

[illegible]

3

Sy  
--  
MT  
MT  
MT

[illegible]

MT  
MT  
MT  
MT  
MT  
MT  
MT

MT  
MT  
MT  
MT  
MT  
MT  
MT

[illegible]

|      |     |                   |                                           |
|------|-----|-------------------|-------------------------------------------|
| (2)  | 40  | HISTORY           | ; DETAILED                                |
| (3)  | 56  | DECLARATION       |                                           |
| (4)  | 86  | MTH\$\$JACKET_HND | ; Math routine error handler              |
| (5)  | 93  | MTH\$\$SIGNAL     | ; Signal any math error                   |
| (7)  | 201 | SIGNAL1           | ; Local signal routine (when user JSB)    |
| (8)  | 246 | DO SIGNAL         | ; Local JSB routine to do signal          |
| (9)  | 305 | MTH\$\$JACKET_TST | ; Routine to test for math jacket handler |
| (10) | 319 | MTH\$\$SIGNAL_CON | ; Error without signaling and continue    |



```
0000 1      .TITLE MTH$SIGNAL      ; MATH ERROR SIGNAL ROUTINE
0000 2      .IDENT /1-003/        ; File: MTHSIGNAL.MAR
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 : FACILITY: Mathematical Library
0000 30 :
0000 31 : ABSTRACT:
0000 32 :
0000 33 :     SIGNAL math error as SEVERE, continue will return reserved operand
0000 34 :
0000 35 : ENVIRONMENT:
0000 36 :
0000 37 :--
0000 38 :
```

MTH\$SIGNAL  
1-003

; MATH ERROR SIGNAL ROUTINE  
HISTORY ; DETAILED

L 10

16-SEP-1984 01:49:10 VAX/VMS Macro V04-00  
6-SEP-1984 11:26:55 [MTHRTL.SRC]MTHSIGNAL.MAR;1

Page 2  
(2)

```
0000 40      .SBTTL HISTORY                ; DETAILED
0000 41      :
0000 42      : AUTHOR:      P. Yuo, CREATION DATE: 28-Sep-77
0000 43      :
0000 44      : MODIFIED BY:
0000 45      : 0-1, : VERSION
0000 46      : 0-2 - Removed incorrect MTH$ abcmnoxyz defns.  TNH 3-Nov-77
0000 47      : 0-3 - Reference previous frame properly.    TNH 10-Nov-77
0000 48      : 0-4 - Signal SEVERE, set subsystem specific bit.  TNH 20-Dec-77
0000 49      : 0-5 - Include FAO arg count.    TNH 10-Jan-78
0000 50      : 0-6 - Remove all debugging FAO stuff.  JMT 17-Jan-78
0000 51      : 0-7 - Make fake frame if JSB, so traceback same as CALL.  TNH 18-Apr-78
0000 52      : 1-001 - Update version number and copyright information.  JBS 16-NOV-78
0000 53      : 1-002 - Add "" to the PSECT directive.  JBS 22-DEC-78
0000 54      : 1-003 - Change shared external references to G^ RNH 25-Sep-81
```

```
0000 56          .SBTTL  DECLARATION
0000 57
0000 58 : INCLUDE FILES:      NONE
0000 59 :
0000 60 : EXTERNAL SYMBOLS:
0000 61 :
0000 62 :
0000 63          .DSABL  GBL          : Declare all externals explicitly
0000 64          .EXTRN  LIB$SIGNAL   : Signal exception
0000 65
0000 66 :
0000 67 : MACROS:
0000 68 :
0000 69          $MTHERR              : MTH$ symbols
0000 70          $$FDEF              : stack frame symbols
0000 71          $$$DEF              : STARLET completion codes
0000 72          $$SDEF              : condition value symbols
0000 73
0000 74 : EQUATED SYMBOLS:
0000 75 :
00000004 0000 76          ERROR_CODE = 4          : error code is first formal
00000008 0000 77          JSB_PC = 8              : JSB PC (optional)
0000 78
0000 79 : PSECT DECLARATIONS:
0000 80 :
0000 81
00000000 82          .PSECT  _MTH$CODE PIC,SHR,LONG,EXE,NOWRT
0000 83          : program section for MTH$ code
0000 84
```



MTH\$\$SIGNAL  
1-003

; MATH ERROR SIGNAL ROUTINE

N 10

16-SEP-1984 01:49:10

VAX/VMS Macro V04-00

Page

4

MTH\$\$JACKET\_HND

; Math routine error ha

6-SEP-1984 11:26:55

[MTHRTL.SRC]MTHSIGNAL.MAR;1

(4)

|    |      |    |      |      |    |        |                     |                              |
|----|------|----|------|------|----|--------|---------------------|------------------------------|
| 50 | 0918 | 8F | 0000 | 0000 | 86 | .SBTTL | MTH\$\$JACKET_HND   | ; Math routine error handler |
|    |      |    | 3C   | 0000 | 87 |        |                     |                              |
|    |      |    | 04   | 0002 | 88 | .ENTRY | MTH\$\$JACKET_HND,0 |                              |
|    |      |    |      | 0007 | 89 | MOVZWL | #SS\$_RESIGNAL, R0  | ; just resignal              |
|    |      |    |      | 0008 | 90 | RET    |                     |                              |
|    |      |    |      |      | 91 |        |                     |                              |

```

0008 93      .SBTTL MTH$$SIGNAL      ; Signal any math error
0008 94
0008 95      :++
0008 96      : FUNCTIONAL DESCRIPTION:
0008 97      :
0008 98      :   Algorithmic steps:
0008 99      :   1) R0/R1 contain results from math routine and will be passed as
0008 100     :       implicit input to LIB$$SIGNAL. LIB$$SIGNAL will save R0/R1 in signal
0008 101     :       mechanism vector, CHF$MCH_R0/R1, so that when (user's or any)
0008 102     :       error handler get called CHF$MCH_R0/R1 may be modified.
0008 103     :   2) Save implicit input R0/R1 on stack for later.
0008 104     :   3) If called with only one argument then the math routine must have
0008 105     :       been called with CALL rather than JSB.
0008 106     :   4) If user called math routine with JSB, create extra stack
0008 107     :       frame and fill in return PCs so looks like user had CALLED instead.
0008 108     :       Get correct user PC depending on whether CALL or JSB.
0008 109     :   5) Convert MTH$ error code to 32-bit VAX error code.
0008 110     :       Set bits 31:16 to MTH$ facility code.
0008 111     :       Shift small error code left 3 places to make room for severity code
0008 112     :   6) Signal condition as SEVERE (CALL LIB$$SIGNAL).
0008 113     :   [NOTE: DO NOT RESTORE R0/R1]
0008 114     :   7) If handler wishes to continue, return to caller (R0/R1 restored from
0008 115     :       CHF$MCH_R0/R1)
0008 116
0008 117     : CALLING SEQUENCE:
0008 118
0008 119     :   CALL MTH$$SIGNAL (error_message.rl.v [,call_pc.rl.v])
0008 120
0008 121     : INPUT PARAMETERS:
0008 122
0008 123     :   error_message.rl.v      ; math error message
0008 124     :   call_pc.rl.v           ; present only if math routine is called
0008 125     :                       ; by JSB
0008 126
0008 127     : IMPLICIT INPUTS:
0008 128
0008 129     :   R0:R1 - passed to LIB$$SIGNAL to be put in signal mechanism vector
0008 130     :       so user's or any error handler can fixup.
0008 131
0008 132     : OUTPUT PARAMETERS:
0008 133     :   NONE
0008 134
0008 135     : IMPLICIT OUTPUTS:
0008 136     :   NONE
0008 137
0008 138     : COMPLETION CODES:
0008 139     :   NONE
0008 140
0008 141     : SIDE EFFECTS:
0008 142     :   Signal error message.
0008 143
0008 144     :--
0008 145

```



```
000C 0008 147 .ENTRY MTH$$SIGNAL, ^M<R2,R3> ; save R2:R3
      000A 148 ; but do not use R0:R1, (passed as
      000A 149 ; implicit input to LIB$$SIGNAL) and
      000A 150 ; DO NOT RESTORE R0/R1, since they
      000A 151 ; may have been modified by error
      000A 152 ; handling routine.
      000A 153
      000A 154 ;+
      000A 155 ; Check number of arguments to see if math routine is called by CALL or JSB
      000A 156 ; -
      000A 157
01 6C 91 000A 158 CMPB (AP), #1 ; if called with only one arg
OE 13 000D 159 BEQLU 5$ ; assume mth routine called with CALL
      000F 160 ; (rather than JSB)
      000F 161
      000F 162 ;+
      000F 163 ; If math routine is called by JSB then check if its caller is a math
      000F 164 ; routine or the user.
      000F 165 ; -
      000F 166
52 08 AC DO 000F 167 MOVL JSB_PC(AP), R2 ; R2 = JSB PC
53 EA AF DE 0013 168 MOVAL B^MTH$$JACKET_HND, R3 ; R3 = absolute adr. of handler
      0017 169 ; setup by each CALL MTH$
      0017 170 ; (as opposed to JSB MTH$)
53 0C BD D1 0017 171 CMPL @SF$L_SAVE_FP(FP), R3 ; compare prev. frame handler
      OB 12 001B 172 BNEQ 10$ ; branch if not jacket handler
      001D 173
      001D 174 ;+
      001D 175 ; Previous frame had established MTH$$JACKET_HND as a handler
      001D 176 ; so it must have been math routine called with CALL (not JSB)
      001D 177 ; Get user PC stored by CALL instruction
      001D 178 ; -
      001D 179
52 0C AD DO 001D 180 5$: MOVL SF$L_SAVE_FP(FP), R2 ; R2 = previous frame adr.
      0021 181 ; i.e., math routine frame
52 10 A2 DO 0021 182 MOVL SF$L_SAVE_PC(R2), R2 ; R2 = saved user's PC
      24 10 0025 183 BSBB DO_SIGNAL ; signal error code 4(AP) and
      0027 184 ; user PC (R2)
      04 0027 185 RET ; return to caller (math routine)
      0028 186
      0028 187 ;+
      0028 188 ; User called math routine with JSB - create extra frame
      0028 189 ; so that stack looks just like user had called math routine with CALL.
      0028 190 ; Then traceback and stack depth (handler argument) will be same
      0028 191 ; for JSB and CALL.
      0028 192 ; -
      0028 193
53 10 AD DO 0028 194 10$: MOVL SF$L_SAVE_PC(FP), R3 ; R3 = return PC to math routine
      53 DD 002C 195 PUSHL R3 ; save return PC to math routine
10 AD 52 DO 002E 196 MOVL R2, SF$L_SAVE_PC(FP) ; set return PC in frame to be user JSB PC
3B AF 6C FA 0032 197 CALLG (AP), B^SIGNAL1 ; create another frame and signal
10 AD 8E DO 0036 198 MOVL (SP)+, SF$L_SAVE_PC(FP) ; restore return PC to math routine
      04 003A 199 RET ; return to caller (math routine)
```

```

; MATH ERROR SIGNAL ROUTINE
SIGNAL1 ; Local signal routine (when u

003B 201          .SBTTL SIGNAL1          ; Local signal routine (when user JSB)
003B 202
003B 203      ;++
003B 204      : FUNCTIONAL DESCRIPTION:
003B 205      :
003B 206      :     Move return PCs up a frame and signal error
003B 207      :
003B 208      : CALLING SEQUENCE:
003B 209      :
003B 210      :     MOVL    user JSB PC, R2
003B 211      :     MOVL    return PC to math routine, R3
003B 212      :     CALL    SIGNAL1 (error_message.rl.v [,ignored])
003B 213      :
003B 214      : INPUT PARAMETERS:
003B 215      :
003B 216      :     error_message.rl.v          ; math error message
003B 217      :
003B 218      : IMPLICIT INPUTS:
003B 219      :
003B 220      :     R0/R1 - passed to LIB$$SIGNAL to be put in signal mechanism vector
003B 221      :             so user's or any error handler can fixup.
003B 222      :     R2    - user JSB PC
003B 223      :     R3    - return PC to math routine
003B 224      :
003B 225      : OUTPUT PARAMETERS:
003B 226      :     NONE
003B 227      :
003B 228      : IMPLICIT OUTPUTS:
003B 229      :     NONE
003B 230      :
003B 231      : COMPLETION CODES:
003B 232      :     NONE
003B 233      :
003B 234      : SIDE EFFECTS:
003B 235      :     Signal error message
003B 236      : --
003B 237
003B 238 SIGNAL1:
003B 239      .WORD ^M<>          ; save nothing - access R0:R3
10 AD 10 AD 0000 003B 240      PUSHL SF$$_SAVE_PC(FP)      ; save return PC to MTH$$SIGNAL
10 AD 53 DO 0040 241      MOVL R3, SF$$_SAVE_PC(FP)      ; set return PC to be math routine
10 AD 05 10 0044 242      BSBB DO SIGNAL1          ; signal using error code in R2
10 AD 8E DO 0046 243      MOVL (SP)+, SF$$_SAVE_PC(FP)    ; restore return PC to MTH$$SIGNAL
04 004A 244      RET          ; return to caller (MTH$$SIGNAL)
```



```
004B 246 .SBTTL DO_SIGNAL ; Local JSB routine to do signal
004B 247
004B 248 :++
004B 249 : FUNCTIONAL DESCRIPTION:
004B 250 :
004B 251 : Convert MTH$ error code to 32-bit VAX error code.
004B 252 : Set bits 31:16 to MTH$ facility code
004B 253 : Shift small error code left 3 places to make room for severity code
004B 254 : and then call LIB$SIGNAL with implicit input in R0/R1 (= math routine
004B 255 : result. Then perform the following call to do signal:
004B 256 :
004B 257 : CALL LIB$SIGNAL (cond_val, 1, user_PC)
004B 258 :
004B 259 : CALLING SEQUENCE:
004B 260 :
004B 261 : JSB DO_SIGNAL
004B 262 :
004B 263 : INPUT PARAMETERS:
004B 264 :
004B 265 : error_code.rlu.v small math error number
004B 266 :
004B 267 : IMPLICIT INPUTS:
004B 268 :
004B 269 : R0/R1 Math routine function value to be copied to CHF$MCH_R0/R1
004B 270 : R2 User PC to be used in message
004B 271 :
004B 272 : OUTPUT PARAMETERS:
004B 273 : NONE
004B 274 :
004B 275 : IMPLICIT OUTPUTS:
004B 276 : NONE
004B 277 :
004B 278 : COMPLETION CODES:
004B 279 : NONE
004B 280 :
004B 281 : SIDE EFFECTS:
004B 282 :
004B 283 : Signal error
004B 284 :--
004B 285
004B 286 DO_SIGNAL: ; Local JSB entry point
004B 287 PUSH R2 ; setup last arg as user PC
004B 288 PUSH #1 ; indicate that one FAO arg is following
004B 289 ; so SY$PUT_MESSAGE will know.
004B 290 ASHL #16, #MTH$K_FAC_NO, -(SP) ; MTH$ error prefix code to LH
004B 291 ; set error severity to SEVERE
004B 292 MOV B #ST$K_SEVERE, (SP) ; so image will EXIT unless user handles.
004B 293 ; insert math code shifted left
004B 294 INSV ERROR_CODE(AP), #ST$V_CODE, - ; Set subsystem specific message bit
004B 295 #ST$S_CODE, (SP) ; 3 bits to make room for severity
004B 296 BISW #ST$M_FAC_SP, (SP) ; save R0/R1 in signal mechanism
004B 297 ; vector (CHF$MCH_R0/R1)
004B 298 CALLS #3, G^LIB$SIGNAL ; return with R0/R1 = signal mechanism
004B 299 ; vector CHF$MCH_R0/R1 which any
004B 300 ; error handler may have modified
004B 301
004B 302
```

|          |     |      |     |       |      |                                              |
|----------|-----|------|-----|-------|------|----------------------------------------------|
| 52       | DD  | 004B | 287 | PUSHL | R2   |                                              |
| 01       | DD  | 004D | 288 | PUSHL | #1   |                                              |
| 7E       | 16  | 10   | 78  | 004F  | 289  | ASHL #16, #MTH\$K_FAC_NO, -(SP)              |
|          | 6E  | 04   | 90  | 0053  | 291  | MOV B #ST\$K_SEVERE, (SP)                    |
| 6E       | 0C  | 03   | 04  | AC    | F0   | 0056 293 INSV ERROR_CODE(AP), #ST\$V_CODE, - |
|          | 6E  | 8000 | 8F  | A8    | 005C | 295 #ST\$S_CODE, (SP)                        |
| 00000000 | 'GF | 03   | FB  | 0061  | 297  | BISW #ST\$M_FAC_SP, (SP)                     |
|          |     |      |     | 0061  | 298  | CALLS #3, G^LIB\$SIGNAL                      |
|          |     |      |     | 0068  | 299  |                                              |
|          |     |      |     | 0068  | 300  |                                              |
|          |     |      |     | 0068  | 301  |                                              |
|          |     |      |     | 0068  | 302  |                                              |



MTHSSIGNAL  
1-003

F 11  
; MATH ERROR SIGNAL ROUTINE  
DO\_SIGNAL ; Local JSB routine to do sig  
05 0068 303 RSB  
; return

16-SEP-1984 01:49:10 VAX/VMS Macro V04-00  
6-SEP-1984 11:26:55 [MTHRTL.SRC]MTHSIGNAL.MAR;1 Page 9  
(8)

```
0069 305 .SBTTL MTH$$JACKET_TST ; Routine to test for math jacket handler
0069 306
50 0000 0069 307 .ENTRY MTH$$JACKET_TST,0
D4 006B 308 CLRL R0 ; assume not jacket routine
006D 309 ; setup as handler in previous fram
006D 310 ; i.e., JSB to math routine
51 90 AF DE 006D 311 MOVAL MTH$$JACKET_HND, R1 ; R1 = adr. of handler set up by
0C BD 51 D1 0071 312 ; CALL MTH$xxx.
03 12 0071 313 CMPL R1, @SF$SAVE_FP(FP) ; test previous frame handler
50 01 D0 0075 314 BNEQ 10$ ; branch if not jacket handler
04 0077 315 MOVL #1, R0 ; return TRUE
007A 316 10$: RET ; return
007B 317
```



MTH\$\$SIGNAL  
1-003

; MATH ERROR SIGNAL ROUTINE

H 11

16-SEP-1984 01:49:10

VAX/VMS Macro V04-00

Page 11

MTH\$\$SIGNAL\_CON ; Error without signal

6-SEP-1984 11:26:55

[MTHRTL.SRC]MTHSIGNAL.MAR;1

(10)

|    |      |      |      |        |                      |                                        |
|----|------|------|------|--------|----------------------|----------------------------------------|
|    |      | 007B | 319  | .SBTTL | MTH\$\$SIGNAL_CON    | ; Error without signaling and continue |
|    |      | 007B | 320  |        |                      |                                        |
|    | 0000 | 007B | 321  | .ENTRY | MTH\$\$SIGNAL_CON, 0 |                                        |
| 50 | 01   | D0   | 007D | MOVL   | #SS\$_CONTINUE, R0   | ; error condition and continue         |
|    | 04   | 0080 | 323  | RET    |                      |                                        |
|    |      | 0081 | 324  |        |                      |                                        |
|    |      | 0081 | 325  |        |                      |                                        |
|    |      | 0081 | 326  | .END   |                      | ; end of module                        |

MTH\$\$SIGNAL  
Symbol table

; MATH ERROR SIGNAL ROUTINE

I 11

16-SEP-1984 01:49:10  
6-SEP-1984 11:26:55

VAX/VMS Macro V04-00  
[MTHRTL.SRC]MTHSIGNAL.MAR;1

Page 12  
(10)

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\$S\_SAVE\_FP = 0000000C  
SF\$S\_SAVE\_PC = 00000010  
SIGNAL1 0000003B R 02  
SS\$\_CONTINUE = 00000001  
SS\$\_RESIGNAL = 00000918  
ST\$K\_SEVERE = 00000004  
ST\$M\_FAC\_SP = 00008000  
ST\$S\_CODE = 0000000C  
ST\$V\_CODE = 00000003

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

| PSECT name | Allocation       | PSECT No. | Attributes |
|------------|------------------|-----------|------------|
| . ABS .    | 00000000 ( 0.)   | 00 ( 0.)  | NOPIC USR  |
| \$AB\$\$   | 00000000 ( 0.)   | 01 ( 1.)  | NOPIC USR  |
| _MTH\$CODE | 00000081 ( 129.) | 02 ( 2.)  | PIC USR    |

|  | CON | ABS | LCL | NOSHR | NOEXE | NORD | NOWRT | NOVEC | BYTE |
|--|-----|-----|-----|-------|-------|------|-------|-------|------|
|  | CON | ABS | LCL | NOSHR | EXE   | RD   | WRT   | NOVEC | BYTE |
|  | 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

-----  
\$255SDUA28:[MTHRTL.OBJ]MTHRTL.MLB;1  
\$255SDUA28:[SYSLIB]STARLET.MLB;2  
TOTALS (all libraries)

-----  
1  
6  
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.

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY