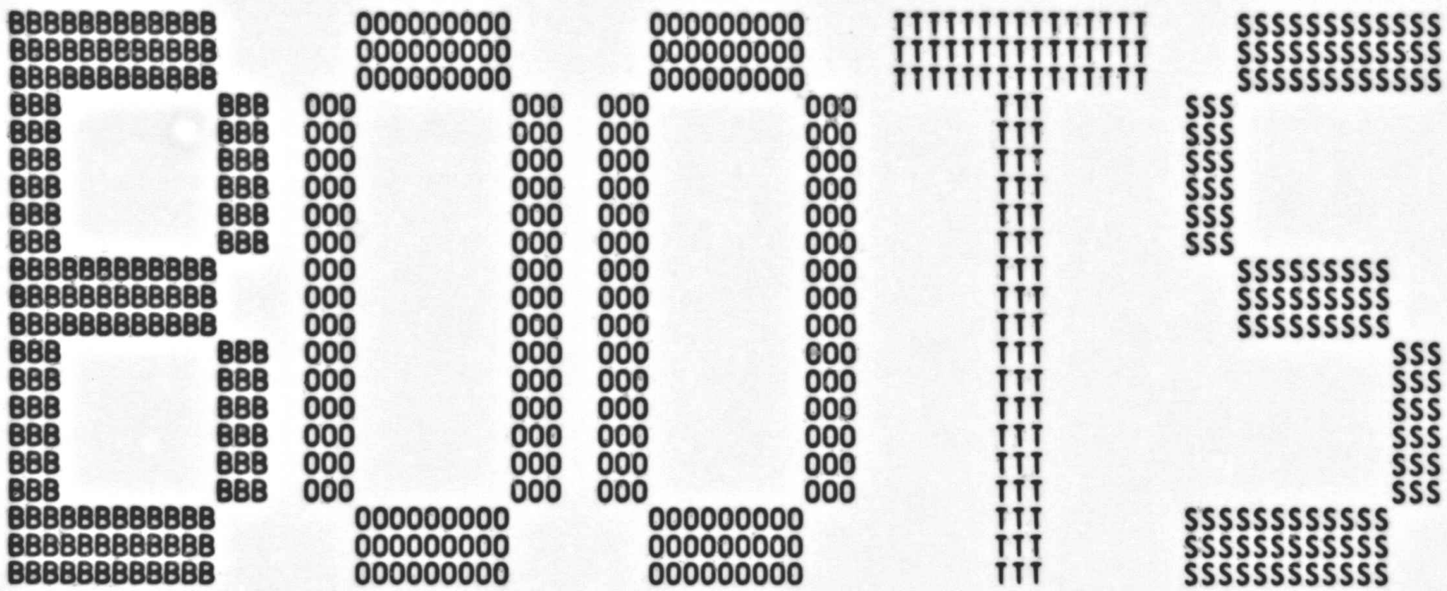
BBBBBBBBBBB OOOOOOOOO OOOOOOOOO TTTTTTTTTTTTTTT SSSSSSSSSSS
BBB BBB OOO OOO OOO OOO TTT SSS
BBBBBBBBBBB OOO OOO OOO OOO TTT SSSSSSSSS
BBB BBB OOO OOO OOO OOO TTT SSS
BBBBBBBBBBB OOOOOOOOO OOOOOOOOO TTT SSSSSSSSSSS

FILEID**BOOTS

```
BBBBBBBB      OOOOOO      OOOOOO    TTTTTTTTTT    SSSSSSSS
BBBBBBBB      OOOOOO      OOOOOO    TTTTTTTTTT    SSSSSSSS
BB      BB  OO      OO  OO      OO      TT      SS
BB      BB  OO      OO  OO      OO      TT      SS
BB      BB  OO      OO  OO      OO      TT      SS
BB      BB  OO      OO  OO      OO      TT      SS
BBBBBBBB    OO      OO  OO      OO      TT        SSSSSS
BBBBBBBB    OO      OO  OO      OO      TT        SSSSSS
BB      BB  OO      OO  OO      OO      TT              SS
BB      BB  OO      OO  OO      OO      TT              SS
BB      BB  OO      OO  OO      OO      TT              SS      ....
BB      BB  OO      OO  OO      OO      TT              SS      ....
BBBBBBBB      OOOOOO      OOOOOO        TT      SSSSSSSS      ....
BBBBBBBB      OOOOOO      OOOOOO        TT      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  BOOTS MACROS
        .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:     BOOTS
;
; ABSTRACT:
;       This module contains macros for the BOOTS facility
;
; ENVIRONMENT:
;
; AUTHOR:  STEVE BECKHARDT,     CREATION DATE:  31-Oct-1979
;
; MODIFIED BY:
;
;       V03-005 KDM0073         Kathleen D. Morse       22-Aug-1983
;               Change TIMEDWAIT macro to include load address for
;               VMB (e.g., address of RPB).
;
;       V03-004 KDM0058         Kathleen D. Morse       13-Jul-1983
;               Add boot-time specific TIMEDWAIT macro for boot drivers.
;
;       V03-003 KTA3058         Kerbey T. Altmann       20-Jun-1983
;               Add cell for boot device name (may be different from
;               driver name!).  Also unit disconnect routine.
;
;       V03-002 KTA3034         Kerbey T. Altmann       02-Feb-1983
;               Add cell for booting node name.
;
;--
```

```
        .SBTTL  DECLARATIONS
;
; INCLUDE FILES:
;
;
; MACROS:
;
; $BOOT_DRIVER MACRO - SETS UP A TABLE ENTRY FOR A BOOT DEVICE DRIVER.
;       EACH TABLE ENTRY CONTAINS:
;
;       CPUTYPE         CPU TYPE.  DEFAULT = -1 (DON'T CARE)
;
;       DEVTYPE         BOOT DEVICE TYPE VALUE.  DEFAULT = -1 (DON'T CARE)
;
;       ACTION          ACTION ROUTINE ADDRESS (ACTUALLY OFFSET FROM
;                       START OF TABLE ENTRY).  DEFAULT = 0 (NONE).
;
;       SIZE            SIZE OF ENTIRE DRIVER IN BYTES.  CANNOT BE DEFAULTED.
;
;       ADDR            ADDRESS OF DRIVER (ACTUALLY OFFSET FROM START
;                       OF TABLE ENTRY).  CANNOT BE DEFAULTED.
;
;       ENTRY           ADDRESS OF DRIVER ENTRY POINT (ACTUALLY OFFSET
;                       FROM ADDRESS OF DRIVER).  DEFAULT = 0 (ADDRESS
;                       OF DRIVER AND ENTRY POINT ARE THE SAME).
;
;       DRIVRNAME       ADDRESS OF DRIVER NAME IN .ASCIC.  (ACTUALLY
;                       OFFSET FROM ADDRESS OF DRIVER).  CANNOT BE
;                       DEFAULTED.
;
;       AUXDRNAME       AUXDRNAME ROUTINE ADDRESS (ACTUALLY OFFSET FROM
;                       START OF TABLE ENTRY).  DEFAULT = 0 (NONE).
;
;       UNIT_INIT       UNIT_INIT ROUTINE ADDRESS (ACTUALLY OFFSET FROM
;                       START OF TABLE ENTRY).  DEFAULT = 0 (NONE).
;
;       UNIT_DISC       UNIT_DISCONNECT ROUTINE ADDRESS (ACTUALLY OFFSET
;                       FROM START OF TABLE ENTRY).  DEFAULT = 0 (NONE).
;
;       DEVNAME         BOOT DEVICE NAME ADDRESS (ACTUALLY OFFSET FROM
;                       START OF TABLE ENTRY).  DEFAULT = FIRST TWO
;                       LETTERS OF DRIVRNAME.
;
        .MACRO  $BOOT_DRIVER    CPUTYPE=-1,DEVTYPE=-1,ACTION,SIZE,ADDR,-
                                ENTRY,DRIVRNAME,AUXDRNAME,UNIT_INIT,-
                                UNIT_DISC,DEVNAME
        .PSECT  BOOTDRIVR_4
$TABLE=.
        .WORD   CPUTYPE
        .IF     EQ      CPUTYPE
        .ERROR  0       ; CPU TYPE CANNOT BE 0 ;
        .ENDC
        .WORD   DEVTYPE
```

```
        .IF     B       ACTION
        .LONG   0
        .IFF
        .LONG   ACTION-$TABLE
        .ENDC
        .LONG   SIZE
        .LONG   ADDR-$TABLE
        .IF     B       ENTRY
        .LONG   0
        .IFF
        .LONG   ENTRY-ADDR
        .ENDC
        .LONG   DRIVRNAME-ADDR
        .IF     B       AUXDRNAME
        .LONG   0
        .IFF
        .LONG   AUXDRNAME-ADDR
        .ENDC
        .IF     B       UNIT_INIT
        .LONG   0
        .IFF
        .LONG   UNIT_INIT-ADDR
        .ENDC
        .IF     B       UNIT_DISC
        .LONG   0
        .IFF
        .LONG   UNIT_DISC-ADDR
        .ENDC
        .IF     B       DEVNAME
        .LONG   DRIVRNAME-ADDR+1
        .IFF
        .LONG   DEVNAME-ADDR
        .ENDC
        .PSECT  BOOTDRIVR_2
        .ENDM   $BOOT_DRIVER


;
; Define the offsets into the argument list passed by VMB to SYSBOOT
;
        .MACRO  $VMBARGDEF,GBL

        $DEFINI VMB,GBL,4

$DEF    VMB$Q_FILECACHE         .BLKQ   1       ; FILEREAD Cache Descriptor
$DEF    VMB$L_LO_PFN            .BLKL   1       ; Lowest PFN found by VMB
$DEF    VMB$L_HI_PFN            .BLKL   1       ; Highest PFN exclusive
$DEF    VMB$Q_PFNMAP            .BLKQ   1       ; PFN Bitmap descriptor
$DEF    VMB$Q_UCODE             .BLKQ   1       ; Loaded ucode descriptor
$DEF    VMB$B_SYSTEMID          .BLKB   6       ; 48 bit SCS systemid
                                .BLKW   1       ; Spare
$DEF    VMB$L_FLAGS             .BLKL   1       ; Word of flags
$DEF    VMB$L_CI_HIPFN          .BLKL   1       ; Highest PFN used by CI code
$DEF    VMB$Q_NODENAME          .BLKQ   1       ; Booting node name
$DEF    VMB$C_ARGBYTCNT                         ; Size of argument list in bytes
$EQU    VMB$V_LOAD_SCS          0               ; Flag to SYSBOOT to load SCS
```

```
        $DEFEND VMB,GBL,ARGDEF

        .ENDM   $VMBARGDEF

;++
;
;       TIMEDWAIT - Timed Wait Loop with Imbedded Tests
;
; Macro to wait for a specified interval of time.  Uses a processor
; specific value established by system bootstrap to determine an
; approximate interval of time to wait instead of reading the
; processor clock.  Instructions that test for various exit conditions
; may be imbedded within the wait loop, if so desired.
;
; This version of TIMEDWAIT is set up to be used with boot drivers.
; It contains the right kind of PIC references to EXE$GL_TENUSEC and
; EXE$GL_UBDELAY, for code that is moved within the address space of
; an image at run-time instead of remaining bound to the relative
; offset within the image given it at link-time.  (Note that these
; two counters are kept in the BQO structure and are referenced via
; BQO$L_UBDELAY and BQO$L_TENUSEC.)
;
; INPUTS:
;
;       TIME - the number of 10 micro-second intervals to wait
;       INS1 - first instruction to imbed within wait loop
;       INS2 - second instruction to imbed within wait loop
;       INS3 - third instruction to imbed within wait loop
;       INS4 - fourth instruction to imbed within wait loop
;       INS5 - fifth instruction to imbed within wait loop
;       INS6 - sixth instruction to imbed within wait loop
;       DONELBL - label for exit from wait loop
;       IMBEDLBL - Label for imbedded instructions in wait loop
;       UBLBL - Label for UNIBUS SOBGTR loop
;
; OUTPUTS:
;
;       R0 - indicates success of failure status.  Success is defined as
;               the bit being at the specified sense within the specified
;               time interval.
;       R1 - destroyed, all other registers preserved.
;
;--

        .MACRO  TIMEDWAIT TIME,INS1,INS2,INS3,INS4,INS5,INS6,DONELBL,?IMBEDLBL,?UBLBL

        .nlist cnd
        MOVL    RPB$L_IOVEC(R9),R1      ; Get address of IOVEC data cells.
        MULL3   TIME,BQO$L_TENUSEC(R1),R1 ; Calculate time.
        MOVZWL  #SS$_NORMAL,R0          ; Assume success.
        CLRL    -(SP)                   ; Reserve space for delay loop index.
IMBEDLBL:
        'INS1'
        'INS2'
        'INS3'
```

```
        'INS4'
        'INS5'
        'INS6'
        ADDL3   RPB$L_IOVEC(R9), -      ; Get address of IOVEC data cells
                #BQO$C_UBDELAY,(SP)     ; holding delay loop cnt.
        MOVL    @0(SP),(SP)             ; Get delay loop count itself.
UBLBL:  SOBGTR  (SP),UBLBL              ; Delay loop to slow bit tests down
                                        ; to allow Unibus DMA to occur while
                                        ; testing a device register.
        SOBGTR  R1,IMBEDLBL             ; Decrement interval count
        CLRL    R0                      ; Count expired, return failure
        .IF     NOT_BLANK, DONELBL
DONELBL:
        .ENDC
        TSTL    (SP)+                   ; Pop delay loop index off stack.
        .ENDM
        .END
```

BOOTS
MAR

AUTOCONFG
LIS

WRITEBOOT
MAP

BOOCVTATB
LIS

VMB
MAP

SYSGEN
MAP

ACTIMAGE
LIS

BOOT58
LIS

BOOTDEF
SDL