N 11

```
VVV         VVV  MMM         MMM     SSSSSSSSSSSS  LLL                 IIIIIIIII     BBBBBBBBBBBB
VVV         VVV  MMM         MMM     SSSSSSSSSSSS  LLL                 IIIIIIIII     BBBBBBBBBBBB
VVV         VVV  MMM         MMM     SSSSSSSSSSSS  LLL                 IIIIIIIII     BBBBBBBBBBBB
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        BBBBBBBBBBBB
VVV         VVV  MMM         MMM     SSSSSSSSS     LLL                    III        BBBBBBBBBBBB
VVV         VVV  MMM         MMM     SSSSSSSSS     LLL                    III        BBBBBBBBBBBB
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  SSSSSSSSSSSS     LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB
      VVV        MMM         MMM  SSSSSSSSSSSS     LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB
      VVV        MMM         MMM  SSSSSSSSSSSS     LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB
```

```
**FILE**ID**UTLDEFM                                    L 14                                                     **
```

```
UU      UU  TTTTTTTTTT  LL          DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF  MM      MM
UU      UU  TTTTTTTTTT  LL          DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF  MM      MM
UU      UU      TT      LL          DD      DD  EE          FF          MMMM  MMMM
UU      UU      TT      LL          DD      DD  EE          FF          MMMM  MMMM
UU      UU      TT      LL          DD      DD  EE          FF          MM  MM  MM
UU      UU      TT      LL          DD      DD  EE          FF          MM  MM  MM
UU      UU      TT      LL          DD      DD  EEEEEEEE    FFFFFFFF    MM      MM
UU      UU      TT      LL          DD      DD  EEEEEEEE    FFFFFFFF    MM      MM
UU      UU      TT      LL          DD      DD  EE          FF          MM      MM
UU      UU      TT      LL          DD      DD  EE          FF          MM      MM
UU      UU      TT      LL          DD      DD  EE          FF          MM      MM      ....
UU      UU      TT      LL          DD      DD  EE          FF          MM      MM      ....
UUUUUUUUUU      TT      LLLLLLLLLL  DDDDDDDD    EEEEEEEEEE  FF          MM      MM      ....
UUUUUUUUUU      TT      LLLLLLLLLL  DDDDDDDD    EEEEEEEEEE  FF          MM      MM      ....

MM      MM    AAAAAA    RRRRRRRR
MM      MM    AAAAAA    RRRRRRRR
MMMM  MMMM  AA      AA  RR      RR
MMMM  MMMM  AA      AA  RR      RR
MM  MM  MM  AA      AA  RR      RR
MM  MM  MM  AA      AA  RR      RR
MM      MM  AA      AA  RRRRRRRR
MM      MM  AA      AA  RRRRRRRR
MM      MM  AAAAAAAAAA  RR  RR
MM      MM  AAAAAAAAAA  RR  RR
MM      MM  AA      AA  RR    RR
MM      MM  AA      AA  RR    RR
MM      MM  AA      AA  RR      RR
MM      MM  AA      AA  RR      RR
```

```
        .NLIST
;
; 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.                 *
;*                                                                          *
;*                                                                          *
;****************************************************************************
;
; Modified by:
;
;       V03-002 LJK0174         Lawrence J. Kenah       2-Jul-1982
;               Add .ENABLE SUPPRESSION to $DEFINI and .DISABLE SUPPRESSION
;               to $DEFEND when the GLOBAL parameter is not supplied.
;
;       V03-001 LJK0171         Lawrence J. Kenah       18-Jun-1982
;               Add .NOCROSS to $DEFINI and .CROSS to $DEFEND
;
;       V02-003 BLS0130         Benn Schreiber   4-Jan-1982
;               Add $SHR_MSGDEF macro
;
;       V02.02  HJ0002          Herb Jacobs     14-Aug-1980
;               New versions of $DEF, $EQU, $EQULST, $VIELD for improved
;               performance at assembly time.
;
;       V01.01  RN0001          R. Newland      9-Jul-1979
;               Save local symbol block when saving psect information
;               in $DEFINI and $OFFSET.  SPR #11-24166
;
;+
; $DEF - DEFINE A SYMBOL
;       SYM = SYMBOL NAME TO BE DEFINED
;       ALLOC = ASSEMBLER ALLOCATION DIRECTIVE, .BLKL, .BLKB, ETC
;       SIZ = AMOUNT OF STORAGE TO ALLOCATE IN UNITS SPECIFIED BY ALLOC
;-
;+
; $EQU - EQUATE A SYMBOL
;       SYM = SYMBOL NAME TO DEFINE
;       VAL = VALUE TO ASSOCIATE WITH THE SYMBOL
;-
```

```
;+
; $GBLINI - INITIALIZE THE GLOBAL/LOCAL DEFINITION SWITCH
;       GBL     = "GLOBAL", "LOCAL", OR NULL.
;                 IF THIS PARAMETER IS "GLOBAL"
;                 GLOBAL DEFINITIONS ARE GENERATED.  OTHERWISE
;                 LOCAL DEFINITIONS ARE GENERATED.
;-
        .MACRO  $GBLINI GBL=LOCAL
        .IF     IDN <GBL> <GLOBAL>
        .MACRO  $DEF    SYM,ALLOC,SIZ
        .IIF    NB,SYM, SYM::
        .IIF    NB,ALLOC,       ALLOC   SIZ
        .ENDM   $DEF
        .MACRO  $EQU    SYM,VAL
        SYM==VAL
        .ENDM   $EQU
        .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
        SIZ...=1
        .IIF    NB,SIZ, SIZ...=SIZ
        .IF     NB,SYM
        MOD'SEP'V_'SYM==BIT...
        .IIF    NB,SIZ, MOD'SEP'S_'SYM==SIZ
        .IIF    NB,MSK, MOD'SEP'M_'SYM==<<<1@SIZ...>-1>@BIT...>
        .ENDC
        BIT...=BIT...+SIZ...
        .ENDM   $VIELD1
        .IFF
        .IIF    DIF <GBL> <LOCAL>,.ERROR ;ARG MUST BE "GLOBAL"," OCAL",OR NULL;
        .MACRO  $DEF    SYM,ALLOC,SIZ
        .IIF    NB,SYM, SYM:
        .IIF    NB,ALLOC,       ALLOC   SIZ
        .ENDM   $DEF
        .MACRO  $EQU    SYM,VAL
        SYM=VAL
        .ENDM   $EQU
        .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
        SIZ...=1
        .IIF    NB,SIZ, SIZ...=SIZ
        .IF     NB,SYM
        MOD'SEP'V_'SYM=BIT...
        .IIF    NB,SIZ, MOD'SEP'S_'SYM=SIZ
        .IIF    NB,MSK, MOD'SEP'M_'SYM=<<<1@SIZ...>-1>@BIT...>
        .ENDC
        BIT...=BIT...+SIZ...
        .ENDM   $VIELD1
        .ENDC
        .ENDM   $GBLINI
;+
; $DEFINI - INTIALIZE FOR A STRUCTURE DEFINTION
;       STRUC = STRUCTURE BEING DEFINED
;       GBL   = GLOBAL/LOCAL INDICATOR
;       DOT   = INITIAL ABSOLUTE LOCATION, DEFAULT = 0
;-
        .MACRO  $DEFINI STRUC,GBL,DOT=0
        .SAVE   LOCAL_BLOCK
        .NOCROSS
        .IIF    DIF <GBL> <GLOBAL>,.ENABLE      SUPPRESSION
        .PSECT  $ABS$,ABS
        $GBLINI GBL
        .=DOT
        .ENDM   $DEFINI
;+
; $DEFEND - END OF A STRUCTURE DEFINITION
;       STRUC = STRUCTURE NAME BEING DEFINED
;       GBL   = GLOBAL/LOCAL INDICATOR
;       SUF   = STRUCTURE NAME SUFFIX
;-
        .MACRO  $DEFEND STRUC,GBL,SUF=DEF
        .MACRO  $'STRUC'SUF A
        .ENDM   $'STRUC'SUF
        .IIF    DIF <GBL> <GLOBAL>,.DISABLE     SUPPRESSION
        .CROSS
        .RESTORE
        .ENDM   $DEFEND
```

C7

Th
ME

```
;+
; $VIELD1 - MACRO USED INTERNALLY BY $VIELD AND _VIELD
;       MOD = MODULE NAME
;       SEP = SEPARATOR CHARACTER BETWEEN MODULE NAME AND SYMBOL TYPE
;       SYM = SYMBOL NAME
;       SIZ = SIZE OF VIELD IN BITS
;       MSK = 'MASK'' IF MOD$M_SYM IS DESIRED
;-
;+
; $VIELD - DEFINE A LIST OF VIELDS - GLOBAL FORMAT
;       MOD = MODULE NAME
;       INIBIT = INITIAL BIT NUMBER FOR THE VIELDS
;       LIST = < <SYM1,SIZ1>,<SYM2,SIZ2>, ... <SYMN,SIZN> >
;               WHERE A DEGENERATE SYM,SIZ PAIR IS JUST ''SYM''
;-
        .MACRO  $VIELD  MOD,INIBIT,LIST
        .IIF    NB,INIBIT,      BIT...=INIBIT
        .IRP    L,<LIST>
        $VIELD1 MOD,$,L
        .ENDR
        .ENDM   $VIELD
```

```
;+
; _VIELD - DEFINE A LIST OF VIELDS - LOCAL FORMAT
;       MOD = MODULE NAME
;       INIBIT = INITIAL BIT NUMBER FOR THE VIELDS
;       LIST = < <SYM1,SIZ1>,<SYM2,SIZ2>, ... <SYMN,SIZN> >
;               WHERE A DEGENERATE SYM,SIZ PAIR IS JUST ''S'M''
;-
        .MACRO  _VIELD  MOD,INIBIT,LIST
        .IIF    NB,INIBIT,      BIT...=INIBIT
        .IRP    L,<LIST>
        $VIELD1 MOD,_,L
        .ENDR
        .ENDM   _VIELD
```

```
;+
; $EQULS1 - USED INTERNALLY BY $EQULST
;-
;+
; $EQULST - EQUATE A LIST OF SYMBOLS
;       PREFIX = PREFIX TO BE ATTACHED TO NAMES IN THE LIST
;       GBL    = GLOBAL/LOCAL INDICATOR
;       INIT   = INITIAL VALUE TO BE ASSIGNED TO 1ST SYMBOL
;       INCR   = INCREMENT TO BE ADDED TO VALUE ASSIGNED TO SYMBOLS
;       LIST   = LIST OF SYMBOLS TO BE ASSIGNED VALUES
;               ENTRIES ARE OF THE FORM
;               SYMBOL
;       OR
;               <SYMBOL,VALUE>
;
;               IF A SIMPLE LIST OF SYMBOLS IS SPECIFIED, THEY ARE ASSIGNED
;               VALUES STARTING WITH THE INIT VALUE, INCREMENTED BY THE
;               SPECIFIED INCR VALUE (DEFAULT = 1).
;
;               IF A LIST IS SPECIFIED WITH SYMBOL, VALUE PAIRS, THE
;               SPECIFIED SYMBOL IS EQUATED TO THE VALUE AND THE
;               INIT AND INCR PARAMETERS DO NOT APPLY.
;-
        .MACRO  $EQULST PREFIX,GBL,INIT,INCR=1,LIST
        $GBLINI GBL
        .MACRO  $EQULS1 SYM,VAL=BIT...
        $EQU    PREFIX''SYM,<VAL>
        .IIF    IDN <VAL> <BIT...>,     BIT...=BIT...+INCR
        .ENDM   $EQULS1
        .IIF    NB,INIT,        BIT...=INIT
        .IRP    L,<LIST>
        $EQULS1 L
        .ENDR
        .ENDM   $EQULST
```

```
;
; ASSUME - ASSEMBLY TIME CONSISTENCY CHECK
;
;       ASSUME EXP1 RELATION EXP2
;
;       FORCES ASSEMBLY ERROR IF EXP1 DOES NOT HAVE THE SPECIFIED
; NUMERICAL RELATION TO EXP2.
;

        .MACRO  ASSUME  EXP1,RELATION,EXP2
        .IF     RELATION <<EXP1>-<EXP2>>
        .IFF
        .ERROR  ; ***** EXP1 MUST BE RELATION EXP2 ;
        .ENDC
        .ENDM   ASSUME
```

UT
VC

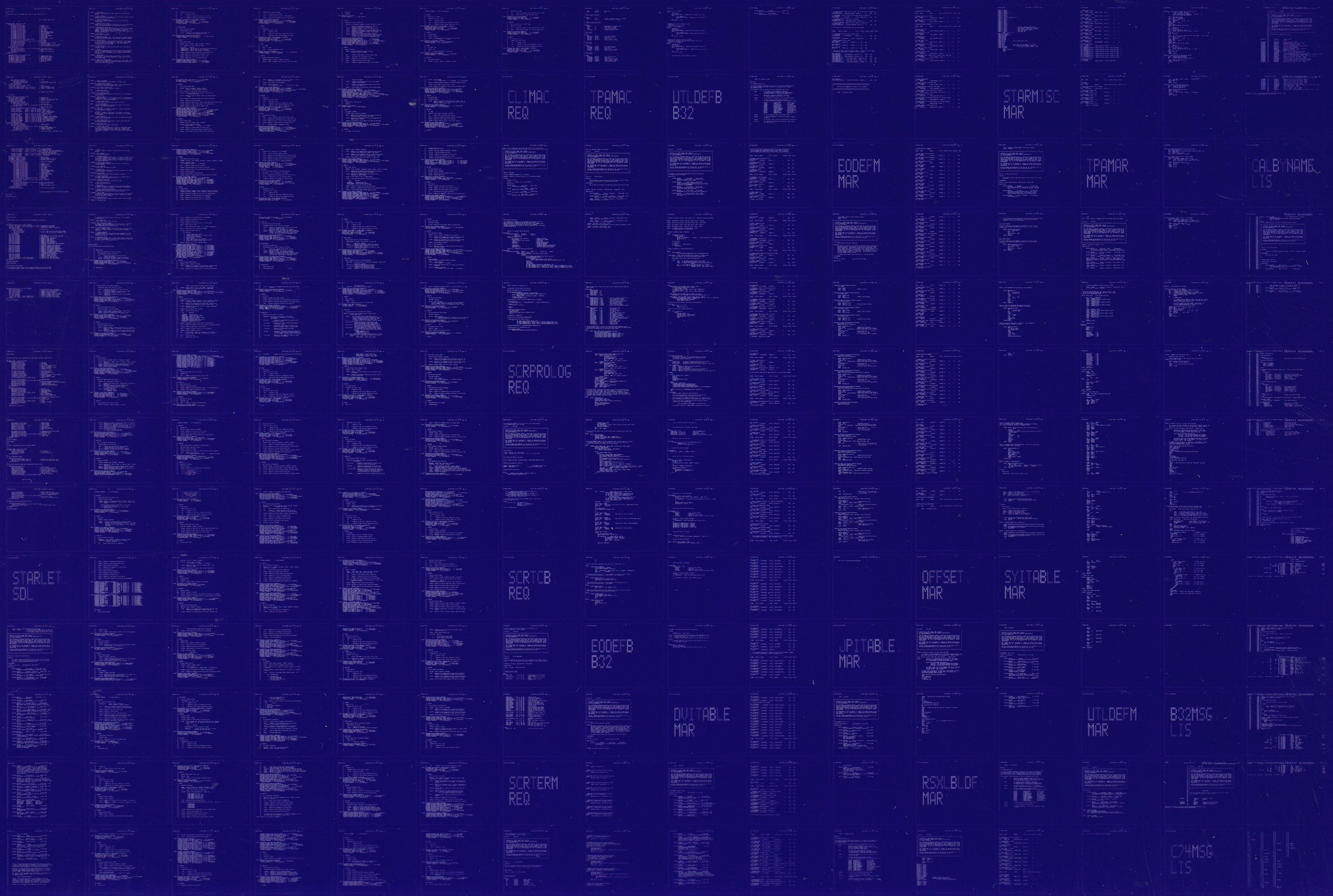
```
;++
; MACRO TO GENERATE A OFFSET LIST FOR A DATA STRUCTURE
;
;       IT IS USEFUL FOR INPUT ARGUMENT LISTS POSITIVELY INDEXED FROM AP, AND
;       WORK AREAS ALLOCATED IN CALL STACK AND NEGATIVELY INDEXED FROM FP.
;
; CALL: $OFFSET INITIAL,DIRECTION,<<LAB1,[SIZE]>,...,<LABN,[SIZE]>>
;
; WHERE:        INITIAL IS A REQUIRED VALUE FOR THE INTIAL INDEX WHEN
;               ORIGINATING A DATA STRUCTURE DEFINITION.  IT IS NORMALLY
;               (+) 4 FOR ARGUMENT LISTS AND 0 FOR WORK AREAS.
;
;               DIRECTION IS A KEYWORD THAT MUST BE:
;                       POSITIVE - FOR STRUCTURES GROWING UP IN MEMORY
;                       NEGATIVE - FOR STRUCTURES GROWING DOWN IN MEMORY
;                       OR BLANK, IN WHICH CASE "POSITIVE" IS ASSUMED.
;
;               THE LABEL, SIZE LIST IS THE SYMBOLIC NAME FOR THE LOCATION
;               AND THE OPTIONAL SIZE OF THE ELEMENT.  IF BLANK, SIZE IS
;               ASSUMED TO BE 4 ( ONE LONGWORD ).
;
;       TO PERMIT THE DEFINITION OF AN INDEFINITLY LARGE NUMBER OF LABELS,
;       THE MACRO MAY BE CONTINUED.  IN THIS CASE THE "INITIAL" AND
;       "DIRECTION" ARGUMENTS MUST BE BLANK.
;--
        .MACRO  $OFFSET INITVALUE,DIRECTION,SYMLST
        .SAVE   LOCAL_BLOCK
        .PSECT $ABS$ ABS
        .IF  B,INITVALUE
        .=SAVABS...
        .IF NB,DIRECTION
        .ERROR  ; DIRECTION MUST BE BLANK WHEN CONTINUING;
        .MEXIT
        .ENDC
        .IFF
        DIR...=1
        .=INITVALUE
        .IF NB,DIRECTION
        .IF IDN <DIRECTION>,<POSITIVE>
        .IFF
        .IF IDN <DIRECTION>,<NEGATIVE>
        DIR...=-1
        .IFF
        .ERROR  ; "DIRECTION" MUST BE "POSITIVE","NEGATIVE", OR BLANK;
        .ENDC
        .ENDC
        .ENDC
        .ENDC
        .IRP SYM,<SYMLST>
        $OFFST1 SYM
        .ENDR
        SAVABS...=.
        .RESTORE
        .ENDM $OFFSET

        .MACRO $OFFST1 SYM,SIZ=4
        .IF     LT,SIZ
        .ERROR                  ;***** SIZ PARAMETER NEGATIVE *****;
        .ENDC
        .IF     LT,DIR...
        .BLKB   -SIZ
        .ENDC
        .IF NB,SYM
        .LIST   MEB
SYM:
        .NLIST  MEB
        .ENDC
        .IF     GT,DIR...
        .BLKB   SIZ
        .ENDC
        .ENDM $OFFST1

; The $SHR_MSGDEF macro defines facility-specific message codes
;       which are based on the system-wide shared message codes.
;
;       $SHR_MSGDEF     name, code, scope, <<msg,severity>, ... >
;
;       where:
;               name    is the name of the facility (e.g., COPY)
;               code    is the corresponding facility code (e.g., 103)
;               scope   is GLOBAL to define globally, else defined locally
;               msg     is the name of the shared message (e.g., COPIEDB)
;               severity is the desired message severity (e.g., 1, 0, 2, 4)
;
;       Symbols of the form 'name'$_'msg' (e.g. COPY$_COPIEDB) are defined
;-
        .MACRO  $SHR_MSGDEF     NAME,CODE,SCOPE,MSGCODES
;
        .IF     NDF,SHR$K_SHRDEF        ; issue $SHRDEF if not done yet
                SHR$K_SHRDEF = 1        ; define symbol to indic $SHRDEF done
                $SHRDEF                 ; define shared message codes
        .ENDC
        $$GBL   = 0
        .IIF    IDN,SCOPE,GLOBAL,$$GBL = 1
        .IRP    MSGPAIR, <'MSGCODES>
                $SHR_MSGCOD 'NAME', 'CODE', MSGPAIR
        .ENDR
        .ENDM
        .MACRO  $SHR_MSGCOD NAME, CODE, MSG, SEVERITY

        .IF     IDN,SEVERITY,WARNING            ; if WARNING, set 0 sev
         .IF EQ $$GBL
          'NAME'$_'MSG' = 0                     ; set 0 sev (WARNING)
         .IFF
          'NAME'$_'MSG' == 0                    ; set 0 sev (WARNING)
         .ENDC
        .IFF
          .IF   IDN,SEVERITY,SUCCESS            ; if SUCCESS, set 1 sev
           .IF EQ       $$GBL
            'NAME'$_'MSG' = 1                   ; set 1 sev (SUCCESS)
           .IFF
            'NAME'$_'MSG' == 1                  ; set 1 sev (SUCCESS)
```

```
	    .ENDC
	   .IFF
	     .IF IDN,SEVERITY,ERROR		; if ERROR, set 2 sev
	      .IF EQ	$$GBL
	       'NAME'$_'MSG' = 2		; set 2 sev (ERROR)
	      .IFF
	       'NAME'$_'MSG' == 2		; set 2 sev (ERROR)
	      .ENDC
	     .IFF
	       .IF IDN,SEVERITY,INFO		; if INFO, set 3 sev
	        .IF EQ	$$GBL
	         'NAME'$_'MSG' = 3		; set 3 sev (INFO)
	        .IFF
	         'NAME'$_'MSG' == 3		; set 3 sev (INFO)
	        .ENDC
	       .IFF
	         .IF IDN,SEVERITY,SEVERE	; if SEVERE, set 4 sev
	          .IF EQ $$GBL
	           'NAME'$_'MSG' = 4		; set 4 sev (SEVERE)
	          .IFF
	           'NAME'$_'MSG' == 4		; set 4 sev (SEVERE)
	          .ENDC
	         .IFF
	          .IF EQ $$GBL
	           'NAME'$_'MSG' = 'SEVERITY	; set specified sev
	          .IFF
	           'NAME'$_'MSG' == 'SEVERITY	; set specified sev
	          .ENDC
	         .ENDC
	       .ENDC
	     .ENDC
	    .ENDC
	.ENDC
	.IF EQ $$GBL
	 'NAME'$_'MSG' = 'NAME'$_'MSG'+SHR$_'MSG'+<'CODE'@16>
	.IFF
	 'NAME'$_'MSG' == 'NAME'$_'MSG'+SHR$_'MSG'+<'CODE'@16>
	.ENDC
	.ENDM
```

```
        .LIST
```

0434 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


CLIMAC
REQ

TPAMAC
REQ

UTLDEFB
B32

STARMISC
MAR

EODEFM
MAR

TPAMAR
MAR

CALBYNAME
LIS

SCRPROLOG
REQ

STARLET
SDL

SCRTCB
REQ

OFFSET
MAR

SYITABLE
MAR

EODEFB
B32

JPITABLE
MAR

DVITABLE
MAR

UTLDEFM
MAR

B32MSG
LIS

SCRTERM
REQ

RSXLBLDF
MAR

C74MSG
LIS