F 13

```
MMM         MMM  TTTTTTTTTTTTTTT  HHH         HHH  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
MMM         MMM  TTTTTTTTTTTTTTT  HHH         HHH  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
MMM         MMM  TTTTTTTTTTTTTTT  HHH         HHH  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
MMMMMM   MMMMMM        TTT        HHH         HHH  RRR         RRR        TTT        LLL
MMMMMM   MMMMMM        TTT        HHH         HHH  RRR         RRR        TTT        LLL
MMMMMM   MMMMMM        TTT        HHH         HHH  RRR         RRR        TTT        LLL
MMM   MMM   MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLL
MMM   MMM   MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLL
MMM   MMM   MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLL
MMM         MMM        TTT        HHHHHHHHHHHHHHH  RRRRRRRRRRRR           TTT        LLL
MMM         MMM        TTT        HHHHHHHHHHHHHHH  RRRRRRRRRRRR           TTT        LLL
MMM         MMM        TTT        HHHHHHHHHHHHHHH  RRRRRRRRRRRR           TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR   RRR              TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR   RRR              TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR   RRR              TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR      RRR           TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR      RRR           TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR      RRR           TTT        LLL
MMM         MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLLLLLLLLLLLLLL
MMM         MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLLLLLLLLLLLLLL
MMM         MMM        TTT        HHH         HHH  RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
**FILE**ID**MTHSIGNAL                                    I 10





MM      MM  TTTTTTTTTT  HH      HH    SSSSSSSS    IIIIII      GGGGGGGG  NN      NN    AAAAAA    LL
MM      MM  TTTTTTTTTT  HH      HH    SSSSSSSS    IIIIII      GGGGGGGG  NN      NN    AAAAAA    LL
MMMM  MMMM      TT      HH      HH  SS              II      GG          NN      NN  AA      AA  LL
MMMM  MMMM      TT      HH      HH  SS              II      GG          NN      NN  AA      AA  LL
MM  MM  MM      TT      HH      HH  SS              II      GG          NNNN    NN  AA      AA  LL
MM  MM  MM      TT      HH      HH  SS              II      GG          NNNN    NN  AA      AA  LL
MM      MM      TT      HHHHHHHHHH    SSSSSS        II      GG          NN  NN  NN  AA      AA  LL
MM      MM      TT      HHHHHHHHHH    SSSSSS        II      GG          NN  NN  NN  AA      AA  LL
MM      MM      TT      HH      HH          SS      II      GG  GGGGGG  NN    NNNN  AAAAAAAAAA  LL
MM      MM      TT      HH      HH          SS      II      GG  GGGGGG  NN    NNNN  AAAAAAAAAA  LL
MM      MM      TT      HH      HH          SS      II      GG      GG  NN      NN  AA      AA  LL                ....
MM      MM      TT      HH      HH          SS      II      GG      GG  NN      NN  AA      AA  LL                ....
MM      MM      TT      HH      HH  SSSSSSSS      IIIIII      GGGGGG    NN      NN  AA      AA  LLLLLLLLLL      ....
MM      MM      TT      HH      HH  SSSSSSSS      IIIIII      GGGGGG    NN      NN  AA      AA  LLLLLLLLLL      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
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
```

Table of contents

```
0000            .TITLE MTH$$SIGNAL       ; MATH ERROR SIGNAL ROUTINE
0000            .IDENT  /1-003/          ; File: MTHSIGNAL.MAR
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ;++
0000    ; FACILITY:  Mathematical Library
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ;       SIGNAL math error as SEVERE, continue will return reserved operand
0000    ;
0000    ; ENVIRONMENT:
0000    ;
0000    ;--
0000    ;
```

```
0000          .SBTTL  HISTORY                 ; DETAILED
0000  ;
0000  ; AUTHOR:       P. Yuo, CREATION DATE: 28-Sep-77
0000  ;
0000  ; MODIFIED BY:
0000  ;       0-1, : VERSION
0000  ; 0-2   - Removed incorrect MTH$_abcmnoxyz defns.  TNH 3-Nov-77
0000  ; 0-3   - Reference previous frame properly.  TNH 10-Nov-77
0000  ; 0-4   - Signal SEVERE, set subsystem specific bit.  TNH 20-Dec-77
0000  ; 0-5   - Include FAO arg count.  TNH 10-Jan-78
0000  ; 0-6   - Remove all debugging FAO stuff.  JMT 17-Jan-78
0000  ; 0-7   - Make fake frame if JSB, so traceback same as CALL.  TNH 18-Apr-78
0000  ; 1-001 - Update version number and copyright information.  JBS 16-NOV-78
0000  ; 1-002 - Add "_" to the PSECT directive.  JBS 22-DEC-78
0000  ; 1-003 - Change shared external references to G^ RNH 25-Sep-81
```

```
                    0000           .SBTTL  DECLARATION
                    0000
                    0000   ; INCLUDE FILES:        NONE
                    0000   ;
                    0000   ; EXTERNAL SYMBOLS:
                    0000   ;
                    0000
                    0000           .DSABL  GBL                     ; Declare all externals explicitly
                    0000           .EXTRN  LIB$SIGNAL              ; Signal exception
                    0000
                    0000   ;
                    0000   ; MACROS:
                    0000   ;
                    0000           $MTHERR                         ; MTH$ symbols
                    0000           $SFDEF                          ; stack frame symbols
                    0000           $SSDEF                          ; STARLET completion codes
                    0000           $STSDEF                         ; condition value symbols
                    0000
                    0000   ; EQUATED SYMBOLS:
                    0000   ;
          00000004  0000           ERROR_CODE = 4                  ; error code is first formal
          00000008  0000           JSB_PC = 8                      ; JSB PC (optional)
                    0000
                    0000   ; PSECT DECLARATIONS:
                    0000   ;
                    0000
                00000000           .PSECT  _MTH$CODE PIC,SHR,LONG,EXE,NOWRT
                    0000                                           ; program section for MTH$ code
                    0000
```

```
                          0000        .SBTTL  MTH$$JACKET_HND          ; Math routine error handler
                          0000
                    0000  0000        .ENTRY  MTH$$JACKET_HND,0
    50    0918 8F     3C  0002        MOVZWL  #SS$_RESIGNAL, R0        ; just resignal
                      04  0007        RET
                          0008
```

```
0008           .SBTTL  MTH$$SIGNAL                ; Signal any math error
0008
0008    ;++
0008    ; FUNCTIONAL DESCRIPTION:
0008    ;
0008    ;       Algorithmic steps:
0008    ;       1) R0/R1 contain results from math routine and will be passed as
0008    ;          implicit input to LIB$SIGNAL. LIB$SIGNAL will save R0/R1 in signal
0008    ;          mechanism vector, CHF$L_MCH_R0/R1, so that when (user's or any)
0008    ;          error handler get called CHF$L_MCH_R0/R1 may be modified.
0008    ;       2) Save implicit input R0/R1 on stack for later.
0008    ;       3) If called with only one argument then the math routine must have
0008    ;          been called with CALL rathen then JSB.
0008    ;       4) If user called math routine with JSB, create extra stack
0008    ;          frame and fill in return PCs so looks like user had CALLed instead.
0008    ;          Get correct user PC depending on whether CALL or JSB.
0008    ;       5) Convert MTH$ error code to 32-bit VAX error code.
0008    ;               Set bits 31:16 to MTH$ facility code.
0008    ;               Shift small error code left 3 places to make room for severity code
0008    ;       6) Signal condition as SEVERE (CALL LIB$SIGNAL).
0008    ;       [NOTE: DO NOT RESTORE R0/R1]
0008    ;       7) If handler wishes to continue, return to caller (R0/R1 restored from
0008    ;          CHF$L_MCH_R0/R1)
0008    ;
0008    ; CALLING SEQUENCE:
0008    ;
0008    ;       CALL    MTH$SIGNAL (error_message.rl.v [,call_pc.rl.v])
0008    ;
0008    ; INPUT PARAMETERS:
0008    ;
0008    ;       error_message.rl.v              ; math error message
0008    ;       call_pc.rl.v                    ; present only if math routine is called
0008    ;                                       ; by JSB
0008    ;
0008    ; IMPLICIT INPUTS:
0008    ;
0008    ;       R0:R1 - passed to LIB$SIGNAL to be put in signal mechanism vector
0008    ;               so user's or any error handler can fixup.
0008    ;
0008    ; OUTPUT PARAMETERS:
0008    ;       NONE
0008    ;
0008    ; IMPLICIT OUTPUTS:
0008    ;       NONE
0008    ;
0008    ; COMPLETION CODES:
0008    ;       NONE
0008    ;
0008    ; SIDE EFFECTS:
0008    ;       Signal error message.
0008    ;
0008    ;--
0008
```

```
                 000C  0008        .ENTRY  MTH$$SIGNAL, ^M<R2,R3>  ; save R2:R3
                       000A                                        ; but do not use R0:R1, (passed as
                       000A                                        ; implicit input to LIB$SIGNAL) and
                       000A                                        ; DO NOT RESTORE R0/R1, since they
                       000A                                        ; may have been modified by error
                       000A                                        ; handling routine.
                       000A
                       000A ;+
                       000A ; Check number of arguments to see if math routine is called by CALL or JSB
                       000A ;-
                       000A
      01     6C    91  000A        CMPB    (AP), #1                ; if called with only one arg
             0E    13  000D        BEQLU   5$                      ; assume mth routine called with CALL
                       000F                                        ; (rather than JSB)
                       000F
                       000F ;+
                       000F ; If math routine is called by JSB then check if its caller is a math
                       000F ; routine or the user.
                       000F ;-
                       000F
52        08 AC    D0  000F        MOVL    JSB_PC(AP), R2          ; R2 = JSB PC
53        EA AF    DE  0013        MOVAL   B^MTH$$JACKET_HND, R3   ; R3 = absolute adr. of handler
                       0017                                        ; setup by each CALL MTH$
                       0017                                        ; (as opposed to JSB MTH$)
53        0C BD    D1  0017        CMPL    @SF$L_SAVE_FP(FP), R3   ; compare prev. frame handler
             0B    12  001B        BNEQ    10$                     ; branch if not jacket handler
                       001D
                       001D ;+
                       001D ; Previous frame had established MTH$$JACKET_HND as a handler
                       001D ; so it must have been math routine called with CALL (not JSB)
                       001D ; Get user PC stored by CALL instruction
                       001D ;-
                       001D
52        0C AD    D0  001D 5$:    MOVL    SF$L_SAVE_FP(FP), R2    ; R2 = previous frame adr.
                       0021                                        ; i.e., math routine frame
52        10 A2    D0  0021        MOVL    SF$L_SAVE_PC(R2), R2    ; R2 = saved user's PC
             24    10  0025        BSBB    DO_SIGNAL               ; signal error code 4(AP) and
                       0027                                        ; user PC (R2)
                   04  0027        RET                             ; return to caller (math routine)
                       0028
                       0028 ;+
                       0028 ; User called math routine with JSB - create extra frame
                       0028 ; so that stack looks just like user had called math routine with CALL.
                       0028 ; Then traceback and stack depth (handler argument) will be same
                       0028 ; for JSB and CALL.
                       0028 ;-
                       0028
53        10 AD    D0  0028 10$:   MOVL    SF$L_SAVE_PC(FP), R3    ; R3 = return PC to math routine
             53    DD  002C        PUSHL   R3                      ; save return PC to math routine
10 AD        52    D0  002E        MOVL    R2, SF$L_SAVE_PC(FP)    ; set return PC in frame to be user JSB PC
3B'AF        6C    FA  0032        CALLG   (AP), B^SIGNAL[1        ; create another frame and signal
10 AD        8E    D0  0036        MOVL    (SP)+, SF$L_SAVE_PC(FP) ; restore return PC to math routine
                   04  003A        RET                             ; return to caller (math routine)
```

```
                     003B          .SBTTL  SIGNAL1               ; Local signal routine (when user JSB)
                     003B
                     003B  ;++
                     003B  ; FUNCTIONAL DESCRIPTION:
                     003B  ;
                     003B  ;       Move return PCs up a frame and signal error
                     003B  ;
                     003B  ; CALLING SEQUENCE:
                     003B  ;
                     003B  ;       MOVL    user JSB PC, R2
                     003B  ;       MOVL    return PC to math routine, R3
                     003B  ;       CALL    SIGNAL1 (error_message.rl.v [,ignored])
                     003B  ;
                     003B  ; INPUT PARAMETERS:
                     003B  ;
                     003B  ;       error_message.rl.v            ; math error message
                     003B  ;
                     003B  ; IMPLICIT INPUTS:
                     003B  ;
                     003B  ;       R0/R1 - passed to LIB$SIGNAL to be put in signal mechanism vector
                     003B  ;               so user's or any error handler can fixup.
                     003B  ;       R2    - user JSB PC
                     003B  ;       R3    - return PC to math routine
                     003B  ;
                     003B  ; OUTPUT PARAMETERS:
                     003B  ;       NONE
                     003B  ;
                     003B  ; IMPLICIT OUTPUTS:
                     003B  ;       NONE
                     003B  ;
                     003B  ; COMPLETION CODES:
                     003B  ;       NONE
                     003B  ;
                     003B  ; SIDE EFFECTS:
                     003B  ;       Signal error message
                     003B  ;--
                     003B
                     003B SIGNAL1:
               0000  003B          .WORD ^M<>                    ; save nothing - access R0:R3
         10 AD   DD  003D          PUSHL   SF$L_SAVE_PC(FP)      ; save return PC to MTH$$SIGNAL
   10 AD   53    D0  0040          MOVL    R3, SF$L_SAVE_PC(FP)  ; set return PC to be math routine
           05    10  0044          BSBB    DO_SIGNAL             ; signal using error code in R2
   10 AD   8E    D0  0046          MOVL    (SP)+, SF$L_SAVE_PC(FP) ; restore return PC to MTH$$SIGNAL
                 04  004A          RET                           ; return to caller (MTH$$SIGNAL)
```

```
                                  004B          .SBTTL  DO_SIGNAL               ; Local JSB routine to do signal
                                  004B
                                  004B  ;++
                                  004B  ; FUNCTIONAL DESCRIPTION:
                                  004B  ;
                                  004B  ;       Convert MTH$ error code to 32-bit VAX error code.
                                  004B  ;       Set bits 31:16 to MTH$ facility code
                                  004B  ;       Shift small error code left 3 places to make room for severity code
                                  004B  ;       and then call LIB$SIGNAL with implicit input in R0/R1 (= math routine
                                  004B  ;       result.  Then perform the following call to do signal:
                                  004B  ;
                                  004B  ;       CALL LIB$SIGNAL (cond_val, 1, user_PC)
                                  004B  ;
                                  004B  ; CALLING SEQUENCE:
                                  004B  ;
                                  004B  ;       JSB     DO_SIGNAL
                                  004B  ;
                                  004B  ; INPUT PARAMETERS:
                                  004B  ;
                                  004B  ;       error_code.rlu.v                small math error number
                                  004B  ;
                                  004B  ; IMPLICIT INPUTS:
                                  004B  ;
                                  004B  ;       R0/R1   Math routine function value to be copied to CHF$L_MCH_R0/R1
                                  004B  ;       R2      User PC to be used in message
                                  004B  ;
                                  004B  ; OUTPUT PARAMETERS:
                                  004B  ;       NONE
                                  004B  ;
                                  004B  ; IMPLICIT OUTPUTS:
                                  004B  ;       NONE
                                  004B  ;
                                  004B  ; COMPLETION CODES:
                                  004B  ;       NONE
                                  004B  ;
                                  004B  ; SIDE EFFECTS:
                                  004B  ;
                                  004B  ;       Signal error
                                  004B  ;--
                                  004B
                                  004B  DO_SIGNAL:                              ; Local JSB entry point
                         52   DD  004B          PUSHL   R2                      ; setup last arg as user PC
                         01   DD  004D          PUSHL   #1                      ; indicate that one FAO arg is following
                                  004F                                          ; so SYS$PUT_MESSAGE will know.
             7E    16    10   78  004F          ASHL    #16, #MTH$K_FAC_NO, -(SP)
                                  0053                                          ; MTH$ error prefix code to LH
                   6E    04   90  0053          MOVB    #STS$K_SEVERE, (SP)     ; set error severity to SEVERE
                                  0056                                          ; so image will EXIT unless user handles.
     6E    0C    03    04 AC  F0  0056          INSV    ERROR_CODE(AP), #STS$V_CODE, -
                                  005C                  #STS$S_CODE, (SP)       ; insert math code shifted left
             6E    8000 8F    A8  005C          BISW    #STS$M_FAC_SP, (SP)     ; Set subsystem specific message bit
                                  0061                                          ; 3 bits to make room for severity
        00000000'GF      03   FB  0061          CALLS   #3, G^LIB$SIGNAL        ; save R0/R1 in signal mechanism
                                  0068                                          ; vector (CHF$L_MCH_R0/R1)
                                  0068                                          ; return with R0/R1 = signal mechanism
                                  0068                                          ; vector CHF$L_MCH_R0/R1 which any
                                  0068                                          ; error handler may have modified
```

```
                             05  0068   303         RSB                                   ; return
```

```
                      0069            .SBTTL  MTH$$JACKET_TST           ; Routine to test for math jacket handler
                      0069
                0000  0069            .ENTRY  MTH$$JACKET_TST,0
          50      D4  006B            CLRL    R0                        ; assume not jacket routine
                      006D                                              ; setup as handler in previous fram
                      006D                                              ; i.e., JSB to math routine
51    90 AF       DE  006D            MOVAL   MTH$$JACKET_HND, R1       ; R1 = adr. of handler set up by
                      0071                                              ; CALL MTH$xxx.
0C BD     51      D1  0071            CMPL    R1, @SF$L_SAVE_FP(FP)     ; test previous frame handler
          03      12  0075            BNEQ    10$                       ; branch if not jacket handler
   50     01      D0  0077            MOVL    #1, R0                    ; return TRUE
                  04  007A    10$:    RET                               ; return
                      007B
```

```
                 007B        .SBTTL  MTH$$SIGNAL_CON          ; Error without signaling and continue
                 007B
           0000  007B        .ENTRY  MTH$$SIGNAL_CON, 0
  50   01    D0  007D        MOVL    #SS$_CONTINUE, R0        ; error condition and continue
             04  0080        RET
                 0081
                 0081
                 0081        .END                             ; end of module
```

```
DO_SIGNAL                          0000004B R      02
ERROR_CODE                       = 00000004
JSB_PC                           = 00000008
LIB$SIGNAL                         ********    X   00
MTH$$JACKET_HND                    00000000 RG     02
MTH$$JACKET_TST                    00000069 RG     02
MTH$$SIGNAL                        00000008 RG     02
MTH$$SIGNAL_CON                    0000007B RG     02
MTH$K_FAC_NO                     = 00000016
SF$L_SAVE_FP                     = 0000000C
SF$L_SAVE_PC                     = 00000010
SIGNAL1                            0000003B R      02
SS$_CONTINUE                     = 00000001
SS$_RESIGNAL                     = 00000918
STS$K_SEVERE                     = 00000004
STS$M_FAC_SP                     = 00008000
STS$S_CODE                       = 0000000C
STS$V_CODE                       = 00000003


                                                   +-----------------+
                                                   ! 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
_MTH$CODE                          00000081  (   129.)  02 (  2.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD  NOWRT NOVEC LONG


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

Phase                       Page faults    CPU Time       Elapsed Time
-----                       -----------    --------       ------------
Initialization                      29     00:00:00.08    00:00:00.86
Command processing                 122     00:00:00.63    00:00:03.56
Pass 1                             217     00:00:04.94    00:00:15.55
Symbol table sort                    0     00:00:00.69    00:00:01.30
Pass 2                              70     00:00:01.10    00:00:03.71
Symbol table output                  4     00:00:00.03    00:00:00.06
Psect synopsis output                2     00:00:00.02    00:00:00.37
Cross-reference output               0     00:00:00.00    00:00:00.00
Assembler run totals               446     00:00:07.50    00:00:25.42

The working set limit was 1200 pages.
25872 bytes (51 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 486 non-local and 3 local symbols.
326 source lines were read in Pass 1, producing 22 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:[MTHRTL.OBJ]MTHRTL.MLB;1                 1
_$255$DUA28:[SYSLIB]STARLET.MLB;2                    6
TOTALS (all libraries)                               7

549 GETS were required to define 7 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:MTHSIGNAL/OBJ=OBJ$:MTHSIGNAL MSRC$:MTHSIGNAL/UPDATE=(ENH$:MTHSIGNAL)+LI
```

0263 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY