


```

MM      MM      PPPPPPP      MM      MM      AAAAAA      CCCCCCCC      RRRRRRRR      000000      SSSSSSSS
MM      MM      PPPPPPP      MM      MM      AAAAAA      CCCCCCCC      RRRRRRRR      000000      SSSSSSSS
MMMM    MMMM    PP          PP    MMMM    MMMM    AA          AA      CC          RR          RR      00          00      SS
MMMM    MMMM    PP          PP    MMMM    MMMM    AA          AA      CC          RR          RR      00          00      SS
MM      MM      MM      PP          PP    MM      MM      AA          AA      CC          RR          RR      00          00      SS
MM      MM      MM      PP          PP    MM      MM      AA          AA      CC          RR          RR      00          00      SS
MM      MM      PPPPPPP      MM      MM      AA          AA      CC          RRRRRRRR      00          00      SSSSSS
MM      MM      PPPPPPP      MM      MM      AA          AA      CC          RRRRRRRR      00          00      SSSSSS
MM      MM      PP          MM      MM      AAAAAAAAAA      CC          RR      RR      00          00      SS
MM      MM      PP          MM      MM      AAAAAAAAAA      CC          RR      RR      00          00      SS
MM      MM      PP          MM      MM      AA          AA      CC          RR      RR      00          00      SS
MM      MM      PP          MM      MM      AA          AA      CC          RR      RR      00          00      SS
MM      MM      PP          MM      MM      AA          AA      CCCCCCCC      RR          RR      000000      SSSSSSSS
MM      MM      PP          MM      MM      AA          AA      CCCCCCCC      RR          RR      000000      SSSSSSSS

```

```

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

```

```

....
....
....
....

```


.TITLE MPMACROS - Macros for Multi-Processing Code
.IDENT '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: Multi-Processing Code
: Abstract: Macros used by multi-processing code
: Environment: MODE=Kernel
: Author: Kathleen D. Morse, Creation date: 21-Jul-1981
: Modified by:
:   V03-002 KDM0053      Kathleen D. Morse      11-Jul-1983
:           Change references to cpu-specific processor register
:           to use new cpu-specific symbol, PR780$_TODR.
:   V03-001 KDM0030      Kathleen D. Morse      18-Nov-1982
:           Added IFPRIMARY macro.

```

--
.PAGE

: Include files:

: MACROS:

MF
S)
AC
AC
AC
AC
AS
E)
E)
IF
L)
MF
MF
MF
MF
MF
MF
MF
PC
PC
PC
PC
PH
PR
PR
PS
QE
QR
SC
SC
SC
SC
SS
SS

PS
--
\$)
A)

```

:
:
: This macro is used for secondary processor bugchecks. It
: has a different name so that primary processor bugchecks
: work as they always have.
:

```

```

.MACRO SECBUG_CHECK ERROR,TYPE=CONT
BSBW W^MPSS$SECBUGCHK
.IIF IDN <TYPE>,<FATAL> , .WORD BUG$_'ERROR'!4
.IIF DIF <TYPE>,<FATAL> , .WORD BUG$_'ERROR'
.ENDM SECBUG_CHECK

```

```

:
: This macro updates the time-of-day processor register before
: it is accessed with an MFPR instruction. This guarantees that
: the correct value is used, since $SETTIME requests are not
: reflected on the secondary processor.
:

```

```

.MACRO MFPR SRC,DST,?L1
.IF IDN <SRC>,<#PR780$ TODR>
PUSHR #^M<R0,R1,R2,R3>
L1: MOVQ G^EXE$GQ_SYSTIME,R0
MOVQ G^EXE$GQ_SYSTIME,R2
CML R0,R2
BNEQ L1
CML R1,R3
BNEQ L1
MOVQ G^EXE$GQ_TODCBASE,R2
SUBL R2,R0
SBWC R3,R1
EDIV #<100*1000*2>,R0,R0,R1
ASHL #1,R1,R1
ADDL G^EXE$GL TODR,R1
MTPR R1,#PR780$ TODR
POPR #^M<R0,R1,R2,R3>
.ENDC
.MDELETE MFPR
MFPR 'SRC','DST'
.MCALL MFPR
.ENDM MFPR

```

```

:
: This macro checks if the code is being executed on the primary
: processor or the secondary processor. If it is on the primary,
: then the input argument is the instruction to execute. Otherwise,
: it branches around the input argument.
:

```

```

.MACRO IFPRIMARY INSTR,?L1
MFPR #PR$ SCBB,-(SP)
MOVZBL #RPB$L SCBB,-(SP)
ADDL G^EXE$GL RPB,(SP)
CML @(SP)+,(SP)+

```


L1: BNEQ L1
INSTR

.ENDM IFPRIMARY

:
: Equated Symbols:
:

.END

