N 3

LP

```
EEEEEEEEEE  XX      XX    AAAAAA    MM      MM  PPPPPPPP    LL          EEEEEEEEEE    SSSSSSSS
EEEEEEEEEE  XX      XX    AAAAAA    MM      MM  PPPPPPPP    LL          EEEEEEEEEE    SSSSSSSS
EEEEEEEEEE  XX      XX    AAAAAA    MM      MM  PPPPPPPP    LL          EEEEEEEEEE    SSSSSSSS
EE          XX      XX  AA      AA  MMMM  MMMM  PP      PP  LL          EE          SS
EE          XX      XX  AA      AA  MMMM  MMMM  PP      PP  LL          EE          SS
EE          XX      XX  AA      AA  MMMM  MMMM  PP      PP  LL          EE          SS
EE            XX  XX    AA      AA  MM  MM  MM  PP      PP  LL          EE          SS
EE            XX  XX    AA      AA  MM  MM  MM  PP      PP  LL          EE          SS
EE            XX  XX    AA      AA  MM  MM  MM  PP      PP  LL          EE          SS
EEEEEEEE        XX      AA      AA  MM      MM  PPPPPPPP    LL          EEEEEEEE      SSSSSS
EEEEEEEE        XX      AA      AA  MM      MM  PPPPPPPP    LL          EEEEEEEE      SSSSSS
EEEEEEEE        XX      AA      AA  MM      MM  PPPPPPPP    LL          EEEEEEEE      SSSSSS
EE            XX  XX    AAAAAAAAAA  MM      MM  PP          LL          EE                  SS
EE            XX  XX    AAAAAAAAAA  MM      MM  PP          LL          EE                  SS
EE            XX  XX    AAAAAAAAAA  MM      MM  PP          LL          EE                  SS
EE          XX      XX  AA      AA  MM      MM  PP          LL          EE                  SS
EE          XX      XX  AA      AA  MM      MM  PP          LL          EE                  SS
EE          XX      XX  AA      AA  MM      MM  PP          LL          EE                  SS
EEEEEEEEEE  XX      XX  AA      AA  MM      MM  PP          LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS
EEEEEEEEEE  XX      XX  AA      AA  MM      MM  PP          LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS
EEEEEEEEEE  XX      XX  AA      AA  MM      MM  PP          LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS
```

**FILE**ID**DRSLV                                    K 7                                    **F

```
DDDDDDDD    RRRRRRRR      SSSSSSSS  LL          VV      VV
DDDDDDDD    RRRRRRRR      SSSSSSSS  LL          VV      VV
DD      DD  RR      RR  SS          LL          VV      VV
DD      DD  RR      RR  SS          LL          VV      VV
DD      DD  RR      RR  SS          LL          VV      VV
DD      DD  RR      RR  SS          LL          VV      VV
DD      DD  RRRRRRRR      SSSSSS    LL          VV      VV
DD      DD  RRRRRRRR      SSSSSS    LL          VV      VV
DD      DD  RR  RR              SS  LL          VV      VV
DD      DD  RR  RR              SS  LL          VV      VV
DD      DD  RR    RR            SS  LL            VV  VV          ....
DD      DD  RR    RR            SS  LL            VV  VV          ....
DDDDDDDD    RR      RR  SSSSSSSS    LLLLLLLLLL      VV            ....
DDDDDDDD    RR      RR  SSSSSSSS    LLLLLLLLLL      VV            ....


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
```

```
        .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.                 *
;*                                                                          *
;*                                                                          *
;****************************************************************************
;

        .SBTTL  DECLARATIONS

;
; INCLUDE FILES:
;

;
; MACROS:
;

; Define the format of device messages that contain fields of a FAB

        $DEFINI DEVMSG

$DEF    DEVMSG$L_TYPE                           ;type of device message
                        .BLKL   1
$DEF    DEVMSG$L_BFRSIZ                         ;Master buffer size
                        .BLKL   1
$DEF    DEVMSG$L_ALQ                            ;allocation quantity
                        .BLKL   1
$DEF    DEVMSG$L_FOP                            ;file processing options
                        .BLKL   1
$DEF    DEVMSG$L_MRN                            ;maximum record number
                        .BLKL   1
$DEF    DEVMSG$W_DEQ                            ;default extension quantity
                        .BLKW   1
$DEF    DEVMSG$W_BLS                            ;block size
                        .BLKW   1
$DEF    DEVMSG$W_MRS                            ;maximum record size
```

```
                          .BLKW   1
 $DEF    DEVMSG$B_BKS                        ;bucket size
                          .BLKB   1
 $DEF    DEVMSG$B_FSZ
                          .BLKB   1
 $DEF    DEVMSG$B_ORG                        ;file organization
                          .BLKB   1
 $DEF    DEVMSG$B_RAT                        ;record attributes
                          .BLKB   1
 $DEF    DEVMSG$B_RFM                        ;record format
                          .BLKB   1
 $DEF    DEVMSG$B_FNS                        ;file spec string size
                          .BLKB   1
 $DEF    DEVMSG$T_FN                         ;file specification string

                          .BLKB   64
         $DEFEND DEVMSG
```

```
DTE

; I
;


;
; M
;

;
; E
;
REM
REM
CR
LF

;
; O
;

CTR


CTR

CON

CON
CON

FAI

FAI
FAI

REA
IOS
REA

USE
```

```
;
; EQUATED SYMBOLS:
;

;
; OWN STORAGE:
;
        .PSECT  SLVDATA RD,WRT,NOEXE

SLVFAB:
        $FAB    FAC = <BIO,GET,PUT>
SLVRAB:
        $RAB    FAB = SLVFAB -
                ROP = <BIO,ASY> -
                BKT = 0
        .SBTTL  OPEN/CREATE -- OPEN DISK FILE

;++
; FUNCTIONAL DESCRIPTION:
;
; SLV_CREATE creates a file that is to contain data sent by Master.
; SLV_OPEN opens an already existing file that contains data to be
;  sent to Master.
; Both  of these routines must initialize the FAB with fields sent
; over from the Master containing information about the file.
; CALLING SEQUENCE:
;
;       NONE
;
; INPUT PARAMETERS:
;
;       DEVMSG = 4              ;the device msg containing FAB field
;
; IMPLICIT INPUTS:
;
;       NONE
;
; OUTPUT PARAMETERS:
;
;       STATUS = 8              ;status of call
;
; IMPLICIT OUTPUTS:
;
;       NONE
;
; COMPLETION CODES:
;
;       NONE
;
; SIDE EFFECTS:
;
;       NONE
;
;--
```

DTE
20$
30$
40$
100

```
        .PSECT  SLVCODE         EXE,NOWRT

        .ENTRY  SLV_CREATE      ^M<R6,R7,R8,R9>
        MOVL    #1, R8                          ;"create" flag
        MOVAL   SLVFAB, R6              ;R6 <- addr of FAB
        MOVAL   SLVRAB, R9              ;R9 <- addr of RAB
        BRB     COMMON_FAB              ;join common FAB fields code

        .ENTRY  SLV_OPEN        ^M<R6,R7,R8,R9>
        CLRL    R8                      ;"open" flag
        MOVAL   SLVFAB, R6              ;R6 <- addr of FAB
        MOVAL   SLVRAB, R9              ;R9 <- addr of RAB
;
; Move the fields that are common inputs for both the $OPEN and the
; $CREATE routines from the device message into the FAB.
;

COMMON_FAB:
        MOVL    DEVMSG(AP), R7          ;R7 <- addr of device message

        MOVB    DEVMSG$B_FNS(R7), FAB$B_FNS(R6)
        MOVAL   DEVMSG$T_FN(R7), FAB$L_FNA(R6)
        BLBS    R8, CREATFILE           ;branch to create file

;
; Come here to open the file.
;

OPENFILE:
        $OPEN   FAB = (R6)
        BLBS    R0, SUCCESS
        BRB     STAT
```

```
;
; Must create the file.  Copy remaining input to $CREATE fields.
;

CREATFILE:
        MOVL    DEVMSG$L_ALQ(R7), FAB$L_ALQ(R6)
        MOVL    DEVMSG$L_FOP(R7), FAB$L_FOP(R6)
        MOVL    DEVMSG$L_MRN(R7), FAB$L_MRN(R6)
        MOVW    DEVMSG$W_BLS(R7), FAB$W_BLS(R6)
        MOVW    DEVMSG$W_DEQ(R7), FAB$W_DEQ(R6)
        MOVW    DEVMSG$W_MRS(R7), FAB$W_MRS(R6)
        MOVB    DEVMSG$B_BKS(R7), FAB$B_BKS(R6)
        MOVB    DEVMSG$B_FSZ(R7), FAB$B_FSZ(R6)
        MOVB    DEVMSG$B_ORG(R7), FAB$B_ORG(R6)
        MOVB    DEVMSG$B_RAT(R7), FAB$B_RAT(R6)
        MOVB    DEVMSG$B_RFM(R7), FAB$B_RFM(R6)

;
; Create the file.
;

        $CREATE FAB = (R6)
        BLBC    R0, STAT

SUCCESS:                                ;file open/create successful
        $CONNECT        RAB = (R9)

STAT:
        MOVL    R0, @STATUS(AP)         ;store status
        RET
        .SBTTL  _CLOSE
;++
; FUNCTIONAL DESCRIPTION:
;
;
;
        .ENTRY  SLV_CLOSE       ^M<R2,R3>
        MOVAL   SLVFAB, R3              ;R3 <- addr of FAB
        MOVAL   SLVRAB, R2              ;R2 <- addr of RAB

        $DISCONNECT     RAB = (R2)
        BLBC    R0, 10$

        $CLOSE  FAB = (R3)

10$:    RET
```

```
        .SBTTL  _READ
;++
; FUNCTIONAL DESCRIPTION:
;
;       SLV_READ is called when the slave must read a buffer
;
; CALLING SEQUENCE:
;
;       NONE
;
; INPUT PARAMETERS:
;
        BFRADR = 4              ;address of user buffer
        BFRSIZ = 8              ;size in bytes of user buffer
        SUCCOMP = 12            ;address of success completion routine
        ERRCOMP = 16            ;address of error completion routine
;
; OUTPUT PARAMETERS:
;
        STATUS = 20             ;status of call
;
; COMPLETION CODES:
;
;       NONE
;
; SIDE EFFECTS:
;
;       NONE
;
;--
        .ENTRY  SLV_READ        0
        MOVAL   SLVRAB, R1              ;R1 <- addr of RAB


;
; Copy buffer address and size into appropriate fields in RAB
;

INI_RAB:
        MOVL    BFRADR(AP), RAB$L_UBF(R1)
        MOVW    @BFRSIZ(AP), RAB$W_USZ(R1)

;
; Issue the read.
;

        $READ   RAB = (R1) -
                SUC = @SUCCOMP(AP) -
                ERR = @ERRCOMP(AP)

        MOVL    R0, @STATUS(AP)         ;store status
        RET
        .SBTTL  _WRITE
;++
; FUNCTIONAL DESCRIPTION:
;
;       SLV_WRITE is called when slave must write a buffer
;
; CALLING SEQUENCE:
;
;       NONE
;
; INPUT PARAMETERS:
;
        BFRADR = 4              ;address of user buffer
        BFRSIZ = 8              ;size of user buffer
        SUCCOMP = 12            ;success completion routine
        ERRCOMP = 16            ;error completion routine

;
; OUTPUT PARAMETERS:
;
        STATUS = 20             ;status of call
;
; COMPLETION CODES:
;
;       NONE
;
; SIDE EFFECTS:
;
;       NONE
;
;--
        .ENTRY  SLV_WRITE       0
        MOVAL   SLVRAB, R1

;
; Copy buffer address and size into appropriate fields in RAB.
;

INIT_RAB:
        MOVL    BFRADR(AP), RAB$L_RBF(R1)       ;address of user buffer
        MOVW    @BFRSIZ(AP), RAB$W_RSZ(R1)      ;size of user buffer

;
; Issue the write.
;
        $WRITE  RAB = (R1) -
                SUC = @SUCCOMP(AP) -
                ERR = @ERRCOMP(AP)

        MOVL    R0, @STATUS(AP)
        RET
        .SBTTL  GETBYTCNT
;++
; FUNCTIONAL DESCRIPTION:
;
;       Get the byte count of the most recent transfer from the RAB.
;
; INPUT PARAMETERS:
;
;       NONE
;
; OUTPUT PARAMETERS:
;
;       This is a function subroutine; it returns the byte count in R0.
;--

        .ENTRY  GETBYTCNT       0

        MOVAL   SLVRAB, R1              ;R1 <- addr of RAB
        MOVZWL  RAB$W_RSZ(R1), R0       ;# bytes read into last buffer

        RET

        .SBTTL  GETRMSTAT
;++
; Get RMS completion status
;--

        .ENTRY  GETRMSTAT       0

        MOVAL   SLVRAB, R1              ;R1 <- addr of RAB
        MOVL    RAB$L_STS(R1), R0       ;completion status

        RET
        .SBTTL  _COPYFAB
;++
; FUNCTIONAL DESCRIPTION:
;
;       SLV_COPYFAB creates a file attributes device message
;
; CALLING SEQUENCE:
;
;       NONE
;
; INPUT PARAMETERS:
;
;       DEVMSG = 4
;
; IMPLICIT INPUTS:
;
;       NONE
;
; OUTPUT PARAMETERS:
;
;       NONE
;
; IMPLICIT OUTPUTS:
;
;       NONE
;
; COMPLETION CODES:
;
;       NONE
;
; SIDE EFFECTS:
;
;       NONE
;
;--
        .ENTRY  SLV_COPYFAB     ^M<R6,R7>

        MOVAL   SLVFAB, R7              ;R7 <- addr of FAB
        MOVL    DEVMSG(AP), R6          ;R6 <- addr of DEVMSG

;
; move necessary fields from FAB into DEVMSG
;
        MO''    FAB$L_ALQ(R7), DEVMSG$L_ALQ(R6)
        MOVL    FAB$L_FOP(R7), DEVMSG$L_FOP(R6)
        MOVL    FAB$L_MRN(R7), DEVMSG$L_MRN(R6)
        MOVW    FAB$W_DEQ(R7), DEVMSG$W_DEQ(R6)
        MOVW    FAB$W_BLS(R7), DEVMSG$W_BLS(R6)
        MOVW    FAB$W_MRS(R7), DEVMSG$W_MRS(R6)
        MOVB    FAB$B_BKS(R7), DEVMSG$B_BKS(R6)
        MOVB    FAB$B_FSZ(R7), DEVMSG$B_BKS(R6)
        MOVB    FAB$B_ORG(R7), DEVMSG$B_ORG(R6)
        MOVB    FAB$B_RAT(R7), DEVMSG$B_RAT(R6)
        MOVB    FAB$B_RFM(R7), DEVMSG$B_RFM(R6)

        RET
        .SBTTL  SLV_RMSERR

SM_MSG_ERROR:
        .LONG   10

        .ENTRY  SLV_RMSERR      0
ERR_ENTRY:
        MOVAL   SLVRAB, R1
        PUSHAL  RAB$L_STS(R1)           ; error status
        PUSHAL  SM_MSG_ERROR            ; device message code
        CALLS   #2, SLV_FINISH          ; error stop slave transfer

        RET
        .END
```

0157 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LPMULT
B32

DRMAST
MAR

ADDRIVER
MAR

TDRIVER
MAR

USSTEST
MAR

GBLSECUFO
MAR

USSDISP
MAR

DOD_ERAPAT
MAR

LBRMAC
MAR

XADRIVER
MAR

LABIOCIN
MAR

DRSLV
MAR

DTE_DF03
MAR

WORKQ
LIS

SCRFT
MAR

EXAMPLES