater than 32.

        V03-050 MHB0139         Mark Bramhall           12-Apr-1984
                Add CLISPEC flag to $PRCDEF.

        V03-049 MCN0165         Maria del C. Nasr       09-Apr-1984
                Add mask values to $PRVDEF.

        V03-048 RSH0128         R. Scott Hanna          28-Mar-1984
                $NSARECDEF Add the packet type NSA$K_PKTTYP_STATUS and
                remove the record type ACL.

        V03-047 RSH0109         R. Scott Hanna          28-Feb-1984
                $NSARECDEF Change time field in the security auditing
                record header from a longword to a quadword.

        V03-046 HH0004          Hai Huang               28-Feb-1984
                Add MNT$V_CLUSTER for cluster-wide mount support.

        V03-045 ROW0318         Ralph O. Weber          27-FEB-1984
                Add OPCOM message codes for shadow set mount verification
                messages; MSG$_SHAMEMFAL, member failed out of shadow set, and
                MSG$_SHARDUCED, shadow set reduced.

        V03-044 MMD0241         Meg Dumont,     24-Feb-1984  11:15
                Add MTADEF codes to suppor the mag tape accessibility routine.

        V03-043 RSH0097         R. Scott Hanna          02-Feb-1984
                Replace $NSARECDEF.

        V03-042 KPL0002         Peter Lieberwirth        2-Feb-1984
                Add $PR8NNDEF for Nautilus.

        V03-041 ACG0386         Andrew C. Goldstein,    10-Jan-1984  16:29
                Add PRC$V_PASSWORD to $CREPRC flags

        V03-040 RLRPR8SS2       Robert L. Rappaport     9-Dec-1983
                Additional minor corrections to $PR8SSDEF.

        V03-039 DAS0001         David Solomon           29-Nov-1983
                Add MNT$_JRNLRECORD_SIZE for specifying max journal recordsize
                on MOUNT.

        V03-038 RLRPR8SS1       Robert L. Rappaport     28-Nov-1983
                Modify $PR8SSDEF according to new Scorpio spec.

        V03-037 RLRPR8SS        Robert L. Rappaport     11-Nov-1983
                Add $PR8SSDEF for Scorpio specific registers.
```

```
        V03-036 KPL0001         Peter Lieberwirth       8-Nov-1983
                Add PR$_SID_TYP8SS for Scorpio, PR$_SID_TYP8NN for Nautilus.

        V03-035 TMK0001         Todd M. Katz            27-Oct-1983
                Add the process quota list code PQL$_JTQUOTA.

        V03-034 KDM0075         Kathleen D. Morse       23-Aug-1983
                Update PR$_TYPMAX to 8.

        V03-033 CWH1011         CW Hobbs                18-Aug-1983
                Add MSG$_CLUMBX and MSG$_TM78MVER messages.

        V03-032 KDM0067         Kathleen D. Morse       4-Aug-1983
                Add processor-specific IPR macros for Micro-VAX, $PRUV1DEF
                and $PRUV2DEF.

        V03-031 SBL0031         Steve Lionel    29-Jul-1983
                Add comment to $PRVDEF about updating [RTL.SRC]LIBLEXICA.B32.

        V03-030 WMC0030         Wayne Cardoza   28-Jul-1983
                PRCDEF item codes for logical name attributes.

        V03-029 MMD0191         Meg Dumont,     28-Jul-1983  9:49
                Changed bit in VCBDEF from AUTO to NOAUTO to make mag tape
                AVL/AVR consistent between DCL and MOUNT system service

        V03-028 KDM0050         Kathleen D. Morse       15-Jul-1983
                Add cpu-dependent IPR definitions:  $PR730DEF, $PR750DEF,
                $PR780DEF, and $PR790DEF.  Remove ACCS, ACCR, PME, TODR,
                ICR, and NICR from $PRDEF and add to cpu-dependent IPRs.

        V03-027 MLJ0114         Martin L. Jack          22-Jun-1983
                Add MSG$_GETQUI.

        V03-026 RSH0035         R. Scott Hanna          16-Jun-1983
                Provide permanent fix for $NSARECDEF symbols.

        V03-025 ADE9001         A. Eldridge             27-May-1983
                Temporary modifications to $NSARECDEF to allow build
                to proceed.

        V03-025 RSH0023         R. Scott Hanna          24-May-1983
                Add $NSARECDEF (Security Auditing record definitions)

        V03-024 RSH0018         R. Scott Hanna          21-May-1983
                Add SECURITY privilege to $PRVDEF

        V03-023 KDM0046         Kathleen D. Morse       20-May-1983
                Add Micro-VAX cpu definitions to $PRDEF.

        V03-022 PRB0172         Paul Beck               26-Apr-1983
                Add TMPJNL and PRMJNL privileges

        V03-021 WMC0017         Wayne Cardoza           10-Apr-1983
                Add IMGDMP flag to PRCDEF
```

```
{       V03-020 MMD0111         Meg Dumont,     25-Mar-1983  9:51
        Added MNTDEF bits for new mount qualifiers

        V03-019 WMC0016         Wayne Cardoza           08-Mar-1983
        Add item codes to PRCDEF

        V03-018 WMC0015         Wayne Cardoza           06-Mar-1983
        Add PRC$V_INTER and PRC$V_DETACH

        V03-017 LMP0083         L. Mark Pilant,         28-Feb-1983  9:54
        Add a blurb that indicates what modules must be changed when
        privileges are added.

        V03-16  LMP0082         L. Mark Pilant,         28-Feb-1983  8:41
        Add difinitions for the following privileges: UPGRADE,
        DOWNGRADE, GRPPRV, and READALL.

        V03-015 JLV0233         Jake VanNoy             24-FEB-1983
        Add definition for SHARE privilege.

        V03-014 WMC0014         Wayne Cardoza           04-Jan-1983
        Add 790 scratchpad registers to PRDEF.

        V03-013 RLRDENS         Robert L. Rappaport     21-Dec-1982
        Add density support bit fields to MTDEF.  These bits
        will tell what densities are supported on a drive.

        V03-012 TCM0004         Trudy C. Matthews       13-Dec-1982
        Add PR$_STXCS and PR$_STXDB definitions.

        V03-011 ACG0303         Andrew C. Goldstein,    9-Dec-1982  16:06
        Add FILL attribute to extraneous field names

        V03-010 RLRSPEEDA       Robert L. Rappaport     09-Nov-1982
        Corrected Speed definition in MTDEF.

        V03-009 TCM0003         Trudy C. Matthews       02-Nov-1982
        Changed PR$_SID_TYP7VV to PR$_SID_TYP790.

        V03-008 RLRSPEED        Robert L. Rappaport     21-Oct-1982
        Add Speed field and speed values to MTDEF.

        V03-007 MLJ0097         Martin L. Jack, 9-Sep-1982  16:38
        Add MSG$_SNDJBC.

        V03-006 RLRSEREX        Robert L. Rappaport     26-Aug-1982
        Remove MT$M_CLSEREXCP.

        V03-005 STJ0320         Steven T. Jeffreys      25-Aug-1982
        Add support for recovery unit journalling in $MOUNT.

        V03-004 RLR0001         Robert L. Rappaport     4-Aug-1982
        Add serious exception bits to MTDEF

        V03-003 TCM0002         Trudy C. Matthews       28-Jul-1982
        Replace 11/790-specific Internal Processor Register definitions.
```

STA

con

/*+
/*
/*-

agg

end

end

```
;
;       V03-002 STJ0307         Steven T. Jeffreys      18-May-1982
;                Added MNT$V_NOUNLOAD.
;
;       V03-001 RLRV3A1         Robert L. Rappaport     5-Apr-1982
;               Added MSG$_UDA50MVER and MSG$_DUPUNITNO.
;----
```

STA

mod

/**+
/*
/*-

con

agg

end

end

```
module $MNTDEF;
/*+
/*
/* FLAG BITS FOR THE $MOUNT SYSTEM SERVICE.
/*
/*-


aggregate MNTDEF  union prefix MNT$;
    MNTDEF_BITS structure fill;
        FOREIGN bitfield mask;                  /* FOREIGN OPTION SELECTED
        GROUP bitfield mask;                    /* GROUP OPTION SELECTED
        NOASSIST bitfield mask;                 /* NOASSIST OPTION SELECTED
        NODISKQ bitfield mask;                  /* NODISKQ OPTION SELECTED
        NOHDR3 bitfield mask;                   /* NOHDR3 OPTION SELECTED
        NOLABEL bitfield mask;                  /* NOLABEL OPTION SELECTED
        NOWRITE bitfield mask;                  /* NOWRITE OPTION SELECTED
        OVR_ACCESS bitfield mask;               /* OVERRIDE ACCESSIBLITY OPTION SELECTED
        OVR_EXP bitfield mask;                  /* OVERRIDE EXPIRATON OPTION SELECTED
        OVR_IDENT bitfield mask;                /* OVERRIDE VOLUME LABEL
        OVR_SETID bitfield mask;                /* OVERRIDE VOLUME SET IDENT OPTION SELECTIED
        READCHECK bitfield mask;                /* READCHECK OPTION SELECTED
        SHARE bitfield mask;                    /* SHARE OPTION SELECTED
        MESSAGE bitfield mask;                  /* ALLOW $MOUNT TO PRINT MESSAGES
        SYSTEM bitfield mask;                   /* SYSTEM OPTION SELECTED
        WRITECHECK bitfield mask;               /* WRITECHECK OPTION SELECTED
        WRITETHRU bitfield mask;                /* WRITETHRU OPTION SELECTED
        NOCACHE bitfield mask;                  /* TURN OFF ALL CACHING
        OVR_LOCK bitfield mask;                 /* OVERRIDE AUTOMATIC WRITE-LOCK
        NOMNTVER bitfield mask;                 /* DISABLE MOUNT VERIFICATION
        NOUNLOAD bitfield mask;                 /* DO NOT UNLOAD VOLUME AT DISMOUNT
        NOJRNL bitfield mask;                   /* DO NOT ACTIVATE RECOVERY UNIT JOURNAL FILE
        NEWJRNL bitfield mask;                  /* CREATE A NEW RECOVERY UNIT JOURNAL FILE
        NOAUTO bitfield mask;                   /* DO NOT SET THE MTAACP INTO AVR AND AVL MODE
        INIT_ALL bitfield mask;                 /* INITIALIZE ALL VOLUMES IN SET BEFORE WRITING
        INIT_CONT bitfield mask;                /* INITIALIZE CONTINUATION VOLUMES BEFORE WRITING
        OVR_VOLO bitfield mask;                 /* OVERRIDE VOL1 VOLUME IDENTIFIER FIELD
        INTERCHG bitfield mask;                 /* VOL FOR INTERCHG NO VMS SPECIFIC INFO WRITTEN TO TAPE
        CLUSTER bitfield mask;                  /* CLUSTER-WIDE MOUNT OPTION SELECTED
        NOREBUILD bitfield mask;                /* DO NOT REBUILD VOLUME
    end MNTDEF_BITS;

/*
/* Item codes for mount parameters.
/*
                                                /* DEFINE CODES AS CONSTANTS
    constant(
          DEVNAM                                /* DEVICE NAME
        , VOLNAM                                /* VOLUME NAME
        , LOGNAM                                /* LOGICAL NAME
        , FLAGS                                 /* MOUNT FLAGS
        , ACCESSED                              /* ACCESSED VALUE
        , PROCESSOR                             /* PROCESSOR NAME
        , VOLSET                                /* VOLUME SET NAME
        , BLOCKSIZE                             /* BLOCKSIZE VALUE
        , DENSITY                               /* TAPE DENSITY VALUE
```

```
        , EXTENT                                        /* NUMBER OF EXTENT CACHE ENTRIES
        , FILEID                                        /* FILE ID CACHE SIZE
        , LIMIT                                         /* EXTENT CACHE LIMIT
        , OWNER                                         /* VOLUME OWNER UIC
        , VPROT                                         /* VOLUME PROTECTION
        , QUOTA                                         /* QUOTA CACHE SIZE
        , RECORDSIZ                                     /* RECORD SIZE VALUE
        , WINDOW                                        /* NUMBER OF WINDOWS
        , EXTENSION                                     /* DEFAULT FILE EXTENSION
        , VISUAL_ID                                     /* VISUAL IDENTIFICATION
        , COMMENT                                       /* USER COMMENT
        , JRNLSIZE                                      /* INITIAL JOURNAL SIZE
        , JRNLEXTEND                                    /* JOURNAL EXTENSION QUANTITY
        , JRNLQUOTA                                     /* JOURNAL BYTE QUOTA (PER R.U.)
        , JRNLRECORD_SIZE                                       /* JOURNAL MAXIMUM RECORD SIZE
        ) equals 1 increment 1  prefix MNT tag $;
end MNTDEF;

end_module $MNTDEF;
```

```
module $MSGDEF;
/*+
/*
/* SYSTEM WIDE MAILBOX MESSAGE TYPES
/*
/*-

                                                        /* DEFINE CODES AS CONSTANTS

constant(
        TRMUNSOLIC                                      /* UNSOLICITED TERMINAL DATA
      , CRUNSOLIC                                       /* UNSOLICTED CARD READER DATA
      , DELPROC                                         /* DELETE PROCESS
      , SNDSMB                                          /* SEND TO SYMBIONT MANAGER
      , DEVOFFLIN                                       /* DEVICE OFFLINE
      , TRMHANGUP                                       /* TERMINAL HANG UP
      , DEVONLIN                                        /* DEVICE ONLINE
      , OPRQST                                          /* OPERATOR REQUEST *** OVERLAPPED CODE ***
      , OPREPLY                                         /* OPERATOR REPLY *** OVERLAPPED CODE ***
      ) equals 1 increment 1  prefix MSG tag $;

                                                        /* DEFINE SYMBIONT RESPONSE MESSAGES

constant(
        SMBINI                                          /* SYMBIONT HAS INITED
      , SMBDON                                          /* SYMBIONT FINISHED
      , SNDACC                                          /* SEND MESSAGE TO ACCOUNTING MANAGER
      , PURPROC                                         /* PURGE PROCESS *** OVERLAPPED CODE ***
      , DELIMAG                                         /* DELETE IMAGE *** OVERLAPPED CODE ***
      , PURIMAG                                         /* PURGE IMAGE *** OVERLAPPED CODE ***
      , SYSFUNC                                         /* SYSTEM FUNCTION *** OVERLAPPED CODE ***
      , SNDJBC                                          /* Send message to job controller
      , GETQUI                                          /* Get queue information (from job controller)
      ) equals 8 increment 1  prefix MSG tag $;

                                                        /* DEFINE DMC MESSAGES

constant(
        XM_DATAVL                                       /* DMC UNSOLICITED DATA
      , XM_SHUTDN                                       /* DMC LINE DOWN
      , XM_ATTN                                         /* DMC ATTENTION MESSAGE
      ) equals 11 increment 1  prefix MSG tag $;

                                                        /* SYMBIONT COMMAND MESSAGES

constant(
        INIOPR                                          /* INITIATE PRINTING A FILE
      , ABOOPR                                          /* ABORT PRINTING A FILE
      , SUSOPR                                          /* PAUSE PRINTING THE FILE
      , RESOPR                                          /* RESUME PRINTING THE FILE
      , DELSMB                                          /* SYMBIONT SHOULD DELETE ITSELF
      , REQUE                                           /* REQUEUE A FILE FOR PRINTING
      ) equals 16 increment 1  prefix MSG tag $;

                                                        /*

constant(
        SMBRSP                                          /* SYMBIONT MANAGER RESPONSE
      , ACCRSP                                          /* ACCOUNTING MANAGER RESPONSE
      ) equals 32 increment 1  prefix MSG tag $;

                                                        /* FILE ACP MESSAGES

constant(
        SCANBAD                                         /* SCAN FILE FOR BAD BLOCKS
      , SCANRSP                                         /* RESPONSE FROM FILE SCANNER
      ) equals 40 increment 1  prefix MSG tag $;

                                                        /* NETWORK ATTENTION CODES
```

```
constant(
        ABORT                                   /* PARTNER ABORTED LINK
      , CONFIRM                                 /* CONNECT CONFIRM
      , CONNECT                                 /* INBOUND CONNECT INITIATE
      , DISCON                                  /* PARTNER DISCONNECTED - HANGUP
      , EXIT                                    /* PARTNER EXITED PREMATURELY
      , INTMSG                                  /* INTERRUPT MESSAGE - UNSOLICITED DATA
      , PATHLOST                                /* NFW - PATH LOST TO PARTNER
      , PROTOCOL                                /* PROTOCOL ERROR
      , REJECT                                  /* CONNECT REJECT
      , THIRDPARTY                              /* THIRD PARTY DISCONNECT
      , TIMEOUT                                 /* CONNECT TIMEOUT
      , NETSHUT                                 /* Network shutting down
      , NODEACC                                 /* Node has become accessible
      , NODEINACC                               /* Node has become inaccessible
      , EVTAVL                                  /* Events are available to EVL
      , EVTRCVCHG                               /* Event receiver database change
      , INCDAT                                  /* X25 INCOMING DATA
      , RESET                                   /* X25 CIRCUIT RESET
      , LINUP                                   /* X25 PVC LINE UP
      , LINDWN                                  /* X25 PVC LINE DOWN
      , EVTXMTCHG                               /* Event transmitter database change
      ) equals 48 increment 1  prefix MSG tag $;

                                                /* MOUNT VERIFICATION MESSAGES

constant(
        DEVOFFLINX                              /* DEVICE OFFLINE
      , WRONGVOL                                /* WRONG VOLUME IN DEVICE
      , DEVWRTLCK                               /* DEVICE HAS BEEN WRITE LOCKED
      , TRMBRDCST                               /* TERMINAL BROADCAST
      , MVCOMPLETE                              /* MOUNT VERIFICATION COMPLETED
      , MVABORTED                               /* MOUNT VERIFICATION ABORTED
      , DISMOUNTED                              /* VOLUME DISMOUNTED
      , UDA50MVER                               /* UDA50 MICORCODE NOT UPTO REV
      , DUPUNITNO                               /* MSCP CONTROLLER - DUPLICATE UNIT !
      , CLUMBX                                  /* CNXMGR to OPCOM messages
      , TM78MVER                                /* TM78 Microcode not up to rev level
      , SHAMEMFAL                               /* Member failed out of shadow set
      , SHARDUCED                               /* Shadow set reduced
      , RC25MVER                                /* RC25 MICORCODE NOT UPTO REV
      , RDRXMVER                                /* RDRX MICORCODE NOT UPTO REV
      , TU81MVER                                /* TU81 MICORCODE NOT UPTO REV
      , MAYAMVER                                /* MAYA MICORCODE NOT UPTO REV
      ) equals 80 increment 1  prefix MSG tag $;

end_module $MSGDEF;
```

```
module $MTADEF;
/*+
/* MAGTAPE ACCESSIBILTY ROUTINE CODES
/*-

aggregate MTADEF union prefix MTA$;

/* DEFINITIONS FOR ACCESS_SPEC

    constant NOCHAR     equals 0  prefix MTA tag $K;    /* ACCESS CHAR IS NOVALID
    constant CHARVALID  equals 1  prefix MTA tag $K;    /* ACCESS CHAR IS VALID

/* DEFINITIONS FOR TYPE

    constant INVOL1     equals 0  prefix MTA tag $K;    /* INPUT A VOL1 ACCESS CODE
    constant INHDR1     equals 1  prefix MTA tag $K;    /* INPUT A HDR1 ACCESS CODE
    constant OUTVOL1    equals 2  prefix MTA tag $K;    /* OUTPUT A VOL1 ACCESS CODE
    constant OUTHDR1    equals 3  prefix MTA tag $K;    /* OUTPUT A HDR1 ACCESS CODE

end MTADEF;

end_module $MTADEF;
```

```
module $MTDEF;
/*+
/* MAGTAPE STATUS BITS
/*-

aggregate MTDEF  union prefix MT$;
    MTDEF_BITS structure fill;
        SEREXCP bitfield mask;                  /* SERIOUS EXCEPTION PRESENT
        FILL_1 bitfield fill prefix MTDEF tag $$;   /* SPARE UNUSED BIT
        ENSEREXCP bitfield mask;                /* ENABLE SERIOUS EXCEPTION MODE
        PARITY bitfield mask;                   /* PARITY SELECT (0=ODD, 1=EVEN)
        FORMAT bitfield mask length 4;          /* RECORDING FORMAT
        DENSITY bitfield mask length 5;         /* RECORDING DENSITY AND METHOD
        FILL_2 bitfield fill prefix MTDEF tag $$;   /* SPARE UNUSED BIT
        LOGSOFT bitfield mask;                  /* LOG SOFT (TU78) ERRORS (0=NO, 1=YES)
        LOGSOFTOG bitfield mask;                /* TOGGLE TO REVERSE LOGSOFT STATE BIT
        BOT bitfield mask;                      /* AT BEGINNING OF TAPE
        EOF bitfield mask;                      /* AT END OF FILE
        EOT bitfield mask;                      /* AT END OF TAPE
        HWL bitfield mask;                      /* TAPE IS HARDWARE WRITELOCKED
        LOST bitfield mask;                     /* TAPE POSITION LOST
        SUP_NRZI bitfield mask;                 /* DRIVE SUPPORTS NRZI (800 BPI)
        SUP_PE bitfield mask;                   /* DRIVE SUPPORTS PE (1600 BPI)
        SUP_GCR bitfield mask;                  /* DRIVE SUPPORTS GCR (6250 BPI)
        SPEED bitfield mask length 8;           /* TAPE SPEED
    end MTDEF_BITS;

/*
/* RECORDING FORMAT DEFINTIONS
/*

                                                /*
    constant "DEFAULT"  equals 0  prefix MT tag $K;     /* DEFAULT FORMAT
    constant NORMAL11   equals 12  prefix MT tag $K;    /* PDP-11 NORMAL
    constant CORDMP11   equals 13  prefix MT tag $K;    /* PDP-11 CORE DUMP
    constant NORMAL15   equals 14  prefix MT tag $K;    /* PDP-15 NORMAL

/*
/* RECORDING DENTITY AND METHOD DEFINITIONS
/*

                                                /*
/*              DEFAULT,0                   /* DEFAULT DENSITY (SAME AS ABOVE)
    constant NRZI_800   equals 3  prefix MT tag $K;     /* NRZI 800 BPI
    constant PE_1600    equals 4  prefix MT tag $K;     /* PE 1600 BPI
    constant GCR_6250   equals 5  prefix MT tag $K;     /* GCR 6250 BPI

/*
/* TAPE SPEED VALUE DEFINITIONS
/*

    constant SPEED_DEF  equals 0  prefix MT tag $K;     /* DEFAULT SPEED
    constant SPEED_25   equals 25  prefix MT tag $K;    /* 25 IPS
    constant SPEED_75   equals 75  prefix MT tag $K;    /* 75 IPS

end MTDEF;
```

```
end_module $MTDEF;
```

```
STA

mod
/*+
/*
/*-

con
con
con
con
con
con
con
con
con

con
con
con
con
con
con
con
con

end
```

```
module $NSARECDEF;

/*+
/* Security Auditing record definitions
/*-

constant REC_MAXLENGTH equals 1024 tag C prefix NSA$; /* Maximum record size
constant REC_MAXLENGTH equals 1024 tag K prefix NSA$; /* Maximum record size
constant REC_MAXLENGTH equals 1024 tag S prefix NSA$; /* Maximum record size


/*+
/* Audit record type definitions
/*-

constant (RECTYP_FIL,                   /* File access
          RECTYP_SYSUAF,                /* System UAF
          RECTYP_NETUAF,                /* Network UAF
          RECTYP_LOGB,                  /* Login breakin detection
          RECTYP_LOGI,                  /* Successful login
          RECTYP_LOGF,                  /* Login failure
          RECTYP_LOGO,                  /* Logout
          RECTYP_VOL)                   /* Volume operations
         equals 1 increment 1 counter #TYPNUM prefix NSA$;

constant RECTYPNUM equals #TYPNUM prefix NSA$;


/*+
/* Audit record subtype and ID definitions
/*-

/* File access

constant (RECTYP_FIL_SUCC,              /* Successful file access
          RECTYP_FIL_FAIL)              /* File access failure
         equals 1 increment 1 counter #SUBTYPNUM prefix NSA$;

constant RECTYPNUM_FIL equals #SUBTYPNUM prefix NSA$;

constant RECID_FIL_SUCC equals NSA$K_RECTYP_FIL+(65536*NSA$K_RECTYP_FIL_SUCC) prefix NSA$;
constant RECID_FIL_FAIL equals NSA$K_RECTYP_FIL+(65536*NSA$K_RECTYP_FIL_FAIL) prefix NSA$;


/* System UAF

constant (RECTYP_SYSUAF_ADD,            /* System UAF record addition
          RECTYP_SYSUAF_DEL,            /* System UAF record deletion
          RECTYP_SYSUAF_MOD,            /* System UAF record modification
          RECTYP_SYSUAF_COP,            /* System UAF record copied
          RECTYP_SYSUAF_REN)            /* System UAF record renamed
         equals 1 increment 1 counter #SUBTYPNUM prefix NSA$;

constant RECTYPNUM_SYSUAF equals #SUBTYPNUM prefix NSA$;

constant RECID_SYSUAF_ADD equals NSA$K_RECTYP_SYSUAF+(65536*NSA$K_RECTYP_SYSUAF_ADD) prefix NSA$;
```

```
constant RECID_SYSUAF_DEL equals NSA$K_RECTYP_SYSUAF+(65536*NSA$K_RECTYP_SYSUAF_DEL) prefix NSA$;
constant RECID_SYSUAF_MOD equals NSA$K_RECTYP_SYSUAF+(65536*NSA$K_RECTYP_SYSUAF_MOD) prefix NSA$;
constant RECID_SYSUAF_COP equals NSA$K_RECTYP_SYSUAF+(65536*NSA$K_RECTYP_SYSUAF_COP) prefix NSA$;
constant RECID_SYSUAF_REN equals NSA$K_RECTYP_SYSUAF+(65536*NSA$K_RECTYP_SYSUAF_REN) prefix NSA$;

/* Network UAF

constant (RECTYP_NETUAF_ADD,            /* Network UAF record addition
          RECTYP_NETUAF_DEL,            /* Network UAF record deletion
          RECTYP_NETUAF_MOD)            /* Network UAF record modification
         equals 1 increment 1 counter #SUBTYPNUM prefix NSA$;

constant RECTYPNUM_NETUAF equals #SUBTYPNUM prefix NSA$;

constant RECID_NETUAF_ADD equals NSA$K_RECTYP_NETUAF+(65536*NSA$K_RECTYP_NETUAF_ADD) prefix NSA$;
constant RECID_NETUAF_DEL equals NSA$K_RECTYP_NETUAF+(65536*NSA$K_RECTYP_NETUAF_DEL) prefix NSA$;
constant RECID_NETUAF_MOD equals NSA$K_RECTYP_NETUAF+(65536*NSA$K_RECTYP_NETUAF_MOD) prefix NSA$;

/* Login breakin detection

constant (RECTYP_LOGB_DIA,              /* Dialup interactive breakin detection
          RECTYP_LOGB_LOC,              /* Local interactive breakin detection
          RECTYP_LOGB_REM,              /* Remote interactive breakin detection
          RECTYP_LOGB_NET,              /* Network breakin detection
          RECTYP_LOGB_DET)              /* Detached process breakin detection
         equals 1 increment 1 counter #SUBTYPNUM prefix NSA$;

constant RECTYPNUM_LOGB equals #SUBTYPNUM prefix NSA$;

constant RECID_LOGB_DIA equals NSA$K_RECTYP_LOGB+(65536*NSA$K_RECTYP_LOGB_DIA) prefix NSA$;
constant RECID_LOGB_LOC equals NSA$K_RECTYP_LOGB+(65536*NSA$K_RECTYP_LOGB_LOC) prefix NSA$;
constant RECID_LOGB_REM equals NSA$K_RECTYP_LOGB+(65536*NSA$K_RECTYP_LOGB_REM) prefix NSA$;
constant RECID_LOGB_NET equals NSA$K_RECTYP_LOGB+(65536*NSA$K_RECTYP_LOGB_NET) prefix NSA$;
constant RECID_LOGB_DET equals NSA$K_RECTYP_LOGB+(65536*NSA$K_RECTYP_LOGB_DET) prefix NSA$;

/* Successful login

constant (RECTYP_LOGI_BAT,              /* Batch process login
          RECTYP_LOGI_DIA,              /* Dialup interactive login
          RECTYP_LOGI_LOC,              /* Local interactive login
          RECTYP_LOGI_REM,              /* Remote interactive login
          RECTYP_LOGI_NET,              /* Network login
          RECTYP_LOGI_SUB,              /* Subprocess login
          RECTYP_LOGI_DET)              /* Detached process login
         equals 1 increment 1 counter #SUBTYPNUM prefix NSA$;

constant RECTYPNUM_LOGI equals #SUBTYPNUM prefix NSA$;

constant RECID_LOGI_BAT equals NSA$K_RECTYP_LOGI+(65536*NSA$K_RECTYP_LOGI_BAT) prefix NSA$;
constant RECID_LOGI_DIA equals NSA$K_RECTYP_LOGI+(65536*NSA$K_RECTYP_LOGI_DIA) prefix NSA$;
constant RECID_LOGI_LOC equals NSA$K_RECTYP_LOGI+(65536*NSA$K_RECTYP_LOGI_LOC) prefix NSA$;
constant RECID_LOGI_REM equals NSA$K_RECTYP_LOGI+(65536*NSA$K_RECTYP_LOGI_REM) prefix NSA$;
constant RECID_LOGI_NET equals NSA$K_RECTYP_LOGI+(65536*NSA$K_RECTYP_LOGI_NET) prefix NSA$;
```


end

end

```
constant RECID_LOGI_SUB equals NSA$K_RECTYP_LOGI+(65536*NSA$K_RECTYP_LOGI_SUB) prefix NSA$;
constant RECID_LOGI_DET equals NSA$K_RECTYP_LOGI+(65536*NSA$K_RECTYP_LOGI_DET) prefix NSA$;

/* Login failure

constant (RECTYP_LOGF_BAT,              /* Batch process login failure
          RECTYP_LOGF_DIA,              /* Dialup interactive login failure
          RECTYP_LOGF_LOC,              /* Local interactive login failure
          RECTYP_LOGF_REM,              /* Remote interactive login failure
          RECTYP_LOGF_NET,              /* Network login failure
          RECTYP_LOGF_SUB,              /* Subprocess login failure
          RECTYP_LOGF_DET)              /* Detached process login failure
        equals 1 increment 1 counter #SUBTYPNUM prefix NSA$;

constant RECTYPNUM_LOGF equals #SUBTYPNUM prefix NSA$;

constant RECID_LOGF_BAT equals NSA$K_RECTYP_LOGF+(65536*NSA$K_RECTYP_LOGF_BAT) prefix NSA$;
constant RECID_LOGF_DIA equals NSA$K_RECTYP_LOGF+(65536*NSA$K_RECTYP_LOGF_DIA) prefix NSA$;
constant RECID_LOGF_LOC equals NSA$K_RECTYP_LOGF+(65536*NSA$K_RECTYP_LOGF_LOC) prefix NSA$;
constant RECID_LOGF_REM equals NSA$K_RECTYP_LOGF+(65536*NSA$K_RECTYP_LOGF_REM) prefix NSA$;
constant RECID_LOGF_NET equals NSA$K_RECTYP_LOGF+(65536*NSA$K_RECTYP_LOGF_NET) prefix NSA$;
constant RECID_LOGF_SUB equals NSA$K_RECTYP_LOGF+(65536*NSA$K_RECTYP_LOGF_SUB) prefix NSA$;
constant RECID_LOGF_DET equals NSA$K_RECTYP_LOGF+(65536*NSA$K_RECTYP_LOGF_DET) prefix NSA$;

/* Logout

constant (RECTYP_LOGO_BAT,              /* Batch process logout
          RECTYP_LOGO_DIA,              /* Dialup interactive logout
          RECTYP_LOGO_LOC,              /* Local interactive logout
          RECTYP_LOGO_REM,              /* Remote interactive logout
          RECTYP_LOGO_NET,              /* Network logout
          RECTYP_LOGO_SUB,              /* Subprocess logout
          RECTYP_LOGO_DET)              /* Detached process logout
        equals 1 increment 1 counter #SUBTYPNUM prefix NSA$;

constant RECTYPNUM_LOGO equals #SUBTYPNUM prefix NSA$;

constant RECID_LOGO_BAT equals NSA$K_RECTYP_LOGO+(65536*NSA$K_RECTYP_LOGO_BAT) prefix NSA$;
constant RECID_LOGO_DIA equals NSA$K_RECTYP_LOGO+(65536*NSA$K_RECTYP_LOGO_DIA) prefix NSA$;
constant RECID_LOGO_LOC equals NSA$K_RECTYP_LOGO+(65536*NSA$K_RECTYP_LOGO_LOC) prefix NSA$;
constant RECID_LOGO_REM equals NSA$K_RECTYP_LOGO+(65536*NSA$K_RECTYP_LOGO_REM) prefix NSA$;
constant RECID_LOGO_NET equals NSA$K_RECTYP_LOGO+(65536*NSA$K_RECTYP_LOGO_NET) prefix NSA$;
constant RECID_LOGO_SUB equals NSA$K_RECTYP_LOGO+(65536*NSA$K_RECTYP_LOGO_SUB) prefix NSA$;
constant RECID_LOGO_DET equals NSA$K_RECTYP_LOGO+(65536*NSA$K_RECTYP_LOGO_DET) prefix NSA$;

/* Volume operations

constant (RECTYP_VOL_MOU,               /* Volume mounts
          RECTYP_VOL_DMOU)              /* Volume dismounts
        equals 1 increment 1 counter #SUBTYPNUM prefix NSA$;

constant RECTYPNUM_VOL equals #SUBTYPNUM prefix NSA$;
```

```
constant RECID_VOL_MOU equals NSA$K_RECTYP_VOL+(65536*NSA$K_RECTYP_VOL_MOU) prefix NSA$;
constant RECID_VOL_DMOU equals NSA$K_RECTYP_VOL+(65536*NSA$K_RECTYP_VOL_DMOU) prefix NSA$;


/*+
/* Record header offset definitions
/*-

aggregate NSARECHDRDEF structure prefix NSA$;

    REC_ID_OVERLAY union fill;
        REC_ID longword unsigned;           /* Record identification longword
        REC_ID_FIELDS structure fill;
            REC_TYPE word unsigned;         /* Record type
            REC_SUBTYPE word unsigned;      /* Record subtype
        end REC_ID_FIELDS;
    end REC_ID_OVERLAY;
    REC_SEQNUM byte unsigned;               /* This records sequence number
    REC_SEQLAST byte unsigned;              /* Last records sequence number
    REC_FLAGS_OVERLAY union fill;
        REC_FLAGS byte unsigned;            /* Record flags byte
        REC_FLAGS_BITS structure fill;
            REC_FLAGS_PKTCON bitfield length 1 mask; /* Last packet in record is
                                                     /* continued in next record
        end REC_FLAGS_BITS;
    end REC_FLAGS_OVERLAY;
    REC_PKTNUM byte unsigned;               /* Number of data packets in record
    REC_PKTOFF word unsigned;               /* Offset to first packet
    REC_PKTHDRSIZ word unsigned;            /* Data packet header size
    REC_EPID longword unsigned;             /* Extended PID
    REC_TIME quadword unsigned;             /* Event time (EXE$GQ_SYSTIME)
    REC_CLUSNAM character length 16;        /* Cluster node name
    REC_PROCNAM character length 16;        /* Process name
    REC_USERNAM character length 12;        /* username
    REC_ACCTNAM character length 8;         /* Account name

    constant RECHDR_LENGTH equals . tag C;
    constant RECHDR_LENGTH equals . tag K;

end NSARECHDRDEF;


/*+
/* Data packet type definitions
/*-

constant (PKTTYP_IMGNAM,                    /* Image name packet
          PKTTYP_FACMOD,                    /* File access mode
          PKTTYP_PRIVUSED,                  /* Privilege used to access file
          PKTTYP_FILNAM,                    /* File name
          PKTTYP_DEVNAM,                    /* Device name
          PKTTYP_LOGNAM,                    /* Logical name
          PKTTYP_VOLNAM,                    /* Volume name
          PKTTYP_VOLSNAM,                   /* Volume set name
          PKTTYP_NODENAM,                   /* Node name
          PKTTYP_USERNAM,                   /* User name
```

```
        PKTTYP_PASSWORD,                /* Password
        PKTTYP_UIC,                     /* User identification code
        PKTTYP_VOLPRO,                  /* Volume protection
        PKTTYP_MOUFLG,                  /* Mount flags
        PKTTYP_DMOUFLG,                 /* Dismount flags
        PKTTYP_NODEID,                  /* Node ID
        PKTTYP_EPID,                    /* Extended PID
        PKTTYP_SYSUAFF,                 /* System UAF record fields
        PKTTYP_STATUS)                  /* Status longword
      equals 1 increment 1 counter #PKTTYPNUM prefix NSA$;

constant PKTTYPNUM equals #PKTTYPNUM prefix NSA$;


/*+
/* Data packet offset definitions
/*-

aggregate NSAPKTDEF structure origin PKT_DATA prefix NSA$;

    #PKTHDRBEG = .;
    PKT_TYPE word unsigned;                     /* Packet data type
    PKT_SIZE word unsigned;                     /* Packet size
    PKT_DATA character length 0;

    constant PKTHDR_LENGTH equals .-#PKTHDRBEG tag C;
    constant PKTHDR_LENGTH equals .-#PKTHDRBEG tag K;

    PKT_DATA_OVERLAY union fill;
        PKT_IMGNAM character length 444;        /* Image name
        PKT_FACMOD longword unsigned;           /* File access mode
        PKT_PRIVUSED longword unsigned;         /* Privilege used to access file
        PKT_FILNAM character length 444;        /* File name
        PKT_DEVNAM character length 15;         /* Device name
        PKT_LOGNAM character length 255;        /* Logical name
        PKT_VOLNAM character length 12;         /* Volume name
        PKT_VOLSNAM character length 12;        /* Volume set name
        PKT_NODENAM character length 6;         /* Node name
        PKT_USERNAM character length 12;        /* User name
        PKT_PASSWORD character length 31;       /* Password
        PKT_UIC longword unsigned;              /* Volume UIC
        PKT_VOLPRO word unsigned;               /* Volume protection
        PKT_MOUFLG longword unsigned;           /* Mount flags
        PKT_DMOUFLG word unsigned;              /* Dismount flags
        PKT_NODEID quadword unsigned;           /* Node ID
        PKT_EPID longword unsigned;             /* Extended PID
        PKT_SYSUAFF quadword unsigned;          /* System UAF record fields
        PKT_STATUS longword unsigned;           /* Status longword
    end PKT_DATA_OVERLAY;
end NSAPKTDEF;

end_module $NSARECDEF;
```

```
module $OPRDEF;

/++
/* OPERATOR COMMUNICATIONS MESSAGE TYPES AND VALUES
/*-

                                                        /* OPERATOR MESSAGE TYPES
constant(
      TERMENABL                                         /* ENABLE TERMINAL
    , LOGINIT                                           /* INITIALIZE THE LOG
    , OPRQST                                            /* OPERATOR REQUEST
    , OPREPLY                                           /* OPERATOR REPLY
    ) equals 1 increment 1  prefix OPR tag $;

aggregate OPRDEF  union prefix OPR$;
    OPRDEF_BITS structure fill;
        CENTRAL bitfield mask;                          /*
        PRINTER bitfield mask;
        TAPES bitfield mask;
        DISKS bitfield mask;
        DEVICES bitfield mask;
        USER1 bitfield mask length 12;
        USER2 bitfield mask;
        USER3 bitfield mask;
        USER4 bitfield mask;
        USER5 bitfield mask;
        USER6 bitfield mask;
        USER7 bitfield mask;
        USER8 bitfield mask;
        USER9 bitfield mask;
        USER10 bitfield mask;
        USER11 bitfield mask;
        USER12 bitfield mask;
    end OPRDEF_BITS;
end OPRDEF;

end_module $OPRDEF;
```

```
module $PCCDEF;
/*+
/* PRINTER/TERMINAL CARRIAGE CONTROL SPECIFIERS
/*--

aggregate PCCDEF structure prefix PCC$;
    FORTRAN word unsigned;                              /* FORTRAN FIELD
    "PREFIX" byte unsigned;                             /* PREFIX FIELD
    POSTFIX_OVERLAY union fill;
        POSTFIX byte unsigned;                          /* POSTFIX FIELD
        POSTFIX_BITS0 structure fill;
            CHAR bitfield mask length 5;                /* CHARACTER FIELD
            FILL_1 bitfield fill prefix PCCDEF tag $$;
            EIGHTBIT bitfield mask;                     /* EIGHTBIT CHARACTER SET
            SINGLE bitfield mask;                       /* SINGLE CHARACTER
        end POSTFIX_BITS0;
        POSTFIX_BITS1 structure fill;
            LINECNT bitfield mask length 7;             /* LINE COUNT FOR NEWLINES
        end POSTFIX_BITS1;
                                                        /* FORTRAN CONSTANTS
        constant FTN_SINGLE     equals 32  prefix PCC tag $;/* SINGLE SPACE
        constant FTN_DOUBLE     equals 48  prefix PCC tag $;/* DOUBLE SPACE
        constant FTN_PAGE       equals 49  prefix PCC tag $;/* PAGE SPACE
        constant FTN_OVRPRT     equals 43  prefix PCC tag $;/* OVER PRINT
        constant FTN_PROMPT     equals 36  prefix PCC tag $;/* PROMPT
    end POSTFIX_OVERLAY;
end PCCDEF;

end_module $PCCDEF;
```

```
module $PLVDEF;
/*+
/* PRIVILEGED LIBRARY VECTOR DEFINITION
/*-



aggregate PLVDEF structure prefix PLV$;
    TYPE longword unsigned;                             /*TYPE CODE FOR VECTOR FORMAT
                                                        /*TYPE CODES FOR PRIVILEGE VECTORS
    constant(
          TYP_CMOD                                      /*CHANGE MODE VECTOR TYPE
        , TYP_MSG                                       /*MESSAGE VECTOR TYPE
        ) equals 1 increment 1 prefix PLV tag $C;
    VERSION longword unsigned;                          /*SYSTEM VERSION NUMBER
    KERNEL_OVERLAY union fill;
        KERNEL longword unsigned;                       /*SELF-REL PTR TO KERNEL MODE DISPATCHER
        MSGDSP longword unsigned;                       /*SELF-REL PTR TO MESSAGE DISPATCHER
    end KERNEL_OVERLAY;
    EXEC longword unsigned;                             /*SELF-REL PTR TO EXEC MODE DISPATCHER
    USRUNDWN longword unsigned;                         /*SELF-REL PTR TO USER RUNDOWN SERVICE
    FILL_1 longword fill prefix PLVDEF tag $$;          /*UNUSED, RESERVED FOR FUTURE USE
    RMS longword unsigned;                              /*SELF-REL PTR TO RMS SERVICES DISPATCHER
    CHECK longword unsigned;                            /*LONGWORD USED TO CHECK VIRTUAL ADDRESS
                                                        /*LOCATION OF VECTOR
end PLVDEF;

end_module $PLVDEF;
```

```
module $PQLDEF;
/*+
/* PROCESS QUOTA LIST CODES
/*-
constant(
      LISTEND                                   /*LIST END CODE (MUST BE FIRST)
    , ASTLM                                     /*AST LIMIT
    , BIOLM                                     /*BUFFERED I/O LIMIT
    , BYTLM                                     /*BYTE LIMIT FOR BUFFERED I/O
    , CPULM                                     /*CPU TIME LIMIT
    , DIOLM                                     /*DIRECT I/O LIMIT
    , FILLM                                     /*OPEN FILE LIMIT
    , PGFLQUOTA                                 /*PAGING FILE QUOTA
    , PRCLM                                     /*SUB-PROCESS LIMIT
    , TQELM                                     /*TIMER QUEUE ENTRY LIMIT
    , WSQUOTA                                   /*WORKING SET QUOTA
    , WSDEFAULT                                 /*WORKING SET DEFAULT
    , ENQLM                                     /*ENQUEUE LIMIT
    , WSEXTENT                                  /*WORKING SET EXTENT LIMIT
    , JTQUOTA                                   /*JOB-WIDE LOGICAL NAME TABLE CREATION QUOTA
    , "LENGTH"                                  /*NUMBER OF QUOTAS (MUST BE LAST)
    ) equals 0 increment 1  prefix PQL tag $;

end_module $PQLDEF;
```

```
module $PRCDEF;
/*+
/* $CREPRC STATUS FLAGS AND ITEM CODES
/*-


aggregate PRCDEF  union prefix PRC$;
    PRCDEF_BITS structure fill;
        SSRWAIT bitfield mask;                  /* RESOURCE WAIT DISABLE
        SSFEXCU bitfield mask;                  /* SYSTEM SERVICE FAIL EXCEPTION MODE
        PSWAPM bitfield mask;                   /* PROCESS SWAP MODE
        NOACNT bitfield mask;                   /* ACCOUNTING MESSAGE DISABLE
        BATCH bitfield mask;                    /* BATCH INDICATOR
        HIBER bitfield mask;                    /* HIBERNATE BEFORE CALLING INITIAL IMAGE
        NOUAF bitfield mask;                    /* BYPASS LOGIN VERIFICATION FOR DETACHED PROC.
        NETWRK bitfield mask;                   /* NETWORK INDICATOR
        DISAWS bitfield mask;                   /* DISABLE WORKING SET ADJUST
        DETACH bitfield mask;                   /* DETACHED PROCESS
        INTER bitfield mask;                    /* INTERACTIVE INDICATOR
        IMGDMP bitfield mask;                   /* IMAGE DUMP REQUESTED
        CLISPEC bitfield mask;                  /* PASS CLI SPECIFICATIONS
        NOPASSWORD bitfield mask;               /* DON'T PROMPT FOR USERNAME AND PASSWORD
    end PRCDEF_BITS;
    PRCDEF_OBSOLETE structure fill;
        FILL_0 bitfield length 6 fill;
        LOGIN bitfield mask;                    /* BYPASS LOGIN VERIFICATION FOR DETACHED PROC.
    end PRCDEF_OBSOLETE;
end PRCDEF;

/*+
/*
/* Create Process Item List Data Identifier Definitions
/*
/* **** NOTE ****
/*
/*        New items must always be added at the END of the list so that
/*        users will not have to relink or reassemble.
/*
/*-

constant(
      LISTEND                           /* End of list (must be first code)
    , PGFLCHAR                          /* Page file characteristics
    , PGFLINDEX                         /* Page file index
    , INPUT_ATT                         /* SYS$INPUT attributes
    , OUTPUT_ATT                        /* SYS$OUTPUT attributes
    , ERROR_ATT                         /* SYS$ERROR attributes
    ) equals 0 increment 1  prefix PRC tag $;

end_module $PRCDEF;
```

```
module $PRVDEF;
/*+
/* PRIVILEGE BIT DEFINITIONS
/*
/*      Note that any privileges added here must also be reflected in the
/*      modules [VMSLIB.SRC]SETPRIV.MAR, [CLIUTL.SRC]SHOWPROC.B32,
/*      [RTL.SRC]LIBLEXICA.B32, and
/*      [CLD.SRC]DCLINT.CLD, MCRINT.CLD, MCRSET.CLD, RUN.CLD, and SET.CLD
/*      to completely add the new privilege.
/*
/*-


aggregate PRVDEF  union prefix PRV$;
    PRVDEF_BITS0 structure fill;
        CMKRNL bitfield mask;                   /* MAY CHANGE MODE TO KERNEL
        CMEXEC bitfield mask;                   /* MAY CHANGE MODE TO EXEC
/* ***** THE PRECEEDING TWO BITS MUST BE ADJACENT
/* ***** THE FOLLOWING TWO BITS MUST BE ADJACENT
        SYSNAM bitfield mask;                   /* MAY INSERT IN SYSTEM LOGICAL NAME TABLE
        GRPNAM bitfield mask;                   /* MAY INSERT IN GROUP LOGICAL NAME TABLE
/* ***** THE PRECEEDING TWO BITS MUST BE ADJACENT
        ALLSPOOL bitfield mask;                 /*MAY ALLOCATE SPOOLED DEVICE
        DETACH bitfield mask;                   /* MAY CREATE DETACHED PROCESSES
        DIAGNOSE bitfield mask;                 /* MAY DIAGNOSE DEVICES
        LOG_IO bitfield mask;                   /* MAY DO LOGICAL I/O
        GROUP bitfield mask;                    /* MAY AFFECT OTHER PROCESSES IN SAME GROUP
        NOACNT bitfield mask;                   /* MAY SUPPRESS ACCOUNTING MESSAGE
        PRMCEB bitfield mask;                   /* MAY CREATE PERMANENT COMMON EVENT CLUSTERS
        PRMMBX bitfield mask;                   /* MAY CREATE PERMANENT MAILBOX
        PSWAPM bitfield mask;                   /* MAY CHANGE PROCESS SWAP MODE
        SETPRI bitfield mask;                   /* MAY SET ANY PRIORITY VALUE
        SETPRV bitfield mask;                   /* MAY SET ANY PRIVILEGE BITS
        TMPMBX bitfield mask;                   /* MAY CREATE TEMPORARY MAILBOX
        WORLD bitfield mask;                    /* MAY AFFECT OTHER PROCESSES IN THE WORLD
        MOUNT bitfield mask;                    /* MAY EXECUTE MOUNT ACP FUNCTIONS
        OPER bitfield mask;                     /* OPERATOR PRIVILEGE
        EXQUOTA bitfield mask;                  /* MAY EXCEED QUOTAS
        NETMBX bitfield mask;                   /* MAY CREATE NETWORK DEVICE
        VOLPRO bitfield mask;                   /* MAY OVERRIDE VOLUME PROTECTION
        PHY_IO bitfield mask;                   /* MAY DO PHYSICAL I/O
        BUGCHK bitfield mask;                   /* MAY MAKE BUG CHECK ERROR LOG ENTRIES
        PRMGBL bitfield mask;                   /* MAY CREATE PERMANENT GLOBAL SECTIONS
        SYSGBL bitfield mask;                   /* MAY CREATE SYSTEM WIDE GLOBAL SECTIONS
        PFNMAP bitfield mask;                   /* MAY MAP TO SECTION BY PFN
        SHMEM bitfield mask;                    /* MAY ALLOCATE STRUCTURES IN SHARED MEMORY
        SYSPRV bitfield mask;                   /* ELIGIBLE FOR SYSTEM PROTECTION FIELD
        BYPASS bitfield mask;                   /* MAY BYPASS UIC BASED PROTECTION
        SYSLCK bitfield mask;                   /* MAY CREATE SYSTEM WIDE LOCKS
        SHARE bitfield mask;                    /* MAY ASSIGN CHANNEL TO NON-SHARED DEVICE
{       The following bits are in the second longword,
{       and thus, cannot have prv$m_ symbols...
        UPGRADE bitfield;                       /* May upgrade classification
        DOWNGRADE bitfield;                     /* May downgrade classification
        GRPPRV bitfield;                        /* Group access via system protection field
```

```
        READALL bitfield;                               /* Read access to everything
        TMPJNL bitfield;                                /* May create temporary journals
        PRMJNL bitfield;                                /* May create permanent journals
        SECURITY bitfield;                              /* May perform security functions
    end PRVDEF_BITS0;
                                                        /* ***** THE FOLLOWING TWO BITS MUST BE ADJACENT

    PRVDEF_BITS1 structure fill;
        FILL_1 bitfield length 9 fill prefix PRVDEF tag $$;/* SKIP 9
        ACNT bitfield mask;                             /* MAY SUPPRESS ACCOUNTING MESSAGES (NOACNT)
        FILL_2 bitfield length 3 fill prefix PRVDEF tag $$;/* SKIP 3
        ALTPRI bitfield mask;                           /* MAY SET ANY PRIORITY VALE (SETPRI)
    end PRVDEF_BITS1;
end PRVDEF;

end_module $PRVDEF;
```

```
module $PRTDEF;
/*+
/* PROTECTION FIELD DEFINITIONS
/*-


constant NA     equals (%B0000)  prefix PRT tag $C;     /* NO ACCESS
constant KR     equals (%B0011)  prefix PRT tag $C;     /* KERNEL READ ONLY
constant KW     equals (%B0010)  prefix PRT tag $C;     /* KERNEL WRITE
constant ER     equals (%B0111)  prefix PRT tag $C;     /* EXEC READ ONLY
constant EW     equals (%B0101)  prefix PRT tag $C;     /* EXEC WRITE
constant SR     equals (%B1011)  prefix PRT tag $C;     /* SUPER READ ONLY
constant SW     equals (%B1000)  prefix PRT tag $C;     /* SUPER WRITE
constant UR     equals (%B1111)  prefix PRT tag $C;     /* USER READ ONLY
constant UW     equals (%B0100)  prefix PRT tag $C;     /* USER WRITE

constant ERKW   equals (%B0110)  prefix PRT tag $C;     /* EXEC READ KERNEL WRITE
constant SRKW   equals (%B1010)  prefix PRT tag $C;     /* SUPER READ KERNEL WRITE
constant SREW   equals (%B1001)  prefix PRT tag $C;     /* SUPER READ EXEC WRITE
constant URKW   equals (%B1110)  prefix PRT tag $C;     /* USER READ KERNEL WRITE
constant UREW   equals (%B1101)  prefix PRT tag $C;     /* USER READ EXEC WRITE
constant URSW   equals (%B1100)  prefix PRT tag $C;     /* USER READ SUPER WRITE
constant RESERVED       equals 1  prefix PRT tag $C;    /* RESERVED

end_module $PRTDEF;
```

```
module $PRDEF;

/*+
/*  PROCESSOR REGISTER DEFINITIONS
/*-


constant KSP     equals 0  prefix PR tag $;          /*KERNEL STACK POINTER
constant ESP     equals 1  prefix PR tag $;          /*EXECUTIVE STACK POINTER
constant SSP     equals 2  prefix PR tag $;          /*SUPERVISOR STACK POINTER
constant USP     equals 3  prefix PR tag $;          /*USER STACK POINTER
constant ISP     equals 4  prefix PR tag $;          /*INTERRUPT STACK POINTER
constant P0BR    equals 8  prefix PR tag $;          /*P0 BASE REGISTER
constant P0LR    equals 9  prefix PR tag $;          /*P0 LIMIT REGISTER
constant P1BR    equals 10  prefix PR tag $;         /*P1 BASE REGISTER
constant P1LR    equals 11  prefix PR tag $;         /*P1 LIMIT REGISTER
constant SBR     equals 12  prefix PR tag $;         /*SYSTEM BASE REGISTER
constant SLR     equals 13  prefix PR tag $;         /*SYSTEM LIMIT REGISTER
constant PCBB    equals 16  prefix PR tag $;         /*PROCESS CONTROL BLOCK BASE
constant SCBB    equals 17  prefix PR tag $;         /*SYSTEM CONTROL BLOCK BASE
constant IPL     equals 18  prefix PR tag $;         /*INTERRUPT PRIORITY LEVEL REGISTER
constant ASTLVL  equals 19  prefix PR tag $;         /*AST LEVEL REGISTER
constant SIRR    equals 20  prefix PR tag $;         /*SOFTWARE INTERRUPT REQUEST REGISTER
constant SISR    equals 21  prefix PR tag $;         /*SOFTWARE INTERRUPT SUMMARY REGISTER
constant ICCS    equals 24  prefix PR tag $;         /* INTERVAL CLOCK CONTROL STATUS REGISTER
constant RXCS    equals 32  prefix PR tag $;         /* CONSOLE RECIEVER CONTROL STATUS REGISTER
constant RXDB    equals 33  prefix PR tag $;         /* CONSOLE RECEIVER DATA BUFFER REGISTER
constant TXCS    equals 34  prefix PR tag $;         /* CONSOLE TRANSMIT CONTROL STATUS REGISTER
constant TXDB    equals 35  prefix PR tag $;         /* CONSOLE TRANSMIT DATA BUFFER REGISTER
constant MAPEN   equals 56  prefix PR tag $;         /* MAPPING ENABLE REGISTER
constant TBIA    equals 57  prefix PR tag $;         /* TRANSLATION BUFFER INVALIDATE ALL
constant TBIS    equals 58  prefix PR tag $;         /* TRANSLATION BUFFER INVALIDATE SINGLE
constant SID     equals 62  prefix PR tag $;         /* SYSTEM IDENTIFICATION REGISTER
constant TBCHK   equals 63  prefix PR tag $;         /* TRANSLATION BUFFER VALID CHECK

aggregate PRDEF  union prefix PR$;
    PRDEF_BITS structure fill;
        SID_SN bitfield length 12;                   /* SERIAL NUMBER FIELD
        SID_PL bitfield length 3;                    /* PLANT ID
        SID_ECO bitfield length 9;                   /* ECO LEVEL
        SID_TYPE bitfield length 8;                  /* CPU TYPE CODE
    end PRDEF_BITS;
                                                     /* SYSTEM ID REGISTER CPU TYPES
    constant TYP780     equals 1  prefix PR$_S tag ID;  /* VAX 11/780
    constant TYP750     equals 2  prefix PR$_S tag ID;  /* VAX 11/750
    constant TYP730     equals 3  prefix PR$_S tag ID;  /* VAX 11/730
    constant TYP790     equals 4  prefix PR$_S tag ID;  /* VAX 11/790
    constant TYP8SS     equals 5  prefix PR$_S TAG ID;  /* Scorpio for now
    constant TYP8NN     equals 6  prefix PR$_S TAG ID;  /* Nautilus for now
    constant TYPMAX     equals 8  prefix PR$_S tag ID;  /* MAX LEGAL CPU TYPE
                                                     /* Micro-VAX cpus
    constant TYPUV1     equals 7  prefix PR$_S tag ID;  /* Micro-VAX UV1
    constant TYPUV2     equals 8  prefix PR$_S tag ID;  /* Micro-VAX UV2
                                                     /*VAX 11/780 IPR'S:
    constant WCSA       equals 44  prefix PR tag $;  /* WCS ADDRESS REGISTER
    constant WCSD       equals 45  prefix PR tag $;  /* WCS DATA REGISTER
```

```
    constant SBIFS      equals 48  prefix PR tag $;     /* SBI FAULT STATUS REGISTER
    constant SBIS       equals 49  prefix PR tag $;     /* SBI SILO REGISTER
    constant SBISC      equals 50  prefix PR tag $;     /* SBI COMPARATOR REGISTER
    constant SBIMT      equals 51  prefix PR tag $;     /* SBI MAINTENANCE REGISTER
    constant SBIER      equals 52  prefix PR tag $;     /* SBI ERROR REGISTER
    constant SBITA      equals 53  prefix PR tag $;     /* SBI TIMEOUT ADDRESS REGISTER
    constant SBIQC      equals 54  prefix PR tag $;     /* SBI QUADWORD CLEAR REGISTER
                                                        /*END OF VAX 11/780-SPECIFIC IPR'S
                                                        /*VAX 11/750 AND 11/730 IPR'S:
    constant CMIERR     equals 23  prefix PR tag $;     /* CMI ERROR SUMMARY REGISTER
    constant CSRS       equals 28  prefix PR tag $;     /* CONSOLE BLK STORE RCV STATUS
    constant CSRD       equals 29  prefix PR tag $;     /* CONSOLE BLK STORE RCV DATA
    constant CSTS       equals 30  prefix PR tag $;     /* CONSOLE BLK STORE XMIT STATUS
    constant CSTD       equals 31  prefix PR tag $;     /* CONSOLE BLK STORE XMIT DATA
    constant TBDR       equals 36  prefix PR tag $;     /* TB DISABLE REGISTER
    constant CADR       equals 37  prefix PR tag $;     /* CACHE DISABLE REGISTER
    constant MCESR      equals 38  prefix PR tag $;     /* MACHINE CHECK ERROR SUMMARY REG
    constant CAER       equals 39  prefix PR tag $;     /* CACHE ERROR REGISTER
    constant UBRESET    equals 55  prefix PR tag $;     /* UNIBUS I/O RESET REGISTER
                                                        /*END OF 11/750 AND 11/730 IPR'S
                                                        /*VAX 11/790 PROCESSOR-SPECIFIC IPRS
    constant PAMACC     equals 64  prefix PR tag $;     /* PAMM ACCESS
    constant PAMLOC     equals 65  prefix PR tag $;     /* PAMM LOCATION
    constant CSWP       equals 66  prefix PR tag $;     /* CACHE SWEEP REGISTER
    constant MDECC      equals 67  prefix PR tag $;     /* MBOX DATA ECC REGISTER
    constant MENA       equals 68  prefix PR tag $;     /* MBOX ERROR ENABLE REGISTER
    constant MDCTL      equals 69  prefix PR tag $;     /* MBOX DATA CONTROL REGISTER
    constant MCCTL      equals 70  prefix PR tag $;     /* MBOX MCC CONTROL REGISTER
    constant MERG       equals 71  prefix PR tag $;     /* MBOX ERROR GENERATOR REGISTER
    constant CRBT       equals 72  prefix PR tag $;     /* CONSOLE REBOOT
    constant DFI        equals 73  prefix PR tag $;     /* DIAGNOSTIC FAULT INSERTION
    constant EHSR       equals 74  prefix PR tag $;     /* ERROR HANDLING STATUS REGISTER
    constant ACCS790    equals 75  prefix PR tag $;     /* ACCELERATOR STATUS REGISTER
    constant STXCS      equals 76  prefix PR tag $;     /* CONSOLE STORAGE CONTROL REG
    constant STXDB      equals 77  prefix PR tag $;     /* CONSOLE STORAGE DATA REGISTER
    constant LSPA       equals 78  prefix PR tag $;     /* SCRATCHPAD ADDRESS
    constant RSPD       equals 79  prefix PR tag $;     /* SCRATCHPAD DATA
                                                        /*END OF 11/790 PROCESSOR-SPECIFIC IPRS
end PRDEF;

end_module $PRDEF;
```

```
module $PR730DEF;

{*+
{*  11/730-Specific Processor Register Definitions
{*-


constant NICR   equals 25  prefix PR730 tag $;  /* INTERVAL CLOCK NEXT INTERVAL REGISTER
constant ICR    equals 26  prefix PR730 tag $;  /* INTERVAL CLOCK INTERVAL COUNT REGISTER
constant TODR   equals 27  prefix PR730 tag $;  /* TIME OF DAY REGISTER
constant ACCS   equals 40  prefix PR730 tag $;  /* ACCELERATOR CONTROL STATUS REGISTER
constant ACCR   equals 41  prefix PR730 tag $;  /* ACCELERATOR RESERVED
constant PME    equals 61  prefix PR730 tag $;  /* PERFORMANCE MONITOR ENABLE

constant CMIERR equals 23  prefix PR730 tag $;  /* CMI ERROR SUMMARY REGISTER
constant CSRS   equals 28  prefix PR730 tag $;  /* CONSOLE BLK STORE RCV STATUS
constant CSRD   equals 29  prefix PR730 tag $;  /* CONSOLE BLK STORE RCV DATA
constant CSTS   equals 30  prefix PR730 tag $;  /* CONSOLE BLK STORE XMIT STATUS
constant CSTD   equals 31  prefix PR730 tag $;  /* CONSOLE BLK STORE XMIT DATA
constant TBDR   equals 36  prefix PR730 tag $;  /* TB DISABLE REGISTER
constant CADR   equals 37  prefix PR730 tag $;  /* CACHE DISABLE REGISTER
constant MCESR  equals 38  prefix PR730 tag $;  /* MACHINE CHECK ERROR SUMMARY REG
constant CAER   equals 39  prefix PR730 tag $;  /* CACHE ERROR REGISTER
constant UBRESET equals 55  prefix PR730 tag $; /* UNIBUS I/O RESET REGISTER

end_module $PR730DEF;
```

```
module $PR750DEF;

{*+
{*  11/750-Specific Processor Register Definitions
{*-


constant NICR   equals 25  prefix PR750 tag $;  /* INTERVAL CLOCK NEXT INTERVAL REGISTER
constant ICR    equals 26  prefix PR750 tag $;  /* INTERVAL CLOCK INTERVAL COUNT REGISTER
constant TODR   equals 27  prefix PR750 tag $;  /* TIME OF DAY REGISTER
constant ACCS   equals 40  prefix PR750 tag $;  /* ACCELERATOR CONTROL STATUS REGISTER
constant ACCR   equals 41  prefix PR750 tag $;  /* ACCELERATOR RESERVED
constant PME    equals 61  prefix PR750 tag $;  /* PERFORMANCE MONITOR ENABLE

constant CMIERR equals 23  prefix PR750 tag $;  /* CMI ERROR SUMMARY REGISTER
constant CSRS   equals 28  prefix PR750 tag $;  /* CONSOLE BLK STORE RCV STATUS
constant CSRD   equals 29  prefix PR750 tag $;  /* CONSOLE BLK STORE RCV DATA
constant CSTS   equals 30  prefix PR750 tag $;  /* CONSOLE BLK STORE XMIT STATUS
constant CSTD   equals 31  prefix PR750 tag $;  /* CONSOLE BLK STORE XMIT DATA
constant TBDR   equals 36  prefix PR750 tag $;  /* TB DISABLE REGISTER
constant CADR   equals 37  prefix PR750 tag $;  /* CACHE DISABLE REGISTER
constant MCESR  equals 38  prefix PR750 tag $;  /* MACHINE CHECK ERROR SUMMARY REG
constant CAER   equals 39  prefix PR750 tag $;  /* CACHE ERROR REGISTER
constant UBRESET equals 55  prefix PR750 tag $; /* UNIBUS I/O RESET REGISTER

end_module $PR750DEF;
```

```
module $PR780DEF;

{*+
{*  11/780-Specific Processor Register Definitions
{*-


constant NICR   equals 25  prefix PR780 tag $;  /* INTERVAL CLOCK NEXT INTERVAL REGISTER
constant ICR    equals 26  prefix PR780 tag $;  /* INTERVAL CLOCK INTERVAL COUNT REGISTER
constant TODR   equals 27  prefix PR780 tag $;  /* TIME OF DAY REGISTER
constant ACCS   equals 40  prefix PR780 tag $;  /* ACCELERATOR CONTROL STATUS REGISTER
constant ACCR   equals 41  prefix PR780 tag $;  /* ACCELERATOR RESERVED
constant PME    equals 61  prefix PR780 tag $;  /* PERFORMANCE MONITOR ENABLE

constant WCSA   equals 44  prefix PR780 tag $;  /* WCS ADDRESS REGISTER
constant WCSD   equals 45  prefix PR780 tag $;  /* WCS DATA REGISTER
constant SBIFS  equals 48  prefix PR780 tag $;  /* SBI FAULT STATUS REGISTER
constant SBIS   equals 49  prefix PR780 tag $;  /* SBI SILO REGISTER
constant SBISC  equals 50  prefix PR780 tag $;  /* SBI COMPARATOR REGISTER
constant SBIMT  equals 51  prefix PR780 tag $;  /* SBI MAINTENANCE REGISTER
constant SBIER  equals 52  prefix PR780 tag $;  /* SBI ERROR REGISTER
constant SBITA  equals 53  prefix PR780 tag $;  /* SBI TIMEOUT ADDRESS REGISTER
constant SBIQC  equals 54  prefix PR780 tag $;  /* SBI QUADWORD CLEAR REGISTER

end_module $PR780DEF;
```

```
module $PR790DEF;

{*+
{*  11/790-Specific Processor Register Definitions
{*-


constant NICR    equals 25  prefix PR790 tag $;  /* INTERVAL CLOCK NEXT INTERVAL REGISTER
constant ICR     equals 26  prefix PR790 tag $;  /* INTERVAL CLOCK INTERVAL COUNT REGISTER
constant TODR    equals 27  prefix PR790 tag $;  /* TIME OF DAY REGISTER
constant ACCS    equals 40  prefix PR790 tag $;  /* ACCELERATOR CONTROL STATUS REGISTER
constant ACCR    equals 41  prefix PR790 tag $;  /* ACCELERATOR RESERVED
constant PME     equals 61  prefix PR790 tag $;  /* PERFORMANCE MONITOR ENABLE

constant PAMACC  equals 64  prefix PR790 tag $;  /* PAMM ACCESS
constant PAMLOC  equals 65  prefix PR790 tag $;  /* PAMM LOCATION
constant CSWP    equals 66  prefix PR790 tag $;  /* CACHE SWEEP REGISTER
constant MDECC   equals 67  prefix PR790 tag $;  /* MBOX DATA ECC REGISTER
constant MENA    equals 68  prefix PR790 tag $;  /* MBOX ERROR ENABLE REGISTER
constant MDCTL   equals 69  prefix PR790 tag $;  /* MBOX DATA CONTROL REGISTER
constant MCCTL   equals 70  prefix PR790 tag $;  /* MBOX MCC CONTROL REGISTER
constant MERG    equals 71  prefix PR790 tag $;  /* MBOX ERROR GENERATOR REGISTER
constant CRBT    equals 72  prefix PR790 tag $;  /* CONSOLE REBOOT
constant DFI     equals 73  prefix PR790 tag $;  /* DIAGNOSTIC FAULT INSERTION
constant EHSR    equals 74  prefix PR790 tag $;  /* ERROR HANDLING STATUS REGISTER
constant ACCS790 equals 75  prefix PR790 tag $;  /* ACCELERATOR STATUS REGISTER
constant STXCS   equals 76  prefix PR790 tag $;  /* CONSOLE STORAGE CONTROL REG
constant STXDB   equals 77  prefix PR790 tag $;  /* CONSOLE STORAGE DATA REGISTER
constant LSPA    equals 78  prefix PR790 tag $;  /* SCRATCHPAD ADDRESS
constant RSPD    equals 79  prefix PR790 tag $;  /* SCRATCHPAD DATA

end_module $PR790DEF;
```

```
module $PRUV1DEF;

{*+
{*  Micro-VAX I Processor-specific Register Definitions
{*-


constant CADR   equals 37  prefix PRUV1 tag $;  /* CACHE DISABLE REGISTER
constant MCESR  equals 38  prefix PRUV1 tag $;  /* MACHINE CHECK ERROR SUMMARY REG
constant IORESET equals 55  prefix PRUV1 tag $; /* INITIALIZE BUS REGISTER

end_module $PRUV1DEF;
```

```
module $PRUV2DEF;

{*+
{*  Micro-VAX II Processor-specific Register Definitions
{*-


constant SAVISP equals 41  prefix PRUV2 tag $;  /* CONSOLE SAVED INTERRUPT STACK POINTER
constant SAVPC  equals 42  prefix PRUV2 tag $;  /* CONSOLE SAVED PC REGISTER
constant SAVPSL equals 43  prefix PRUV2 tag $;  /* CONSOLE SAVED PSL REGISTER
constant IORESET equals 55  prefix PRUV2 tag $; /* INITIALIZE BUS REGISTER

end_module $PRUV2DEF;
```

```
module $PR8NNDEF;

{*+
{*  11/8NN-Specific Processor Register Definitions
{*-

constant NICR   equals 25   prefix PR8NN tag $; /* Next Interval Count Register
constant ICR    equals 26   prefix PR8NN tag $; /* Interval Counter Register
constant TODR   equals 27   prefix PR8NN tag $; /* Time of Year
constant PME    equals 61   prefix PR8NN tag $; /* Performance Monitor Enable

constant MCSTS  equals 38   prefix PR8NN tag $; /* Machine Check Status Register
constant NMION  equals 128  prefix PR8NN tag $; /* NMI Interrupt Enable
constant INOP   equals 129  prefix PR8NN tag $; /* Interrupt Other Processor
constant NMIFSR equals 130  prefix PR8NN tag $; /* NMI Fault/Status Register
constant NMISILO equals 131 prefix PR8NN tag $; /* NMI Bus Silo
constant NMIEAR equals 132  prefix PR8NN tag $; /* NMI Error Address Register
constant CCR    equals 133  prefix PR8NN tag $; /* Cache Control Register
constant REVR1  equals 134  prefix PR8NN tag $; /* Revision Register #1
constant REVR2  equals 135  prefix PR8NN tag $; /* Revision Register #2

end_module $PR8NNDEF;
```

```
module $PR8SSDEF;

{*+
{*  11/8SS-Specific Processor Register Definitions
{*-


constant IPIR   equals 22  prefix PR8SS tag $;  /* Interprocessor Interrupt Reg.
constant NICR   equals 25  prefix PR8SS tag $;  /* Interval Clock Next Interval Register
constant ICR    equals 26  prefix PR8SS tag $;  /* Interval Clock Interval Count Register
constant TODR   equals 27  prefix PR8SS tag $;  /* Time Of Day Register
constant TBDR   equals 36  prefix PR8SS tag $;  /* Translation Buffer Disable Register
constant CADR   equals 37  prefix PR8SS tag $;  /* Cache Disable Register
constant MCESR  equals 38  prefix PR8SS tag $;  /* Machine Check Error Summary Register
constant ACCS   equals 40  prefix PR8SS tag $;  /* Floating Point Accellerator Register
constant WCSA   equals 44  prefix PR8SS tag $;  /* WCS Address Register
constant WCSD   equals 45  prefix PR8SS tag $;  /* WCS Data Register
constant WCSC   equals 46  prefix PR8SS tag $;  /* WCS Cam Register
constant PME    equals 61  prefix PR8SS tag $;  /* Performance Monitor Enable

constant RXCS1  equals 80  prefix PR8SS tag $;  /* Serial Line 1 Receive CSR
constant RXDB1  equals 81  prefix PR8SS tag $;  /* Serial Line 1 Receive Data Buffer
constant TXCS1  equals 82  prefix PR8SS tag $;  /* Serial Line 1 Transmit CSR
constant TXDB1  equals 83  prefix PR8SS tag $;  /* Serial Line 1 Transmit Data Buffer
constant RXCS2  equals 84  prefix PR8SS tag $;  /* Serial Line 2 Receive CSR
constant RXDB2  equals 85  prefix PR8SS tag $;  /* Serial Line 2 Receive Data Buffer
constant TXCS2  equals 86  prefix PR8SS tag $;  /* Serial Line 2 Transmit CSR
constant TXDB2  equals 87  prefix PR8SS tag $;  /* Serial Line 2 Transmit Data Buffer
constant RXCS3  equals 88  prefix PR8SS tag $;  /* Serial Line 3 Receive CSR
constant RXDB3  equals 89  prefix PR8SS tag $;  /* Serial Line 3 Receive Data Buffer
constant TXCS3  equals 90  prefix PR8SS tag $;  /* Serial Line 3 Transmit CSR
constant TXDB3  equals 91  prefix PR8SS tag $;  /* Serial Line 3 Transmit Data Buffer

constant RXCD   equals 92  prefix PR8SS tag $;  /* Receive Console Data Register
constant CACHEX equals 93  prefix PR8SS tag $;  /* Cache Invalidate Register
constant BINID  equals 94  prefix PR8SS tag $;  /* BI Node ID Register
constant BIINIT equals 95  prefix PR8SS tag $;  /* BI Init Nodes Register

aggregate PR8SSDEF  union prefix PR8SS$;
    PR8SSSID_BITS structure fill;                       /* Read only SID register
        SID_UCREV  bitfield length 8;                   /* Ucode Revision Level
        SID_SECP   bitfield  mask;                      /* Secondary Patch Bit
        SID_PATREV bitfield length 10;                  /* Patch Rev Level
        SID_CPUREV bitfield length 5;                   /* CPU Rev level
        SID_TYPE   bitfield length 8;                   /* CPU Type Code
    end PR8SSSID_BITS;

    PR8SSRXCS_BITS structure fill;                      /* Console RCV CSR
        FILL_1 bitfield length 6 fill prefix PR8SS tag $$;/*
        RXCS_IE   bitfield  mask;                       /* Interrupt Enable
        RXCS_DONE bitfield  mask;                       /* 1=> Char. received
    end PR8SSRXCS_BITS;

    PR8SSRXDB_BITS structure fill;                      /* Console RCV Data Register
        RXDB_DATA   bitfield length 8;                  /* Received Data
        FILL_2 bitfield length 7 fill prefix PR8SS tag $$;/*
```

```
    RXDB_ERR    bitfield  mask;                     /* Error
end PR8SSRXDB_BITS;

PR8SSTXCS_BITS structure fill;                      /* Console Transmit CSR
    FILL_3 bitfield length 6 fill prefix PR8SS tag $$;/*
    TXCS_IE   bitfield  mask;                       /* Interrupt Enable
    TXCS_RDY  bitfield  mask;                       /* Ready
    TXCS_BRE  bitfield  mask;                       /* (WO) Baud Rate Enable
    FILL_4 bitfield length 1 fill prefix PR8SS tag $$;/*
    TXCS_BAUD bitfield length 3;                    /* Baud Rate
                                                    /* Values to set baud rates
    constant BAUD300   equals 0  prefix PR8SS tag $;/* Baud Rate of 300
    constant BAUD600   equals 1  prefix PR8SS tag $;/* Baud Rate of 600
    constant BAUD1200  equals 2  prefix PR8SS tag $;/* Baud Rate of 1200
    constant BAUD2400  equals 3  prefix PR8SS tag $;/* Baud Rate of 2400
    constant BAUD4800  equals 4  prefix PR8SS tag $;/* Baud Rate of 4800
    constant BAUD9600  equals 5  prefix PR8SS tag $;/* Baud Rate of 9600
    constant BAUD19200 equals 6  prefix PR8SS tag $;/* Baud Rate of 19200
    constant BAUD38400 equals 7  prefix PR8SS tag $;/* Baud Rate of 38400

end PR8SSTXCS_BITS;

PR8SSTXDB_BITS structure fill;                      /* Console Transmit Data Register
    TXDB_DATA   bitfield length 8;                  /* Data to Transmit
    TXDB_ID     bitfield length 4;                  /* ID - Destination of
                                                    /*  transmitted data -
                                                    /*  0=>UART0, F=>Console
                                                    /*  command
                                                    /* Possible Console Commands
    constant BOOTCPU equals 2  prefix PR8SS tag $;  /* Boot CPU Command
    constant CLRWARM equals 3  prefix PR8SS tag $;  /* Clear Warm-start Flag
    constant CLRCOLD equals 4  prefix PR8SS tag $;  /* Clear Cold-start Flag
end PR8SSTXDB_BITS;

PR8SSCADR_BITS structure fill;                      /* Cache Disable Register
    CADR_D  bitfield  mask;                         /* Disable Cache
    CADR_H  bitfield  mask;                         /* Force 100% Cache Hits
end PR8SSCADR_BITS;

PR8SSWCSA_BITS structure fill;                      /* WCS (Patch) Address Reg
    WCSA_DATA   bitfield length 8;                  /* High Order Data Bits
    FILL_4 bitfield length 8 fill prefix PR8SS tag $$;/*
    WCSA_RAMADR bitfield length 16;                 /* Ram Address
end PR8SSWCSA_BITS;

PR8SSWCSC_BITS structure fill;                      /* WCS (Patch) CAM Reg
    FILL_5 bitfield length 8 fill prefix PR8SS tag $$;/*
    WCSC_CAMADR bitfield length 8;                  /* Cam Address
    WCSC_ROMADR bitfield length 16;                 /* Rom Address
end PR8SSWCSC_BITS;

PR8SSRXCD_BITS structure fill;                      /* Receive Console Data Register
    RXCD_DATA   bitfield length 8;                  /* Received Data
    RXCD_NODEID bitfield length 4;                  /* Sender's Node ID
    FILL_6 bitfield length 3 fill prefix PR8SS tag $$;/*
    RXCD_BSY    bitfield  mask;                     /* Set=>Data has been received
```

```
    end PR8SSRXCD_BITS;

    PR8SSCACHEX_BITS structure fill;                    /* Cache Invalidate Register
        FILL_7 bitfield length 9 fill prefix PR8SS tag $$;/*
        CACHEX_PFN bitfield length 21;                  /* Physical Page Number
    end PR8SSCACHEX_BITS;

    PR8SSBINID_BITS structure fill;                     /* BI Node ID Register
        BINID_NID bitfield length 4;                    /* BI Node ID this node
    end PR8SSBINID_BITS;

end PR8SSDEF;
end_module $PR8SSDEF;
```

```
module $PSLDEF;

/*+
/* PROCESSOR STATUS LONGWORD MASK AND FIELD DEFINITIONS
/*-



aggregate PSLDEF  union prefix PSL$;
    PSLDEF_BITS structure fill;
        C bitfield mask;                                    /* Carry
        V bitfield mask;                                    /* oVerflow
        Z bitfield mask;                                    /* Zero
        N bitfield mask;                                    /* Negative
        TBIT bitfield mask;                                 /* TBIT ENABLE
        IV bitfield mask;                                   /* INTEGER OVERFLOW
        FU bitfield mask;                                   /* FLOATING UNDEFINED
        DV bitfield mask;                                   /* DIVIDE BY ZERO
        FILL_1 bitfield length 8 fill prefix PSLDEF tag $$;/*
        IPL bitfield mask length 5;                         /* INTERRUPT PRIORITY LEVEL
        FILL_2 bitfield fill prefix PSLDEF tag $$;          /*
        PRVMOD bitfield mask length 2;                      /* PREVIOUS PROCESSOR MODE
        CURMOD bitfield mask length 2;                      /* CURRENT PROCESSOR MODE
        IS bitfield mask;                                   /* INTERRUPT STACK BIT
        FPD bitfield mask;                                  /* FIRST PART DONE
        FILL_3 bitfield length 2 fill prefix PSLDEF tag $$;/* MUST BE ZERO
        TP bitfield mask;                                   /* TRACE TRAP PENDING
        CM bitfield mask;                                   /* COMPATIBILITY MODE BIT AND MASK
    end PSLDEF_BITS;
/*
/* MODE SYMBOL DEFINITIONS
/*
    constant(
          KERNEL                                            /* KERNEL MODE
        , EXEC                                              /* EXEC MODE
        , SUPER                                             /* SUPERVISOR MODE
        , USER                                              /* USER MODE
        ) equals 0 increment 1 prefix PSL tag $C;

    constant SAFBITS    equals
        ( (( - (PSL$M_TP!
        PSL$M_CM!
        PSL$M_FPD)a(-16))  ) - 1 )
     prefix PSL tag $M;

end PSLDEF;

end_module $PSLDEF;
```

STARDEFFL
SDL

OPCDEF
SDL

SCRDEF
SDL

OPDEF
SDL

SRMDEF
SDL

STARDEFMP
SDL

STARDEFQZ
SDL

STARDEFAE
SDL