


```

MM      MM      TTTTTTTTTT  HH      HH  DDDDDDDD  AAAAAA  SSSSSSSS  IIIIII  NN      NN
MM      MM      TTTTTTTTTT  HH      HH  DDDDDDDD  AAAAAA  SSSSSSSS  IIIIII  NN      NN
MMMM    MMMM    TT          HH      HH  DD      DD  AA      AA  SS          II      NN      NN
MMMM    MMMM    TT          HH      HH  DD      DD  AA      AA  SS          II      NN      NN
MM      MM      TT          HH      HH  DD      DD  AA      AA  SS          II      NNNN   NN
MM      MM      TT          HH      HH  DD      DD  AA      AA  SS          II      NNNN   NN
MM      MM      TT          HHHHHHHHHH DD      DD  AA      AA  SSSSSS    II      NN      NN
MM      MM      TT          HHHHHHHHHH DD      DD  AA      AA  SSSSSS    II      NN      NN
MM      MM      TT          HH      HH  DD      DD  AAAAAAAAAA SS          II      NN      NNNN
MM      MM      TT          HH      HH  DD      DD  AAAAAAAAAA SS          II      NN      NNNN
MM      MM      TT          HH      HH  DD      DD  AA      AA  SS          II      NN      NN
MM      MM      TT          HH      HH  DD      DD  AA      AA  SS          II      NN      NN
MM      MM      TT          HH      HH  DDDDDDDD AA      AA  SSSSSSSS IIIIII  NN      NN
MM      MM      TT          HH      HH  DDDDDDDD AA      AA  SSSSSSSS IIIIII  NN      NN

```

```

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

```

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

(2)	54
(3)	73
(4)	121
(5)	181
(6)	250
(7)	310

HISTORY ; Detailed Current Edit History
DECLARATIONS ; Declarative Part of Module
MTHSDASIN - Standard Double Precision Floating DASIN
MTHSDASIN_R7 - Special DASIN routine
MTHSDASIND - Standard Double Precision Floating DASIND
MTHSDASIND_R7 - Special DASIND routine

```
0000 1 .TITLE MTHSDASIN ; Double Precision Floating Point Sine routine
0000 2 ; (DASIN,DASIND)
0000 3 .IDENT /1-006/ ; File: MTHSDASIN.MAR Edit: RNH1007
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: MATH LIBRARY
0000 30 :++
0000 31 : ABSTRACT:
0000 32 :
0000 33 : MTHSDASIN is a function which returns the double precision floating point
0000 34 : arcsine in radians of its double precision floating point argument. The call
0000 35 : is standard call-by-reference.
0000 36 :
0000 37 : MTHSDASIND is a function which returns the double precision floating point
0000 38 : arcsine in radians of its double precision floating point argument. The call
0000 39 : is standard call-by-reference.
0000 40 :
0000 41 :--
0000 42 :
0000 43 : VERSION: 01
0000 44 :
0000 45 : HISTORY:
0000 46 : AUTHOR:
0000 47 : Peter Yuo, 29-Jun-77: Version 01
0000 48 :
0000 49 : MODIFIED BY:
0000 50 :
0000 51 :
0000 52 :
```

```
0000 54 .SBTTL HISTORY ; Detailed Current Edit History
0000 55
0000 56
0000 57 : ALGORITHMIC DIFFERENCES FROM FP-11/C ROUTINE: none
0000 58 :
0000 59 : Edit History for Version 01 of MTHSDASIN
0000 60 :
0000 61 : 0-2 MTH$ERROR changed to MTH$$SIGNAL.
0000 62 : MTH$... changed to MTH_...
0000 63 : Changed error handling mechanism. Put error result in R0:R1 before
0000 64 : calling MTH$$SIGNAL in order to allow user modify error result.
0000 65 : 0-3 Fix bug involving wrong addressing mode. JMT 9-Nov-77
0000 66 : 1-001 - Update version number and copyright notice. JBS 16-NOV-78
0000 67 : 1-002 - Change MTH_INVARG to MTH$K_INVARGMAT. JBS 07-DEC-78
0000 68 : 1-003 - Add " " to the PSECT directive. JBS 26-DEC-78
0000 69 : 1-004 - Declare externals. SBL 17-May-1979
0000 70 : 1-005 - Change JSB entry to MTHSDASIN R7. RBG 28-Sept-1979
0000 71 : 1-006 - Added degree entry points. RNH 29-MAR-1981
```

```

0000 73          .SBTTL  DECLARATIONS      ; Declarative Part of Module
0000 74
0000 75  :
0000 76  : INCLUDE FILES:          MTHJACKET.MAR
0000 77  :
0000 78  :
0000 79  :
0000 80  : EXTERNAL SYMBOLS:
0000 81  :
0000 82          .DSABL  GBL
0000 83          .EXTRN  MTH$DSQRT_R5
0000 84          .EXTRN  MTH$DATAN_R7
0000 85          .EXTRN  MTH$DATAND_R7
0000 86          .EXTRN  MTH$K_INVARGMAT
0000 87          .EXTRN  MTH$$SIGNAL
0000 88
0000 89  :
0000 90  : EQUATED SYMBOLS:
0000 91
00004080 0000 92          SD_1.0 = ^F1.0          ; 1.0
00000004 0000 93          value = 4              ; value.rd.r
0000 94
0000 95  :
0000 96  : MACROS:          none
0000 97  :
0000 98  : PSECT DECLARATIONS:
0000 99
00000000 100          .PSECT  _MTH$CODE          PIC,SHR,LONG,EXE,NOWRT
0000 101                                     ; program section for math routines
0000 102  :
0000 103  : OWN STORAGE:  none
0000 104  :
0000 105  :
0000 106  : CONSTANTS:
0000 107  :
0000 108
0000 109 D_PI_OVER_2:
68C2 A221 0FDA 40C9 0000 110          .WORD  ^0040311, ^0007732, ^0121041, ^0064302
0008 111                                     ; PI/2
0008 112 D_PI:
68C2 A221 0FDA 4149 0008 113          .WORD  ^0040511, ^0007732, ^0121041, ^0064302
0010 114                                     ; PI
0010 115 D_90:
00000000 000043B4 0010 116          .LONG  ^X00043B4, ^X0          ; 90
0018 117 D_180:
00000000 00004434 0018 118          .LONG  ^X0004434, ^X0          ; 180
0020 119

```

```

0020 121          .SBTTL MTHSDASIN - Standard Double Precision Floating DASIN
0020 122
0020 123
0020 124 :++
0020 125 : FUNCTIONAL DESCRIPTION:
0020 126 :
0020 127 : DASIN - double precision floating point function
0020 128 :
0020 129 : DASIN(X) is computed as:
0020 130 :
0020 131 :     If X = 0, then DASIN(X) = 0.
0020 132 :     If X = 1, then DASIN(X) = PI/2.
0020 133 :     If X = -1, then DASIN(X) = -PI/2.
0020 134 :     If 0 < |X| < 1, then DASIN(X) = ATAN(X/SQRT(1-X**2)).
0020 135 :     If 1 < |X|, error.
0020 136 :
0020 137 : CALLING SEQUENCE:
0020 138 :
0020 139 :     DASIN.wd.v = MTHSDASIN(x.rd.r)
0020 140 :
0020 141 : INPUT PARAMETERS:
0020 142 :
00000004 0020 143 :     LONG = 4 ; define longword multiplier
00000004 0020 144 :     x = 1 * LONG ; Contents of x is the argument
0020 145 :
0020 146 : IMPLICIT INPUTS: none
0020 147 :
0020 148 : OUTPUT PARAMETERS:
0020 149 :
0020 150 :     VALUE: double precision floating arcsine of the argument
0020 151 :
0020 152 : IMPLICIT OUTPUTS: none
0020 153 :
0020 154 : COMPLETION CODES: none
0020 155 :
0020 156 : SIDE EFFECTS:
0020 157 :
0020 158 : Signals: MTHS_INVARG if |X| > 1 with reserved operand in R0/R1
0020 159 : (copied to the signal mechanism vector CHFSL_MCH_R0/R1 by LIB$SIGNAL).
0020 160 : Associated message is: "INVALID ARGUMENT". Result is reserved
0020 161 : operand -0.0 unless a user supplied (or any) error handler changes CHFSL_MCH_R0/R1
0020 162 :
0020 163 : NOTE: This procedure disables floating point underflow, enables integer
0020 164 : overflow.
0020 165 :
0020 166 : ---
0020 167 :
40FC 0020 168 :
0020 169 : .ENTRY MTHSDASIN, ^M<IV, R2, R3, R4, R5, R6, R7>
0022 170 : ; standard call-by-reference entry
0022 171 : ; disable DV (and FU), enable IV
0022 172 : MTH$FLAG_JACKET ; flag that this is a jacket procedure in
0022 :
6D 00000000'GF 9E 0022 : MOVAB G^MTH$$JACKET_HND, (FP)
0029 : ; set handler address to jacket
0029 : ; handler
0029 :

```

50	04	BC	70	0029	173				
		01	10	0029	174				
			04	0029	175				
				0029	176	MOVD	@value(AP), R0		: case of an error in routine
				002D	177	BSBB	MTHSDASIN_R7		: If an error, convert signal to user PC
				002F	178	RET			: and resignal
				0030	179				: R0/R1 = !X! = @value(AP)
									: call special DASIN routine
									: return with result in R0/R1


```

0030 181      .SBTTL  MTHSDASIN_R7 - Special DASIN routine
0030 182
0030 183      ; Special DASIN - used by the standard routine and direct JSB call.
0030 184      ;
0030 185      ; CALLING SEQUENCE:
0030 186      ; save anything needed in R0:R7
0030 187      ; MOV  R0          ; input in R0/R1
0030 188      ; JSB   MTHSDASIN_R7
0030 189      ; RSB          ; return with result in R0/R1
0030 190      ;
0030 191
0030 192 MTHSDASIN_R7::      ; special DASIN routine
0030 193 MTHSDASIN_R9::      ; Release 1 name
56 50 70 0030 194      MOV  R0, R6      ; save X in R6/R7
2E 13 0033 195      BEQL RETURN      ; return DASIN(0) = 0
0035 196
0035 197      ;
0035 198      ; 0 < !X!
0035 199      ;
0035 200
50 8000 8F AA 0035 201      BICW  #^X8000, R0      ; R0/R1 = !X!
08 50 71 003A 202      CMPD  R0, S^#SD_1.0      ; compare !X! with 1.0
17 18 003D 203      BGEQ  GEQ_TO_1.0      ; branch if !X! >= 1.0
003F 204
003F 205      ;
003F 206      ; 0 < !X! < 1.0
003F 207      ;
003F 208
50 50 50 64 003F 209      MULD2 R0, R0      ; R0/R1 = X**2
50 08 50 63 0042 210      SUBD3 R0, S^#SD_1.0, R0      ; R0/R1 = 1.0 - X**2
00000000'EF 16 0046 211      JSB   MTHSDSQRT_R5      ; R0/R1 = DSQRT(1-X**2)
50 56 50 67 004C 212      DIVD3 R0, R6, R0      ; R0/R1 = X/DSQRT(1-X**2)
00000000'EF 17 0050 213      JMP   MTHSDATAN_R7      ; R0/R1 = DATAN(X/DSQRT(1-X**2))
0056 214
0056 215
0056 216      ;
0056 217      ; 1 =< !X!
0056 218      ;
0056 219
0056 220 GEQ_TO_1.0:
OC 14 0056 221      BGTR  ERROR      ; branch to ERROR if !X! > 1.0
0058 222
0058 223
0058 224      ;
0058 225      ; !X! = 1.0
0058 226      ;
0058 227
50 A5 AF 70 0058 228      MOV  D_PI_OVER_2, R0      ; R0/R1 = PI/2
56 73 005C 229      TSTD  R6      ; test the sign of X
03 18 005E 230      BGEQ  RETURN      ; branch if X > 0
50 50 72 0060 231      MNEGD R0, R0      ; R0/R1 = -PI/2
05 0063 232 RETURN: RSB      ; return with result in R0/R1
0064 233
0064 234
0064 235      ; 1 < !X!, error
0064 236
0064 237

```

MTHSDASIN
1-006

M 14
; Double Precision Floating Point Sine r 16-SEP-1984 01:14:07 VAX/VMS Macro V04-00 Page 7
MTHSDASIN_R7 - Special DASIN routine 6-SEP-1984 11:21:39 [MTHRTL.SRC]MTHSDASIN.MAR;1 (5)

		DD	0064	238	ERROR:	PIJSHL	(SP)	:	return PC from JSB routine	
	7E	00	8F	9A	0066	239	MOVZBL	#MTH\$K_INVARGMAT, -(SP)	:	condition value
50	01	0F	79	006A	240	ASHQ	#15, #T, R0	:	R0 = result = reserved operand -0.0	
				006E	241			:	goes to signal mechanism vector	
				006E	242			:	(CHF\$MCH_R0/R1) so error handler	
00000000	'GF	02	FB	006E	243			:	can modify the result.	
				006E	244	CALLS	#2, G^MTH\$\$SIGNAL	:	signal error and use real user's PC	
			05	0075	245	RSB		:	independent of CALL vs JSB	
				0076	246			:	return - R0 restored from CHF\$MCH_R0/R1	
				0076	247			:		
				0076	248			:		

```

0076 250          .SBTTL MTHSDASIND - Standard Double Precision Floating DASIND
0076 251
0076 252
0076 253 :++
0076 254 : FUNCTIONAL DESCRIPTION:
0076 255 :
0076 256 : DASIND - double precision floating point function
0076 257 :
0076 258 : DASIND(X) is computed as:
0076 259 :
0076 260 :     If X = 0, then DASIND(X) = 0.
0076 261 :     If X = 1, then DASIND(X) = 90.
0076 262 :     If X = -1, then DASIND(X) = -90.
0076 263 :     If 0 < |X| < 1, then DASIND(X) = ATAND(X/SQRT(1-X**2)).
0076 264 :     If 1 < |X|, error.
0076 265 :
0076 266 : CALLING SEQUENCE:
0076 267 :
0076 268 :     dasind.wd.v = MTHSDASIND(x.rd.r)
0076 269 :
0076 270 : INPUT PARAMETERS:
0076 271 :
00000004 0076 272 :     LONG = 4 ; define longword multiplier
00000004 0076 273 :     x = 1 * LONG ; Contents of x is the argument
0076 274 :
0076 275 : IMPLICIT INPUTS: none
0076 276 :
0076 277 : OUTPUT PARAMETERS:
0076 278 :
0076 279 :     VALUE: double precision floating arcsine of the argument
0076 280 :
0076 281 : IMPLICIT OUTPUTS: none
0076 282 :
0076 283 : COMPLETION CODES: none
0076 284 :
0076 285 : SIDE EFFECTS:
0076 286 :
0076 287 : Signals: MTH$ INVARG if |X| > 1 with reserved operand in R0/R1
0076 288 : (col80ed to the signal mechanism vector CHF$L_MCH_R0/R1 by LIB$SIGNAL).
0076 289 : Associated message is: "INVALID ARGUMENT". Result is reserved
0076 290 : operand -0.0 unless a user supplied (or any) error handler changes CHF$L_MCH_R0/R1
0076 291 :
0076 292 : NOTE: This procedure disables floating point underflow, enables integer
0076 293 : overflow.
0076 294 :
0076 295 : ---
0076 296 :
0076 297 :
0076 298 : .ENTRY MTHSDASIND, ^M<IV, R2, R3, R4, R5, R6, R7>
0078 299 : ; standard call-by-reference entry
0078 300 : ; disable DV (and FU), enable IV
0078 301 : MTH$FLAG_JACKET ; flag that this is a jacket procedure in
0078
0078 MOVAB G^MTH$$JACKET_HND, (FP)
007F ; set handler address to jacket
007F ; handler
007F

```

```

                    007F 302
                    007F 303
50 04 BC 70 007F 304
      01 10 007F 305      MOVD @value(AP), R0
                    0083 306      BSBB MTHSDASIND_R7
                    0085 307      RET
                    0086 308
; case of an error in routine
; If an error, convert signal to user PC
; and resignal
; R0/R1 = |X| = @value(AP)
; call special DASIND routine
; Return with result in R0/R1
```

```

0086 310      .SBTTL MTHSDASIND_R7 - Special DASIND routine
0086 311
0086 312      ; Special DASIND - used by the standard routine and direct JSB call.
0086 313      ;
0086 314      ; CALLING SEQUENCE:
0086 315      ; save anything needed in R0:R7
0086 316      ; MOVD      R0      ; input in R0/R1
0086 317      ; JSB      MTHSDASIND_R7
0086 318      ; RSB      ; Return with result in R0/R1
0086 319      ;
0086 320
0086 321      MTHSDASIND_R7::      ; special DASIND routine
56 50 70 0086 322      MOVD      R0, R6      ; save X in R6/R7
2F 13 0089 323      BEQL      D_RETURN      ; Return DASIND(0) = 0
008B 324      ;
008B 325      ;
008B 326      ; 0 < |X|
008B 327      ;
008B 328
50 8000 8F AA 008B 329      BICW      #^X8000, R0      ; R0/R1 = |X|
08 50 71 0090 330      CPD      R0, S^#SD 1.0      ; compare |X| with 1.0
17 18 0093 331      BGEQ      D_GEQ_TO_T.0      ; branch if |X| >= 1.0
0095 332      ;
0095 333      ;
0095 334      ; 0 < |X| < 1.0
0095 335      ;
0095 336
50 50 50 64 0095 337      MULD2     R0, R0      ; R0/R1 = X**2
50 08 50 63 0098 338      SUBD3     R0, S^#SD 1.0, R0      ; R0/R1 = 1.0 - X**2
00000000'EF 16 009C 339      JSB      MTHSDSQRT_R5      ; R0/R1 = DSQRT(1-X**2)
50 56 50 67 00A2 340      DIVD3     R0, R6, R0      ; R0/R1 = X/DSQRT(1-X**2)
00000000'EF 17 00A6 341      JMP      MTHSDATAND_R7      ; R0/R1 = DATAND(X/DSQRT(1-X**2))
00AC 342      ;
00AC 343      ;
00AC 344      ;
00AC 345      ; 1 =< |X|
00AC 346      ;
00AC 347      ;
00AC 348      D_GEQ_TO 1.0:
B6 14 00AC 349      BGTR      ERROR      ; branch to ERROR if |X| > 1.0
00AE 350      ;
00AE 351      ;
00AE 352      ;
00AE 353      ; |X| = 1.0
00AE 354      ;
00AE 355      ;
50 FF5E CF 70 00AE 356      MOVD      D_90, R0      ; R0/R1 = 90
56 73 00B3 357      TSTD      R6      ; test the sign of X
03 18 00B5 358      BGEQ      D_RETURN      ; branch if X > 0
50 50 72 00B7 359      MNEGD     R0, R0      ; R0/R1 = -90
00BA 360      D_RETURN:
05 00BA 361      RSB      ; Return with result in R0/R1
00BB 362      ;
00BB 363      ;
00BB 364      .END

```

MTHSDASIN
Symbol table

```

D_180      00000018 R    01
D_90       00000010 R    01
D_GEQ_TO_1.0 000000AC R    01
D-PI       00000008 R    01
D-PI_OVER_2 00000000 R    01
D-RETURN   000000BA R    01
ERROR      00000064 R    01
GEQ_TO_1.0 00000056 R    01
LONG       = 00000004
MTH$$JACKET_HND ***** X    01
MTH$$SIGNAL ***** X    00
MTHSDASIN  00000020 RG    01
MTHSDASIND 00000076 RG    01
MTHSDASIND R7 00000086 RG    01
MTHSDASIN R7 00000030 RG    01
MTHSDASIN R9 00000030 RG    01
MTHSDATAN R7 ***** X    00
MTHSDATAN R7 ***** X    00
MTHSDSQRT R5 ***** X    00
MTHSK_INVARGMAT ***** X    00
RETURN     00000063 R    01
SD 1.0     = 00004080
VACUE      = 00000004
  
```

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

PSECT name	Allocation	PSECT No.	Attributes										
ABS	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
_MTH\$CODE	000000BB (187.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	34	00:00:00.09	00:00:01.08
Command processing	116	00:00:00.67	00:00:05.66
Pass 1	96	00:00:00.99	00:00:03.39
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	75	00:00:00.88	00:00:03.05
Symbol table output	3	00:00:00.03	00:00:00.03
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	328	00:00:02.69	00:00:13.24

The working set limit was 750 pages.
5305 bytes (11 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 24 non-local and 0 local symbols.
424 source lines were read in Pass 1, producing 14 object records in Pass 2.
1 page of virtual memory was used to define 1 macro.

! Macro library statistics !

<u>Macro library name</u>	<u>Macros defined</u>
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:MTHDASIN/OBJ=OBJ\$:MTHDASIN MSRC\$:MTHJACKET/UPDATE=(ENH\$:MTHJACKET)+MSRC

