MP

```
MM      MM   PPPPPPPP     SSSSSSSS    CCCCCCCC  BBBBBBBB   VV      VV   EEEEEEEEEE   CCCCCCCC
MM      MM   PPPPPPPP     SSSSSSSS    CCCCCCCC  BBBBBBBB   VV      VV   EEEEEEEEEE   CCCCCCCC
MMMM  MMMM   PP      PP   SS          CC        BB      BB VV      VV   EE           CC
MMMM  MMMM   PP      PP   SS          CC        BB      BB VV      VV   EE           CC
MM  MM  MM   PP      PP   SS          CC        BB      BB VV      VV   EE           CC
MM  MM  MM   PP      PP   SS          CC        BB      BB VV      VV   EE           CC
MM      MM   PPPPPPPP      SSSSSS     CC        BBBBBBBB   VV      VV   EEEEEEEE     CC
MM      MM   PPPPPPPP      SSSSSS     CC        BBBBBBBB   VV      VV   EEEEEEEE     CC
MM      MM   PP                SS     CC        BB      BB VV      VV   EE           CC
MM      MM   PP                SS     CC        BB      BB  VV    VV    EE           CC
MM      MM   PP                SS     CC        BB      BB  VV    VV    EE           CC
MM      MM   PP                SS     CC        BB      BB  VV    VV    EE           CC      ....
MM      MM   PP          SSSSSSSS     CCCCCCCC  BBBBBBBB     VV         EEEEEEEEEE   CCCCCCCC  ....
MM      MM   PP          SSSSSSSS     CCCCCCCC  BBBBBBBB     VV         EEEEEEEEEE   CCCCCCCC  ....


LL           IIIIII    SSSSSSSS
LL           IIIIII    SSSSSSSS
LL             II      SS
LL             II      SS
LL             II      SS
LL             II       SSSSSS
LL             II       SSSSSS
LL             II            SS
LL             II            SS
LL             II            SS
LL             II            SS
LLLLLLLLLL   IIIIII    SSSSSSSS
LLLLLLLLLL   IIIIII    SSSSSSSS
```

MPSCBVEC
V04-000

D 15

- MULTI-PROCESSOR SCB VECTOR   16-SEP-1984 02:01:51   VAX/VMS_Macro V04-00   Page 1
                                       5-SEP-1984 15:12:49   [MP.SRC]MPPREFIX.MAR;1   (1)

MF
VC

```
0000      1 ;
0000      2 ; Version:        'V04-000'
0000      3 ;
0000      4
0000      5          .MCALL  MFPR
0000      1          .TITLE  MPSCBVEC - MULTI-PROCESSOR SCB VECTOR
0000      2          .IDENT  'V04-000'
0000      3
0000      4 ;
0000      5 ;********************************************************************
0000      6 ;*                                                                  *
0000      7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                         *
0000      8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.          *
0000      9 ;*  ALL RIGHTS RESERVED.                                            *
0000     10 ;*                                                                  *
0000     11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000     12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
0000     13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
0000     14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000     15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
0000     16 ;*  TRANSFERRED.                                                    *
0000     17 ;*                                                                  *
0000     18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
0000     19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
0000     20 ;*  CORPORATION.                                                    *
0000     21 ;*                                                                  *
0000     22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
0000     23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.          *
0000     24 ;*                                                                  *
0000     25 ;*                                                                  *
0000     26 ;********************************************************************
0000     27 ;
0000     28
0000     29 ;++
0000     30
0000     31 ; Facility:  Executive , Hardware fault handling
0000     32
0000     33 ; Abstract:  SCB vectors
0000     34
0000     35 ; Environment: MODE=Kernel
0000     36
0000     37 ; Author: RICHARD I. HUSTVEDT, Creation date:  15-MAY-1979
0000     38
0000     39 ; Modified by:
0000     40
0000     41 ;       V03-005 KDM0030         Kathleen D. Morse        18-Nov-1982
0000     42 ;               Remove pagefault handler, as now primary can execute
0000     43 ;               secondary-specific code without turning into a secondary.
0000     44
0000     45 ;       V03-004 KDM0025         Kathleen D. Morse        10-Oct-1982
0000     46 ;               Change secondary specific routines to be MPS$xxx instead
0000     47 ;               of EXE$xxx (MPS$SWTIMINT, MPS$HWCLKINT).
0000     48
0000     49 ;       V03-003 KDM0018         Kathleen D. Morse        10-Oct-1982
0000     50 ;               Add CHMK routine for secondary processor.
0000     51
0000     52 ;       V03-002 KDM0005         Kathleen D. Morse        10-Oct-1982
```

MPSCBVEC
V04-000

E 15

- MULTI-PROCESSOR SCB VECTOR

16-SEP-1984 02:01:51   VAX/VMS Macro V04-00
5-SEP-1984 02:07:17   [MP.SRC]MPSCBVEC.MAR;1

Page   2
(1)

```
0000    53  ;
0000    54  ;
0000    55  ;
0000    56  ;
0000    57  ; 01     -
0000    58  ;--
0000    59
```

Add pagefault handler for secondary processor.

```
                    0000    61                .SBTTL   SCB
                    0000    62        ;
                    0000    63        ; INCLUDE FILES:
                    0000    64        ;
                    0000    65
                    0000    66        ;
                    0000    67        ; MACROS:
                    0000    68        ;
                    0000    69
                    0000    70        ;
                    0000    71        ; Macro DEFVEC defines an Unused vector.
                    0000    72        ;
                    0000    73
                    0000    74                .MACRO   DEFVEC,VNUM
                    0000    75                .LONG    ERL$VEC'VNUM+INTSTK
                    0000    76                .ENDM    DEFVEC
                    0000    77
                    0000    78        ;
                    0000    79        ; EQUATED SYMBOLS:
                    0000    80        ;
                    0000    81
                    0000    82                $IPLDEF                          ; Define IPL levels
                    0000    83                $PRDEF                           ; Define processor registers
                    0000    84
        00000001    0000    85 INTSTK = 1                                      ; Definition for interrupt stack
        00000100    0000    86 VNUM = 256                                     ; Vectors ^X100-^X1FC for nexus devices
                    0000    87
                    0000    88        ;
                    0000    89        ; OWN STORAGE:
                    0000    90        ;
                    0000    91
                    0000    92
```

```
                    0000    94  :+
                    0000    95  ;
                    0000    96  ; FUNCTIONAL DESCRIPTION:
                    0000    97  ;
                    0000    98  ; This module contains the SCB vectors which are copied into physically
                    0000    99  ; contiguous space, replacing those used by a single processor VMS system.
                    0000   100  ;
                    0000   101  ;-
          00000000  102          .PSECT   $$$00SCB,PAGE   ;
                    0000   103  SCB$AL_BASE::                    ; SCB base address
   00000000' 0000   104          .LONG   ERL$VEC0         ; Unused, reserved to Digital
   00000001' 0004   105          .LONG   MPS$MCHK+INTSTK  ; Machine check handler.          ^X004
   00000001' 0008   106          .LONG   MPS$KERSTKNV+INTSTK ; Kernel stack not valid halt. ^X008
   00000001' 000C   107          .LONG   MPS$POWERFAIL+INTSTK ; Power fail interrupt.       ^X00C
   00000000' 0010   108          .LONG   ERL$VEC16        ; Reserved/privileged instruction fault. ^X010
   00000000' 0014   109          .LONG   ERL$VEC20        ; Customer reserved instruction fault. ^X014
   00000000' 0018   110          .LONG   ERL$VEC24        ; Reserved operand fault/halt.    ^X018
   00000000' 001C   111          .LONG   ERL$VEC28        ; Reserved addressing mode fault. ^X01C
   00000000' 0020   112          .LONG   ERL$VEC32        ; Access control violation fault. ^X020
   00000000' 0024   113          .LONG   ERL$VEC36        ; Translation not valid fault.    ^X024
   00000000' 0028   114          .LONG   XDEL'BIT         ; Tbit fault.                     ^X028
   00000000' 002C   115          .LONG   XDELBPT          ; Breakpoint fault.               ^X02C
   00000000' 0030   116          .LONG   MPS$COMPAT       ; Compatibility fault.            ^X030
   00000000' 0034   117          .LONG   ERL$VEC52        ; Arithmetic trap.                ^X034
   00000000' 0038   118          .LONG   ERL$VEC56        ; Unused.                         ^X038
   00000000' 003C   119          .LONG   ERL$VEC60        ; Unused.                         ^X03C
   00000000' 0040   120          .LONG   MPS$CMODKRNL     ; CHMK trap.                      ^X040
   00000000' 0044   121          .LONG   EXE$CMODEXEC     ; CHME trap.                      ^X044
   00000000' 0048   122          .LONG   EXE$CMODSUPR     ; CHMS trap.                      ^X048
   00000000' 004C   123          .LONG   EXE$CMODUSER     ; CHMU trap.                      ^X04C
   00000000' 0050   124          .LONG   ERL$VEC80        ; Unused.                         ^X050
   00000001' 0054   125          .LONG   MPS$INT54+INTSTK ; Cpu-dependent fault.            ^X054
   00000001' 0058   126          .LONG   MPS$INT58+INTSTK ; Cpu-dependent fault.            ^X058
   00000001' 005C   127          .LONG   MPS$INT5C+INTSTK ; Cpu-dependent fault.            ^X05C
   00000001' 0060   128          .LONG   MPS$INT60+INTSTK ; Cpu-dependent fault.            ^X060
   00000000' 0064   129          .LONG   ERL$VEC100       ; Unused.                         ^X064
   00000000' 0068   130          .LONG   ERL$VEC104       ; Unused.                         ^X068
   00000000' 006C   131          .LONG   ERL$VEC108       ; Unused.                         ^X06C
   00000000' 0070   132          .LONG   ERL$VEC112       ; Unused.                         ^X070
   00000000' 0074   133          .LONG   ERL$VEC116       ; Unused.                         ^X074
   00000000' 0078   134          .LONG   ERL$VEC120       ; Unused.                         ^X078
   00000000' 007C   135          .LONG   ERL$VEC124       ; Unused.                         ^X07C
   00000000' 0080   136          .LONG   ERL$VEC128       ; Unused.                         ^X080
   00000000' 0084   137          .LONG   ERL$VEC132       ; Software level 1 interrupt.     ^X084
   00000000' 0088   138          .LONG   MPS$ASTDEL       ; Software level 2 interrupt.     ^X088
   00000000' 008C   139          .LONG   MPS$MPSCHED      ; Software level 3 interrupt.     ^X08C
   00000001' 0090   140          .LONG   ERL$VEC144+INTSTK ; Software level 4 interrupt.    ^X090
   00000001' 0094   141          .LONG   ERL$VEC148+INTSTK ; Software level 5 interrupt.    ^X094
   00000001' 0098   142          .LONG   ERL$VEC152+INTSTK ; Software level 6 interrupt.    ^X098
   00000000' 009C   143          .LONG   MPS$SWTIMINT     ; Software level 7 interrupt.     ^X09C
   00000001' 00A0   144          .LONG   ERL$VEC160+INTSTK ; Software level 8 interrupt.    ^X0A0
   00000001' 00A4   145          .LONG   ERL$VEC164+INTSTK ; Software level 9 interrupt.    ^X0A4
   00000001' 00A8   146          .LONG   ERL$VEC168+INTSTK ; Software level 10 interrupt.   ^X0A8
   00000001' 00AC   147          .LONG   ERL$VEC172+INTSTK ; Software level 11 interrupt.   ^X0AC
   00000C01' 00B0   148          .LONG   ERL$VEC176+INTSTK ; Software level 12 interrupt.   ^X0B0
   00000001' 00B4   149          .LONG   ERL$VEC180+INTSTK ; Software level 13 interrupt.   ^X0B4
   00000001' 00B8   150          .LONG   ERL$VEC184+INTSTK ; Software level 14 interrupt.   ^X0B8
```

```
00000000'  00BC   151        .LONG   MPS$XDELTAINT        ; Software level 15 interrupt. ^X0BC
00000001'  00C0   152        .LONG   MPS$HWCLKINT+INTSTK  ; Interval timer.              ^X0C0
00000001'  00C4   153        .LONG   ERL$VEC196+INTSTK    ; Unused.                      ^X0C4
00000001'  00C8   154        .LONG   ERL$VEC200+INTSTK    ; Unused.                      ^X0C8
00000001'  00CC   155        .LONG   ERL$VEC204+INTSTK    ; Unused.                      ^X0CC
00000001'  00D0   156        .LONG   ERL$VEC208+INTSTK    ; Unused.                      ^X0D0
00000001'  00D4   157        .LONG   ERL$VEC212+INTSTK    ; Unused.                      ^X0D4
00000001'  00D8   158        .LONG   ERL$VEC216+INTSTK    ; Unused.                      ^X0D8
00000001'  00DC   159        .LONG   ERL$VEC220+INTSTK    ; Unused.                      ^X0DC
00000001'  00E0   160        .LONG   ERL$VEC224+INTSTK    ; Unused.                      ^X0E0
00000001'  00E4   161        .LONG   ERL$VEC228+INTSTK    ; Unused.                      ^X0E4
00000001'  00E8   162        .LONG   ERL$VEC232+INTSTK    ; Unused.                      ^X0E8
00000001'  00EC   163        .LONG   ERL$VEC236+INTSTK    ; Unused.                      ^X0EC
00000001'  00F0   164        .LONG   ERL$VEC240+INTSTK    ; Unused.                      ^X0F0
00000001'  00F4   165        .LONG   ERL$VEC244+INTSTK    ; Unused.                      ^X0F4
00000001'  00F8   166        .LONG   ERL$VEC248+INTSTK    ; Console interrupt for input  ^X0F8
00000001'  00FC   167        .LONG   ERL$VEC252+INTSTK    ; Console interrupt for output ^X0FC
           0100   168
           0100   169
           0100   170        .REPT   16                   ; Nexus vectors, IPL ^X14
           0100   171        DEFVEC  \VNUM
           0100   172        VNUM=VNUM+4
           0100   173        .ENDR
           0140   174
           0140   175        .REPT   16                   ; Nexus vectors, IPL ^X15
           0140   176        DEFVEC  \VNUM
           0140   177        VNUM=VNUM+4
           0140   178        .ENDR
           0180   179
           0180   180        .REPT   16                   ; Nexus Vectors, IPL ^X16
           0180   181        DEFVEC  \VNUM
           0180   182        VNUM=VNUM+4
           0180   183        .ENDR
           01C0   184
           01C0   185        .REPT   16                   ; Nexus vectors, IPL ^X17
           01C0   186        DEFVEC  \VNUM
           01C0   187        VNUM=VNUM+4
           01C0   188        .ENDR
           0200   189
           0200   190        .END
```

I 15

MPSCBVEC   - MULTI-PROCESSOR SCB VECTOR   16-SEP-1984 02:01:51   VAX/VMS Macro V04-00   Page   6
Symbol table   5-SEP-1984 02:07:17   [MP.SRC]MPSCBVEC.MAR;1   (1)

ERL$VEC0	********	X	02	ERL$VEC324	********	X	02	MPS$CMODKRNL	*****:..	X	02
ERL$VEC100	********	X	02	ERL$VEC328	********	X	02	MPS$COMPAT	********	X	02
ERL$VEC104	********	X	02	ERL$VEC332	********	X	02	MPS$HWCLKINT	********	X	02
ERL$VEC108	********	X	02	ERL$VEC336	********	X	02	MPS$INT54	********	X	02
ERL$VEC112	********	X	02	ERL$VEC340	********	X	02	MPS$INT58	********	X	02
ERL$VEC116	********	X	02	ERL$VEC344	********	X	02	MPS$INT5C	********	X	02
ERL$VEC120	********	X	02	ERL$VEC348	********	X	02	MPS$INT60	********	X	02
ERL$VEC124	********	X	02	ERL$VEC352	********	X	02	MPS$KERSTKNV	********	X	02
ERL$VEC128	********	X	02	ERL$VEC356	********	X	02	MPS$MCHK	********	X	02
ERL$VEC132	********	X	02	ERL$VEC36	********	X	02	MPS$MPSCHED	********	X	02
ERL$VEC144	********	X	02	ERL$VEC360	********	X	02	MPS$POWERFAIL	********	X	02
ERL$VEC148	********	X	02	ERL$VEC364	********	X	02	MPS$SWTIMINT	********	X	02
ERL$VEC152	********	X	02	ERL$VEC368	********	X	02	MPS$XDELTAINT	********	X	
ERL$VEC16	********	X	02	ERL$VEC372	********	X	02	SCB$AL_BASE	00000000	RG	02
ERL$VEC160	********	X	02	ERL$VEC376	********	X	02	VNUM	= 00000200		
ERL$VEC164	********	X	02	ERL$VEC380	********	X	02	XDELBPT	********	X	02
ERL$VEC168	********	X	02	ERL$VEC384	********	X	02	XDELTBIT	********	X	02
ERL$VEC172	********	X	02	ERL$VEC388	********	X	02				
ERL$VEC176	********	X	02	ERL$VEC392	********	X	02				
ERL$VEC180	********	X	02	ERL$VEC396	********	X	02				
ERL$VEC184	********	X	02	ERL$VEC400	********	X	02				
ERL$VEC196	********	X	02	ERL$VEC404	********	X	02				
ERL$VEC20	********	X	02	ERL$VEC408	********	X	02				
ERL$VEC200	********	X	02	ERL$VEC412	********	X	02				
ERL$VEC204	********	X	02	ERL$VEC416	********	X	02				
ERL$VEC208	********	X	02	ERL$VEC420	********	X	02				
ERL$VEC212	********	X	02	ERL$VEC424	********	X	02				
ERL$VEC216	********	X	02	ERL$VEC428	********	X	02				
ERL$VEC220	********	X	02	ERL$VEC432	********	X	02				
ERL$VEC224	********	X	02	ERL$VEC436	********	X	02				
ERL$VEC228	********	X	02	ERL$VEC440	********	X	02				
ERL$VEC232	********	X	02	ERL$VEC444	********	X	02				
ERL$VEC236	********	X	02	ERL$VEC448	********	X	02				
ERL$VEC24	********	X	02	ERL$VEC452	********	X	02				
ERL$VEC240	********	X	02	ERL$VEC456	********	X	02				
ERL$VEC244	********	X	02	ERL$VEC460	********	X	02				
ERL$VEC248	********	X	02	ERL$VEC464	********	X	02				
ERL$VEC252	********	X	02	ERL$VEC468	********	X	02				
ERL$VEC256	********	X	02	ERL$VEC472	********	X	02				
ERL$VEC260	********	X	02	ERL$VEC476	********	X	02				
ERL$VEC264	********	X	02	ERL$VEC480	********	X	02				
ERL$VEC268	********	X	02	ERL$VEC484	********	X	02				
ERL$VEC272	********	X	02	ERL$VEC488	********	X	02				
ERL$VEC276	********	X	02	ERL$VEC492	********	X	02				
ERL$VEC28	********	X	02	ERL$VEC496	********	X	02				
ERL$VEC280	********	X	02	ERL$VEC500	********	X	02				
ERL$VEC284	********	X	02	ERL$VEC504	********	X	02				
ERL$VEC288	********	X	02	ERL$VEC508	********	X	02				
ERL$VEC292	********	X	02	ERL$VEC52	********	X	02				
ERL$VEC296	********	X	02	ERL$VEC56	********	X	02				
ERL$VEC3G0	********	X	02	ERL$VEC60	********	X	02				
ERL$VEC304	********	X	02	ERL$VEC80	********	X	02				
ERL$VEC308	********	X	02	EXE$CMODEXEC	********	X	02				
ERL$VEC312	********	X	02	EXE$CMODSUPR	********	X	02				
ERL$VEC316	********	X	02	EXE$CMODUSER	********	X	02				
ERL$VEC32	********	X	02	INTSTK	= 00000001						
ERL$VEC320	********	X	02	MPS$ASTDEL	********	X	02				

```
                              +------------------+
                              ! Psect synopsis !
                              +------------------+


PSECT name                 Allocation          PSECT No.  Attributes
----------                 ----------          ---------  ----------
.   ABS   .                00000000 (     0.)  00 (   0.) NOPIC  USR  CON  ABS  LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                      00000000 (     0.)  01 (   1.) NOPIC  USR  CON  ABS  LCL NOSHR  EXE   RD    WRT NOVEC BYTE
$$$00SCB                   00000200 (   512.)  02 (   2.) NOPIC  USR  CON  REL  LCL NOSHR  EXE   RD    WRT NOVEC PAGE


                          +----------------------------+
                          ! Performance indicators !
                          +----------------------------+


Phase                  Page faults   CPU Time      Elapsed Time
-----                  -----------   --------      ------------
Initialization                  32   00:00:00.07   00:00:00.70
Command processing             119   00:00:00.92   00:00:04.73
Pass 1                         171   00:00:02.62   00:00:10.74
Symbol table sort                0   00:00:00.15   00:00:00.24
Pass 2                          59   00:00:00.80   00:00:01.84
Symbol table output             11   00:00:00.10   00:00:00.18
Psect synopsis output            2   00:00:00.03   00:00:00.03
Cross-reference output           0   00:00:00.00   00:00:00.00
Assembler run totals           396   00:00:04.70   00:00:18.52

The working set limit was 1200 pages.
11725 bytes (23 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 223 non-local and 0 local symbols.
195 source lines were read in Pass 1, producing 21 object records in Pass 2.
11 pages of virtual memory were used to define 10 macros.


                          +----------------------------+
                          ! Macro library statistics !
                          +----------------------------+


Macro library name                            Macros defined
------------------                            --------------
_$255$DUA28:[MP.OBJ]MP.MLB;1                          1
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                        1
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     4
TOTALS (all libraries)                               6

177 GETS were required to define 6 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:MPSCBVEC/OBJ=OBJ$:MPSCBVEC MSRC$:MPPREFIX/UPDATE=(ENH$:MPPREFIX)+MSRC$:MPSCBVEC/UPDATE=(ENH$:MPSCBVEC)+EXECML$/LIB+LI
```

MPERRLOG
LIS

MPSCBVEC
LIS

MPINT
LIS

MPPFM
LIS

MPDAT
LIS

MPPWRFAIL
LIS

MPMCHECK
LIS

MPINTEXC
LIS

MPLOG
LIS

MPERRMSG
LIS

MPSCHED
LIS

MPSHWPFM
LIS

MPLOAD
LIS

MPINIT
LIS