

[illegible]

3

Sy

MT

MT
MT

MT

MT
MT

MT
MT

MT
MTMT
MTMT
MT

MT

MT

MT
MT

MT

MT
MT

MT
MTMT
MTMT
MT

MT

MT

MT
MT

MI

MT
MTMT
MTMT
MT

MT

M1
M1

M1

M1
M1M1
M1

100

1

1

100

10

1

(2)	51	HISTORY	; Detailed Current Edit History
(3)	64	DECLARATIONS	
(4)	95	MTH\$DFLOTI	
(5)	132	MTH\$DFLOTJ	
(6)	171	MTH\$IINT, MTH\$IIFIX	
(7)	211	MTH\$JINT, MTH\$JIFIX	
(8)	253	MTH\$IIDINT	
(9)	293	MTH\$JIDINT	
(10)	333	MTH\$REAL	
(11)	372	MTH\$AIMAG	
(12)	412	MTH\$CMPLX	
(13)	455	MTH\$DBLE	
(14)	495	MTH\$SNGI	
(15)	535	MTH\$SNGLG	
(16)	576	MTH\$FLOATI	
(17)	614	MTH\$FLOATJ	
(18)	654	MTH\$GFLOTI	
(19)	691	MTH\$GFLOTJ	
(20)	730	MTH\$IIGINT	
(21)	770	MTH\$JIGINT	
(22)	810	MTH\$IIHINT	
(23)	850	MTH\$JIHINT	
(24)	890	MTH\$DREAL	
(25)	929	MTH\$GREAL	
(26)	968	MTH\$DIMAG	
(27)	1008	MTH\$GIMAG	
(28)	1048	MTH\$DCMPLX	
(29)	1099	MTH\$GCMPLX	
(30)	1153	MTH\$GDBLE	

```
0000 1      .TITLE MTH$CONVER      type conversions (FLOAT, FIX, INT)
0000 2      .IDENT /1-004/          ; File: MTHCONVER.MAR Edit: JAW1004
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: MATH LIBRARY
0000 30 ++
0000 31 ABSTRACT:
0000 32      This module contains all of the one or two instruction type
0000 33      conversion routines.
0000 34
0000 35
0000 36 --
0000 37
0000 38 VERSION: 1
0000 39
0000 40 HISTORY:
0000 41
0000 42 AUTHOR:
0000 43      Jonathan M. Taylor, 14-JUL-77: Version 0
0000 44
0000 45 MODIFIED BY:
0000 46
0000 47      Steven B. Lionel, 26-Jan-79: G, H and COMPLEX*16
0000 48
0000 49
```


MTH\$CONVER
1-004

type conversions (FLOAT, FIX, INT) 1 7
HISTORY ; Detailed Current Edit History 16-SEP-1984 01:11:23 VAX/VMS Macro V04-00 Page 2
6-SEP-1984 11:21:17 [MTHRTL.SRC]MTHCONVER.MAR;1 (2)

```
0000 51      .SBTTL HISTORY      ; Detailed Current Edit History
0000 52
0000 53
0000 54 : Edit History for Version 1 of MTH$CONVER
0000 55 :
0000 56 : 0-3 - Remove MTH$FLAG_JACKET. TNH 26-July-78
0000 57 : 1-001 - Update version number and copyright notice. JBS 16-NOV-78
0000 58 : 1-002 - Add G, H, COMPLEX*16. SBL 26-Jan-79
0000 59 : 1-003 - Add MTH$IHINT, MTH$JHINT. SBL 12-Sept-1979
0000 60 : 1-004 - Make MTH$DCMPLX and MTH$GCMPLX fetch and store arguments
0000 61 : properly, and make MTH$DIMAG and MTH$GIMAG use correct offset
0000 62 : for imaginary part. JAW 27-Jan-1981
```

```
0000 64      .SBTTL DECLARATIONS
0000 65
0000 66 :
0000 67 : INCLUDE FILES:
0000 68 :     NONE
0000 69 :
0000 70 :
0000 71 :
0000 72 : EXTERNAL SYMBOLS:
0000 73 :     NONE
0000 74 :
0000 75 :
0000 76 :
0000 77 : MACROS:
0000 78 :     NONE
0000 79 :
0000 80 :
0000 81 :
0000 82 : PSECT DECLARATIONS:
0000 83 :     .PSECT _MTH$CODE      PIC, SHR, LONG, EXE, NOWRT
0000 84 :
0000 85 :
0000 86 : EQUATED SYMBOLS:
0000 87 :     NONE
0000 88 :
0000 89 :
0000 90 :
0000 91 : OWN STORAGE:
0000 92 :     NONE
0000 93 :
```



```
0000 95      .SBTTL MTH$DFLOTI
0000 96
0000 97      :++
0000 98      : FUNCTIONAL DESCRIPTION:
0000 99      :   Convert INTEGER*2 to REAL*8 (exact).
0000 100
0000 101
0000 102      : CALLING SEQUENCE:
0000 103      :   Float.wd.v = MTH$DFLOTI (arg.rw.r)
0000 104
0000 105
0000 106
0000 107      : INPUT PARAMETERS:
0000 108      :   The input parameter is a one-word value and is call-by-reference.
0000 109
0000 110
0000 111      : IMPLICIT INPUTS:
0000 112      :   NONE
0000 113
0000 114      : OUTPUT PARAMETERS:
0000 115      :   NONE
0000 116
0000 117      : IMPLICIT OUTPUTS:
0000 118      :   NONE
0000 119
0000 120      : COMPLETION CODES:
0000 121      :   NONE
0000 122
0000 123      : SIDE EFFECTS:
0000 124      :   NONE
0000 125
0000 126      :--
0000 127
50  04 BC 0000 0000 128      .ENTRY MTH$DFLOTI, ^M<>
        6D 0002 129      CVTWD @4(AP), R0 ; convert INTEGER*2 to REAL*8
        04 0006 130      RET
```

```
0007 132      .SBTTL MTH$DFLOTJ
0007 133
0007 134      :++
0007 135      : FUNCTIONAL DESCRIPTION:
0007 136      :   Convert INTEGER*4 to REAL*8 (exact).
0007 137
0007 138
0007 139      : CALLING SEQUENCE:
0007 140      :   Float.wd.v = MTH$DFLOTJ (arg.rl.r)
0007 141
0007 142
0007 143
0007 144      : INPUT PARAMETERS:
0007 145      :   The one input parameter is a longword value and is
0007 146      :   call-by-reference.
0007 147
0007 148
0007 149      : IMPLICIT INPUTS:
0007 150      :   NONE
0007 151
0007 152      : OUTPUT PARAMETERS:
0007 153      :   NONE
0007 154
0007 155      : IMPLICIT OUTPUTS:
0007 156      :   NONE
0007 157
0007 158      : COMPLETION CODES:
0007 159      :   NONE
0007 160
0007 161      : SIDE EFFECTS:
0007 162      :   NONE
0007 163
0007 164      :--
0007 165
0007 166
0007 167      .ENTRY MTH$DFLOTJ,      ^M<>
0009 168      CVTLD @4(AP), R0      ; convert INTEGER*4 to REAL*8
000D 169      RET
```

50 04 BC 0000 0007 167
6E 0009 168
04 000D 169


```
000E 171 .SBTTL MTH$IINT, MTH$IIFIX
000E 172
000E 173 :++
000E 174 : FUNCTIONAL DESCRIPTION:
000E 175 : Convert REAL*4 to INTEGER*2 (truncated).
000E 176 :
000E 177 :
000E 178 : CALLING SEQUENCE:
000E 179 : Truncation.wv.v = MTH$IINT (arg.rf.r)
000E 180 : Integer.wv.v = MTH$IIFIX (arg.rf.r)
000E 181 :
000E 182 :
000E 183 :
000E 184 : INPUT PARAMETERS:
000E 185 :
000E 186 :
000E 187 : IMPLICIT INPUTS:
000E 188 : NONE
000E 189 :
000E 190 : OUTPUT PARAMETERS:
000E 191 : NONE
000E 192 :
000E 193 : IMPLICIT OUTPUTS:
000E 194 : NONE
000E 195 :
000E 196 : COMPLETION CODES:
000E 197 : NONE
000E 198 :
000E 199 : SIDE EFFECTS:
000E 200 : Integer Overflow and Reserved Operand exceptions can occur.
000E 201 :
000E 202 :
000E 203 :--
000E 204 :
000E 205 :
000E 206 MTH$IINT::
000E 207 .ENTRY MTH$IIFIX, ^M<IV> ; enable integer overflow
50 04 BC 4000 0010 208 CMTFW @4(AP), R0 ; convert REAL*4 to INTEGER*2
04 0014 209 RET
```

```
0015 211 .SBTTL MTH$JINT, MTH$JIFIX
0015 212
0015 213 :++
0015 214 : FUNCTIONAL DESCRIPTION:
0015 215 : Convert REAL*4 to INTEGER*4 (truncated).
0015 216 :
0015 217 :
0015 218 : CALLING SEQUENCE:
0015 219 : Truncation.wl.v = MTH$JINT (arg.rf.r)
0015 220 : Integer.wl.v = MTH$JIFIX (arg.rf.r)
0015 221 :
0015 222 :
0015 223 :
0015 224 : INPUT PARAMETERS:
0015 225 : The one input parameter is a single-precision floating-point
0015 226 : value and is call-by-reference.
0015 227 :
0015 228 :
0015 229 : IMPLICIT INPUTS:
0015 230 : NONE
0015 231 :
0015 232 : OUTPUT PARAMETERS:
0015 233 : NONE
0015 234 :
0015 235 : IMPLICIT OUTPUTS:
0015 236 : NONE
0015 237 :
0015 238 : COMPLETION CODES:
0015 239 : NONE
0015 240 :
0015 241 : SIDE EFFECTS:
0015 242 : Reserved Operand and Integer Overflow exceptions can occur.
0015 243 :
0015 244 :
0015 245 :--
0015 246 :
0015 247 :
0015 248 MTH$JINT::
0015 249 .ENTRY MTH$JIFIX, ^M<IV> ; enable integer overflow
0017 250 CVTFL @4(AP), R0 ; convert REAL*4 to INTEGER*4
001B 251 RET
```

50 04 BC 4000 4A 0017 250
04 001B 251


```
001C 253 .SBTTL MTH$IIDINT
001C 254
001C 255 :++
001C 256 : FUNCTIONAL DESCRIPTION:
001C 257 : Convert REAL*8 to INTEGER*2 (truncated).
001C 258 :
001C 259 :
001C 260 : CALLING SEQUENCE:
001C 261 : Truncation.wv.v = MTH$IIDINT (arg.rd.r)
001C 262 :
001C 263 :
001C 264 :
001C 265 : INPUT PARAMETERS:
001C 266 : The one input parameter is a double-precision floating-point
001C 267 : value and is call-by-reference.
001C 268 :
001C 269 :
001C 270 : IMPLICIT INPUTS:
001C 271 : NONE
001C 272 :
001C 273 : OUTPUT PARAMETERS:
001C 274 : NONE
001C 275 :
001C 276 : IMPLICIT OUTPUTS:
001C 277 : NONE
001C 278 :
001C 279 : COMPLETION CODES:
001C 280 : NONE
001C 281 :
001C 282 : SIDE EFFECTS:
001C 283 : Reserved Operand and Integer Overflow exceptions can occur.
001C 284 :
001C 285 :
001C 286 :--
001C 287
001C 288
50 04 BC 4000 001C 289 .ENTRY MTH$IIDINT, ^M<IV> ; enable integer overflow
69 001E 290 CVTDW @4(AP), R0 ; REAL*8 to INTEGER*2
04 0022 291 RET
```

```
0023 293 .SBTTL MTH$JIDINT
0023 294
0023 295 :++
0023 296 : FUNCTIONAL DESCRIPTION:
0023 297 : Convert REAL*8 to INTEGER*4 (truncated).
0023 298 :
0023 299 :
0023 300 : CALLING SEQUENCE:
0023 301 : Truncation.wl.v = MTH$JIDINT (arg.rd.r)
0023 302 :
0023 303 :
0023 304 :
0023 305 : INPUT PARAMETERS:
0023 306 : The one input parameter is double-precision floating-point value
0023 307 : and is call-by-reference.
0023 308 :
0023 309 :
0023 310 : IMPLICIT INPUTS:
0023 311 : NONE
0023 312 :
0023 313 : OUTPUT PARAMETERS:
0023 314 : NONE
0023 315 :
0023 316 : IMPLICIT OUTPUTS:
0023 317 : NONE
0023 318 :
0023 319 : COMPLETION CODES:
0023 320 : NONE
0023 321 :
0023 322 : SIDE EFFECTS:
0023 323 : Reserved Operand and Integer Overflow exceptions can occur.
0023 324 :
0023 325 :
0023 326 :--
0023 327
0023 328
0023 329 .ENTRY MTH$JIDINT, ^M<IV> : enable integer overflow
0025 330 CVTDL @4(AP), R0 ; REAL*8 to INTEGER*4
0029 331 RET
```

50 04 BC 4000
6A
04


```
002A 333      .SBTTL MTH$REAL
002A 334
002A 335      :++
002A 336      : FUNCTIONAL DESCRIPTION:
002A 337      :   Return the REAL part of the COMPLEX argument.
002A 338      :
002A 339      :
002A 340      : CALLING SEQUENCE:
002A 341      :   Real_part.wf.v = MTH$REAL (complex_arg.rfc.r)
002A 342      :
002A 343      :
002A 344      : INPUT PARAMETERS:
002A 345      :   The one argument is a COMPLEX*8 number and is call-by-reference.
002A 346      :
002A 347      :
002A 348      : IMPLICIT INPUTS:
002A 349      :   NONE
002A 350      :
002A 351      : OUTPUT PARAMETERS:
002A 352      :   NONE
002A 353      :
002A 354      : IMPLICIT OUTPUTS:
002A 355      :   NONE
002A 356      :
002A 357      : COMPLETION CODES:
002A 358      :   NONE
002A 359      :
002A 360      : SIDE EFFECTS:
002A 361      :   Reserved Operand exception can occur.
002A 362      :
002A 363      :
002A 364      :
002A 365      :--
002A 366
002A 367
50  04 BC 0000 002A 368      .ENTRY MTH$REAL,      ^M<>
                                MOVF  @4(AP), R0      ; get 1st half of COMPLEX number
002C 369      RET
0030 370
```

```
0031 372      .SBTTL MTH$AIMAG
0031 373
0031 374      :++
0031 375      : FUNCTIONAL DESCRIPTION:
0031 376      : Returns the IMAGINARY part of a COMPLEX*8 argument.
0031 377
0031 378
0031 379      : CALLING SEQUENCE:
0031 380      : Imaginary_part.wf.v = MTH$AIMAG (complex_arg.rfc.r)
0031 381
0031 382
0031 383
0031 384      : INPUT PARAMETERS:
0031 385      : The one argument is a COMPLEX*8 value and is call-by-reference.
0031 386
0031 387
0031 388      : IMPLICIT INPUTS:
0031 389      : NONE
0031 390
0031 391      : OUTPUT PARAMETERS:
0031 392      : NONE
0031 393
0031 394      : IMPLICIT OUTPUTS:
0031 395      : NONE
0031 396
0031 397      : COMPLETION CODES:
0031 398      : NONE
0031 399
0031 400      : SIDE EFFECTS:
0031 401      : Reserved Operand exception can occur.
0031 402
0031 403
0031 404      :--
0031 405
0031 406
0031 407      .ENTRY MTH$AIMAG,      ^M<>
50  04 AC 0000 0033 408      MOVL 4(AP), R0      ; R0 -> COMPLEX number
50  04 A0 50 0037 409      MOVF 4(R0), R0      ; R0 = imaginary part of COMPLEX number
003B 410      RET
```



```
003C 412 .SBTTL MTH$CMPLX
003C 413
003C 414 :++
003C 415 : FUNCTIONAL DESCRIPTION:
003C 416 : Return a COMPLEX*8 number from two single-precision
003C 417 : floating-point values.
003C 418
003C 419
003C 420 : CALLING SEQUENCE:
003C 421 : Complex_from_two_reals.wfc.v = MTH$CMPLX
003C 422 : (real_part.rf.r, imaginary_part.rf.r)
003C 423
003C 424
003C 425
003C 426 : INPUT PARAMETERS:
003C 427 : The two input parameters are single-precision floating-point
003C 428 : values and are call-by-reference.
003C 429
003C 430
003C 431 : IMPLICIT INPUTS:
003C 432 : NONE
003C 433
003C 434 : OUTPUT PARAMETERS:
003C 435 : NONE
003C 436
003C 437 : IMPLICIT OUTPUTS:
003C 438 : NONE
003C 439
003C 440 : COMPLETION CODES:
003C 441 : NONE
003C 442
003C 443 : SIDE EFFECTS:
003C 444 : Reserved Operand exception can occur.
003C 445
003C 446
003C 447 :--
003C 448
003C 449
003C 450 .ENTRY MTH$CMPLX, ^M<>
50 04 BC 0000 003E 451 MOVF @4(AP), R0 ; real part = 1st arg
51 08 BC 50 0042 452 MOVF @8(AP), R1 ; imag part = 2nd arg
04 0046 453 RET
```

```
0047 455      .SBTTL MTH$DBLE
0047 456
0047 457      :++
0047 458      : FUNCTIONAL DESCRIPTION:
0047 459      :   Convert REAL*4 to REAL*8 (exact).
0047 460      :
0047 461      :
0047 462      : CALLING SEQUENCE:
0047 463      :   Conversion.wd.v = MTH$DBLE (arg.rf.r)
0047 464      :
0047 465      :
0047 466      : INPUT PARAMETERS:
0047 467      :   The one parameter is a single-precision floating-point value
0047 468      :   and is call-by-reference.
0047 469      :
0047 470      :
0047 471      : IMPLICIT INPUTS:
0047 472      :   NONE
0047 473      :
0047 474      : OUTPUT PARAMETERS:
0047 475      :   NONE
0047 476      :
0047 477      : IMPLICIT OUTPUTS:
0047 478      :   NONE
0047 479      :
0047 480      : COMPLETION CODES:
0047 481      :   NONE
0047 482      :
0047 483      : SIDE EFFECTS:
0047 484      :   Reserved Operand exception can occur.
0047 485      :
0047 486      :
0047 487      :
0047 488      :--
0047 489
0047 490
0047 491      .ENTRY MTH$DBLE,      ^M<>
0049 492      CVTFD  @4(AP), R0      ; REAL*4 to REAL*8 in R0/R1
004D 493      RET
```

50 04 BC 0000 0047 491
56 0049 492
04 004D 493


```
004E 495      .SBTTL MTH$SNGL
004E 496
004E 497      :++
004E 498      : FUNCTIONAL DESCRIPTION:
004E 499      :   Convert REAL*8 to REAL*4 (rounded)
004E 500      :
004E 501      :
004E 502      : CALLING SEQUENCE:
004E 503      :   Conversion.wf.v = MTH$SNGL (arg.rd.r)
004E 504      :
004E 505      :
004E 506      : INPUT PARAMETERS:
004E 507      :   The one parameter is a double-precision floating-point value
004E 508      :   and is call-by-reference.
004E 509      :
004E 510      :
004E 511      : IMPLICIT INPUTS:
004E 512      :   NONE
004E 513      :
004E 514      : OUTPUT PARAMETERS:
004E 515      :   NONE
004E 516      :
004E 517      : IMPLICIT OUTPUTS:
004E 518      :   NONE
004E 519      :
004E 520      : COMPLETION CODES:
004E 521      :   NONE
004E 522      :
004E 523      : SIDE EFFECTS:
004E 524      :   Reserved Operand and Floating Overflow exceptions can occur.
004E 525      :
004E 526      :
004E 527      :
004E 528      :--
004E 529
004E 530
004E 531      .ENTRY MTH$SNGL,      ^M<>
50  04 BC 0000 004E 532      CVTDF @4(AP), R0      ; REAL*8 to REAL*4 in R0
04 0050 533      RET
```

```

0055 535      .SBTTL MTH$SNGLG
0055 536
0055 537      :++
0055 538      : FUNCTIONAL DESCRIPTION:
0055 539      :   Convert G REAL*8 to REAL*4 (rounded)
0055 540      :
0055 541      :
0055 542      : CALLING SEQUENCE:
0055 543      :   Conversion.wf.v = MTH$SNGLG (arg.rg.r)
0055 544      :
0055 545      :
0055 546      : INPUT PARAMETERS:
0055 547      :   The one parameter is a G floating-point value
0055 548      :   and is call-by-reference.
0055 549      :
0055 550      :
0055 551      : IMPLICIT INPUTS:
0055 552      :   NONE
0055 553      :
0055 554      : OUTPUT PARAMETERS:
0055 555      :   NONE
0055 556      :
0055 557      : IMPLICIT OUTPUTS:
0055 558      :   NONE
0055 559      :
0055 560      : COMPLETION CODES:
0055 561      :   NONE
0055 562      :
0055 563      : SIDE EFFECTS:
0055 564      :   Reserved Operand and Floating Overflow exceptions can occur.
0055 565      :   Floating Underflow can occur.
0055 566      :
0055 567      :
0055 568      :
0055 569      :--
0055 570
0055 571
0055 572      .ENTRY MTH$SNGLG,      ^M<>
50  04 BC 0000 0057 573      CVTGF @4(AP), R0      ; G REAL*8 to REAL*4 in R0
      04 005C 574      RET

```



```
005D 576      .SBTTL MTH$FLOATI
005D 577
005D 578      :++
005D 579      FUNCTIONAL DESCRIPTION:
005D 580      Convert INTEGER*2 to REAL*4 (exact).
005D 581
005D 582
005D 583      CALLING SEQUENCE:
005D 584      Float.wf.v = MTH$FLOATI (arg.rw.r)
005D 585
005D 586
005D 587
005D 588      INPUT PARAMETERS:
005D 589      The one parameter is a one-word value and is call-by-reference.
005D 590
005D 591
005D 592      IMPLICIT INPUTS:
005D 593      NONE
005D 594
005D 595      OUTPUT PARAMETERS:
005D 596      NONE
005D 597
005D 598      IMPLICIT OUTPUTS:
005D 599      NONE
005D 600
005D 601      COMPLETION CODES:
005D 602      NONE
005D 603
005D 604      SIDE EFFECTS:
005D 605      NONE
005D 606
005D 607      :--
005D 608
005D 609
50  04 BC 0000 005D 610      .ENTRY MTH$FLOATI,      ^M<>
                                CVTWF @4(AP), R0      ; INTEGER*2 to REAL*4
                                04 0063 611      RET
                                0063 612
```

```

0064 614      .SBTTL MTH$FLOATJ
0064 615
0064 616 :++
0064 617 : FUNCTIONAL DESCRIPTION:
0064 618 :   Convert INTEGER*4 to REAL*4 (rounded).
0064 619 :
0064 620 :
0064 621 : CALLING SEQUENCE:
0064 622 :   Float.wf.v = MTH$FLOATJ (arg.rl.r)
0064 623 :
0064 624 :
0064 625 :
0064 626 : INPUT PARAMETERS:
0064 627 :   The one parameter is a longword value and is call-by-reference.
0064 628 :
0064 629 :
0064 630 : IMPLICIT INPUTS:
0064 631 :   NONE
0064 632 :
0064 633 : OUTPUT PARAMETERS:
0064 634 :   NONE
0064 635 :
0064 636 : IMPLICIT OUTPUTS:
0064 637 :   NONE
0064 638 :
0064 639 : COMPLETION CODES:
0064 640 :   NONE
0064 641 :
0064 642 : SIDE EFFECTS:
0064 643 :   NONE
0064 644 :
0064 645 :--
0064 646
0064 647
0064 648      .ENTRY MTH$FLOATJ,      ^M<>
0066 649      CRTL  @4(AP), R0      ; INTEGER*4 to REAL*4 in R0
006A 650      RET
006B 651
006B 652

```

50 04 BC 0000
4E
04


```

006B 654 .SBTTL MTH$GFLOTI
006B 655
006B 656 :++
006B 657 : FUNCTIONAL DESCRIPTION:
006B 658 : Convert INTEGER*2 to G REAL*8 (exact).
006B 659 :
006B 660 :
006B 661 : CALLING SEQUENCE:
006B 662 : Float.wg.v = MTH$GFLOTI (arg.rw.r)
006B 663 :
006B 664 :
006B 665 :
006B 666 : INPUT PARAMETERS:
006B 667 : The input parameter is a one-word value and is call-by-reference.
006B 668 :
006B 669 :
006B 670 : IMPLICIT INPUTS:
006B 671 : NONE
006B 672 :
006B 673 : OUTPUT PARAMETERS:
006B 674 : NONE
006B 675 :
006B 676 : IMPLICIT OUTPUTS:
006B 677 : NONE
006B 678 :
006B 679 : COMPLETION CODES:
006B 680 : NONE
006B 681 :
006B 682 : SIDE EFFECTS:
006B 683 : NONE
006B 684 :
006B 685 :--
006B 686
006B 687 .ENTRY MTH$GFLOTI, ^M<>
006D 688 CVTWG @4(AP), R0 ; convert INTEGER*2 to G REAL*8
0072 689 RET

```

```
0073 691 .SBTTL MTH$GFLOTJ
0073 692
0073 693 :++
0073 694 : FUNCTIONAL DESCRIPTION:
0073 695 : Convert INTEGER*4 to G REAL*8 (exact).
0073 696 :
0073 697 :
0073 698 : CALLING SEQUENCE:
0073 699 : float.wg.v = MTH$GFLOTJ (arg.rl.r)
0073 700 :
0073 701 :
0073 702 :
0073 703 : INPUT PARAMETERS:
0073 704 : The one input parameter is a longword value and is
0073 705 : call-by-reference.
0073 706 :
0073 707 :
0073 708 : IMPLICIT INPUTS:
0073 709 : NONE
0073 710 :
0073 711 : OUTPUT PARAMETERS:
0073 712 : NONE
0073 713 :
0073 714 : IMPLICIT OUTPUTS:
0073 715 : NONE
0073 716 :
0073 717 : COMPLETION CODES:
0073 718 : NONE
0073 719 :
0073 720 : SIDE EFFECTS:
0073 721 : NONE
0073 722 :
0073 723 :--
0073 724
0073 725
50 04 BC 0000 0073 726 .ENTRY MTH$GFLOTJ, ^M<>
04 4EFD 0075 727 CRTLQ @4(AP), R0 ; convert INTEGER*4 to G REAL*8
04 007A 007A 728 RET
```



```
007B 730      .SBTTL MTH$IIGINT
007B 731
007B 732 :++
007B 733 : FUNCTIONAL DESCRIPTION:
007B 734 :   Convert G REAL*8 to INTEGER*2 (truncated).
007B 735 :
007B 736 :
007B 737 : CALLING SEQUENCE:
007B 738 :   Truncation.wv.v = MTH$IIGINT (arg.rg.r)
007B 739 :
007B 740 :
007B 741 :
007B 742 : INPUT PARAMETERS:
007B 743 :   The one input parameter is a G floating-point
007B 744 :   value and is call-by-reference.
007B 745 :
007B 746 :
007B 747 : IMPLICIT INPUTS:
007B 748 :   NONE
007B 749 :
007B 750 : OUTPUT PARAMETERS:
007B 751 :   NONE
007B 752 :
007B 753 : IMPLICIT OUTPUTS:
007B 754 :   NONE
007B 755 :
007B 756 : COMPLETION CODES:
007B 757 :   NONE
007B 758 :
007B 759 : SIDE EFFECTS:
007B 760 :   Reserved Operand and Integer Overflow exceptions can occur.
007B 761 :
007B 762 :
007B 763 :--
007B 764
007B 765
007B 766      .ENTRY MTH$IIGINT,      ^M<IV> ; enable integer overflow
007D 767      CVTGW @4(AP), R0      ; G REAL*8 to INTEGER*2
0082 768      RET
```

50 04 BC 4000 007B 766
04 49FD 007D 767
04 0082 0082 768

```
0083 770      .SBTTL MTH$JIGINT
0083 771
0083 772 :++
0083 773 : FUNCTIONAL DESCRIPTION:
0083 774 :   Convert G REAL*8 to INTEGER*4 (truncated).
0083 775 :
0083 776 :
0083 777 : CALLING SEQUENCE:
0083 778 :   Truncation.wl.v = MTH$JIGINT (arg.rg.r)
0083 779 :
0083 780 :
0083 781 :
0083 782 : INPUT PARAMETERS:
0083 783 :   The one input parameter is G floating-point value
0083 784 :   and is call-by-reference.
0083 785 :
0083 786 :
0083 787 : IMPLICIT INPUTS:
0083 788 :   NONE
0083 789 :
0083 790 : OUTPUT PARAMETERS:
0083 791 :   NONE
0083 792 :
0083 793 : IMPLICIT OUTPUTS:
0083 794 :   NONE
0083 795 :
0083 796 : COMPLETION CODES:
0083 797 :   NONE
0083 798 :
0083 799 : SIDE EFFECTS:
0083 800 :   Reserved Operand and Integer Overflow exceptions can occur.
0083 801 :
0083 802 :
0083 803 :--
0083 804
0083 805
0083 806      .ENTRY MTH$JIGINT,      ^M<IV> ; enable integer overflow
0085 807      CVTGL @4(AP), R0      ; G REAL*8 to INTEGER*4
008A 808      RET
```

50 04 BC 4000
4AFD
04


```
008B 810 .SBTTL MTH$IIHINT
008B 811
008B 812 :++
008B 813 : FUNCTIONAL DESCRIPTION:
008B 814 : Convert REAL*16 to INTEGER*2 (truncated).
008B 815 :
008B 816 :
008B 817 : CALLING SEQUENCE:
008B 818 : Truncation.wv.v = MTH$IIHINT (arg.rh.r)
008B 819 :
008B 820 :
008B 821 :
008B 822 : INPUT PARAMETERS:
008B 823 : The one input parameter is a H-floating
008B 824 : value and is call-by-reference.
008B 825 :
008B 826 :
008B 827 : IMPLICIT INPUTS:
008B 828 : NONE
008B 829 :
008B 830 : OUTPUT PARAMETERS:
008B 831 : NONE
008B 832 :
008B 833 : IMPLICIT OUTPUTS:
008B 834 : NONE
008B 835 :
008B 836 : COMPLETION CODES:
008B 837 : NONE
008B 838 :
008B 839 : SIDE EFFECTS:
008B 840 : Reserved Operand and Integer Overflow exceptions can occur.
008B 841 :
008B 842 :
008B 843 :--
008B 844 :
008B 845 :
50 04 BC 4000 008B 846 .ENTRY MTH$IIHINT, ^M<IV> : enable integer overflow
04 69FD 008D 847 CVTHW @4(AP), R0 ; REAL*16 to INTEGER*2
0092 0092 848 RET
```

```
0093 850      .SBTTL MTH$JIHINT
0093 851
0093 852      :++
0093 853      : FUNCTIONAL DESCRIPTION:
0093 854      :   Convert REAL*16 to INTEGER*4 (truncated).
0093 855      :
0093 856      :
0093 857      : CALLING SEQUENCE:
0093 858      :   Truncation.wl.v = MTH$JIHINT (arg.rh.r)
0093 859      :
0093 860      :
0093 861      : INPUT PARAMETERS:
0093 862      :   The one input parameter is H_floating
0093 863      :   and is call-by-reference.
0093 864      :
0093 865      :
0093 866      : IMPLICIT INPUTS:
0093 867      :   NONE
0093 868      :
0093 869      :
0093 870      : OUTPUT PARAMETERS:
0093 871      :   NONE
0093 872      :
0093 873      : IMPLICIT OUTPUTS:
0093 874      :   NONE
0093 875      :
0093 876      : COMPLETION CODES:
0093 877      :   NONE
0093 878      :
0093 879      : SIDE EFFECTS:
0093 880      :   Reserved Operand and Integer Overflow exceptions can occur.
0093 881      :
0093 882      :
0093 883      :--
0093 884
0093 885
0093 886      .ENTRY MTH$JIHINT,      ^M<IV> ; enable integer overflow
0095 887      CVTHL @4(AP), R0      ; REAL*16 to INTEGER*4
009A 888      RET
```

50 04 BC 4000 6AFD 04


```
009B 890      .SBTTL MTH$DREAL
009B 891
009B 892      ++
009B 893      FUNCTIONAL DESCRIPTION:
009B 894      Return the REAL*8 part of the COMPLEX*16 argument.
009B 895
009B 896
009B 897      CALLING SEQUENCE:
009B 898      Real_part.wd.v = MTH$DREAL (complex_arg.rdc.r)
009B 899
009B 900
009B 901
009B 902      INPUT PARAMETERS:
009B 903      The one argument is a COMPLEX*16 number and is call-by-reference.
009B 904
009B 905
009B 906      IMPLICIT INPUTS:
009B 907      NONE
009B 908
009B 909      OUTPUT PARAMETERS:
009B 910      NONE
009B 911
009B 912      IMPLICIT OUTPUTS:
009B 913      NONE
009B 914
009B 915      COMPLETION CODES:
009B 916      NONE
009B 917
009B 918      SIDE EFFECTS:
009B 919      Reserved Operand exception can occur.
009B 920
009B 921
009B 922      --
009B 923
009B 924
009B 925      .ENTRY MTH$DREAL,      ^M<>
009D 926      MOVD  @4(AP), R0      ; get 1st half of COMPLEX*16 number
00A1 927      RET
```

50 04 BC 0000
70
04

```
00A2 929      .SBTTL MTH$GREAL
00A2 930
00A2 931      :++
00A2 932      : FUNCTIONAL DESCRIPTION:
00A2 933      :   Return the G REAL*8 part of the G COMPLEX*16 argument.
00A2 934      :
00A2 935      :
00A2 936      : CALLING SEQUENCE:
00A2 937      :   Real_part.wg.v = MTH$GREAL (complex_arg.rgc.r)
00A2 938      :
00A2 939      :
00A2 940      :
00A2 941      : INPUT PARAMETERS:
00A2 942      :   The one argument is a G COMPLEX*16 number and is call-by-reference.
00A2 943      :
00A2 944      :
00A2 945      : IMPLICIT INPUTS:
00A2 946      :   NONE
00A2 947      :
00A2 948      : OUTPUT PARAMETERS:
00A2 949      :   NONE
00A2 950      :
00A2 951      : IMPLICIT OUTPUTS:
00A2 952      :   NONE
00A2 953      :
00A2 954      : COMPLETION CODES:
00A2 955      :   NONE
00A2 956      :
00A2 957      : SIDE EFFECTS:
00A2 958      :   Reserved Operand exception can occur.
00A2 959      :
00A2 960      :
00A2 961      :--
00A2 962
00A2 963
00A2 964      .ENTRY MTH$GREAL,      ^M<>
00A4 965      MOVG  @4(AP), R0      ; get 1st half of G COMPLEX*16 number.
00A9 966      RET
```

50 04 BC 0000 00A2 964
50FD 00A4 965
04 00A9 966


```
00AA 968 .SBTTL MTH$DIMAG
00AA 969
00AA 970 :++
00AA 971 : FUNCTIONAL DESCRIPTION:
00AA 972 : Returns the IMAGINARY part of a COMPLEX*16 argument.
00AA 973 :
00AA 974 :
00AA 975 : CALLING SEQUENCE:
00AA 976 : Imaginary_part.wd.v = MTH$DIMAG (complex_arg.rdc.r)
00AA 977 :
00AA 978 :
00AA 979 :
00AA 980 : INPUT PARAMETERS:
00AA 981 : The one argument is a COMPLEX*16 value and is call-by-reference.
00AA 982 :
00AA 983 :
00AA 984 : IMPLICIT INPUTS:
00AA 985 : NONE
00AA 986 :
00AA 987 : OUTPUT PARAMETERS:
00AA 988 : NONE
00AA 989 :
00AA 990 : IMPLICIT OUTPUTS:
00AA 991 : NONE
00AA 992 :
00AA 993 : COMPLETION CODES:
00AA 994 : NONE
00AA 995 :
00AA 996 : SIDE EFFECTS:
00AA 997 : Reserved Operand exception can occur.
00AA 998 :
00AA 999 :
00AA 1000 :--
00AA 1001
00AA 1002
00AA 1003 .ENTRY MTH$DIMAG, ^M<>
50 04 AC 0000 00AC 1004 MOVL 4(AP), R0 ; R0 -> COMPLEX*16 number
50 08 A0 70 00B0 1005 MOVD 8(R0), R0 ; R0 = imaginary part of COMPLEX*16 number
04 00B4 1006 RET
```

```
00B5 1008      .SBTTL MTH$GIMAG
00B5 1009
00B5 1010      :++
00B5 1011      : FUNCTIONAL DESCRIPTION:
00B5 1012      : Returns the IMAGINARY part of a G COMPLEX*16 argument.
00B5 1013      :
00B5 1014      :
00B5 1015      : CALLING SEQUENCE:
00B5 1016      : Imaginary_part.wg.v = MTH$GIMAG (complex_arg.rgc.r)
00B5 1017      :
00B5 1018      :
00B5 1019      :
00B5 1020      : INPUT PARAMETERS:
00B5 1021      : The one argument is a G COMPLEX*16 value and is call-by-reference.
00B5 1022      :
00B5 1023      :
00B5 1024      : IMPLICIT INPUTS:
00B5 1025      : NONE
00B5 1026      :
00B5 1027      : OUTPUT PARAMETERS:
00B5 1028      : NONE
00B5 1029      :
00B5 1030      : IMPLICIT OUTPUTS:
00B5 1031      : NONE
00B5 1032      :
00B5 1033      : COMPLETION CODES:
00B5 1034      : NONE
00B5 1035      :
00B5 1036      : SIDE EFFECTS:
00B5 1037      : Reserved Operand exception can occur.
00B5 1038      :
00B5 1039      :
00B5 1040      :--
00B5 1041
00B5 1042
00B5 1043      .ENTRY MTH$GIMAG,      ^M<>
50 04 AC 0000 00B7 1044      MOVL 4(AP), R0      ; R0 -> COMPLEX*16 number
50 08 A0 50FD 00B8 1045      MOVG 8(R0), R0      ; R0 = imaginary part of COMPLEX*16 number
04 00C0 1046      RET
```



```
00C1 1048 .SBTTL MTH$DCMPLX
00C1 1049
00C1 1050 :++
00C1 1051 : FUNCTIONAL DESCRIPTION:
00C1 1052 : Return a COMPLEX*16 number from two double-precision
00C1 1053 : floating-point values.
00C1 1054 :
00C1 1055 :
00C1 1056 : CALLING SEQUENCE:
00C1 1057 :
00C1 1058 : dcmplx.wdc.v = MTH$DCMPLX (real_part.rd.r, imag_part.rd.r)
00C1 1059 :
00C1 1060 : -or-
00C1 1061 :
00C1 1062 : CALL MTH$DCMPLX (dcmplx.wdc.r, real_part.rd.r, imag_part.rd.r)
00C1 1063 :
00C1 1064 : Because a COMPLEX*16 result cannot be expressed in 64 bits, it
00C1 1065 : is returned as the first argument, with the input parameter
00C1 1066 : displaced to the second argument, in accordance with the
00C1 1067 : Procedure Calling Standard.
00C1 1068 :
00C1 1069 : INPUT PARAMETERS:
00C1 1070 : The two input parameters are double-precision floating-point
00C1 1071 : values and are call-by-reference.
00C1 1072 :
00C1 1073 :
00C1 1074 : IMPLICIT INPUTS:
00C1 1075 : NONE
00C1 1076 :
00C1 1077 : OUTPUT PARAMETERS:
00C1 1078 : NONE
00C1 1079 :
00C1 1080 : IMPLICIT OUTPUTS:
00C1 1081 : NONE
00C1 1082 :
00C1 1083 : COMPLETION CODES:
00C1 1084 : NONE
00C1 1085 :
00C1 1086 : SIDE EFFECTS:
00C1 1087 : Reserved Operand exception can occur.
00C1 1088 :
00C1 1089 :
00C1 1090 : --
00C1 1091 :
00C1 1092 :
00C1 1093 : .ENTRY MTH$DCMPLX, ^M<>
00C3 1094 : MOVL 4(AP), R0 ; R0 -> COMPLEX*16 result
00C7 1095 : MOVD @8(AP), (R0) ; real part = 2nd arg
00CB 1096 : MOVD @12(AP), 8(R0) ; imag part = 3rd arg
00D0 1097 : RET
```

50 04 AC 0000
60 08 BC 70
08 A0 0C BC 70
04 00D0

```
00D1 1099      .SBTTL MTH$GCMPLX
00D1 1100
00D1 1101      :++
00D1 1102      FUNCTIONAL DESCRIPTION:
00D1 1103      Return a G COMPLEX*16 number from two G
00D1 1104      floating-point values.
00D1 1105
00D1 1106
00D1 1107      CALLING SEQUENCE:
00D1 1108
00D1 1109      gcplex.wgc.v = MTH$GCMPLX (real_part.rg.r, imag_part.rg.r)
00D1 1110
00D1 1111      -or-
00D1 1112
00D1 1113      CALL MTH$GCMPLX (gcplex.wgc.r, real_part.rg.r, imag_part.rg.r)
00D1 1114
00D1 1115      Because a COMPLEX*16 result cannot be expressed in 64 bits, it
00D1 1116      is returned as the first argument, with the input parameter
00D1 1117      displaced to the second argument, in accordance with the
00D1 1118      Procedure Calling Standard.
00D1 1119
00D1 1120
00D1 1121
00D1 1122      INPUT PARAMETERS:
00D1 1123      The two input parameters are G floating-point
00D1 1124      values and are call-by-reference.
00D1 1125
00D1 1126
00D1 1127      IMPLICIT INPUTS:
00D1 1128      NONE
00D1 1129
00D1 1130      OUTPUT PARAMETERS:
00D1 1131      NONE
00D1 1132
00D1 1133      IMPLICIT OUTPUTS:
00D1 1134      NONE
00D1 1135
00D1 1136      COMPLETION CODES:
00D1 1137      NONE
00D1 1138
00D1 1139      SIDE EFFECTS:
00D1 1140      Reserved Operand exception can occur.
00D1 1141
00D1 1142
00D1 1143      --
00D1 1144
00D1 1145
00D1 1146      .ENTRY MTH$GCMPLX,      ^M<>
00D3 1147      MOVL      4(AP), R0      ; R0 -> COMPLEX*16 result
00D7 1148      MOVG      @8(AP), (R0)    ; real part = 2nd arg
00DC 1149      MOVG      @12(AP), 8(R0)   ; imag part = 3rd arg
00E2 1150      RET
00E3 1151
```

50 04 AC 0000
60 08 BC 50FD
08 A0 0C BC 50FD 04


```
00E3 1153      .SBTTL  MTH$GDBLE
00E3 1154
00E3 1155      :++
00E3 1156      : FUNCTIONAL DESCRIPTION:
00E3 1157      :   Convert REAL*4 to G REAL*8 (rounded).
00E3 1158      :
00E3 1159      :
00E3 1160      : CALLING SEQUENCE:
00E3 1161      :   Conversion.wg.v = MTH$GDBLE (arg.rf.r)
00E3 1162      :
00E3 1163      :
00E3 1164      :
00E3 1165      : INPUT PARAMETERS:
00E3 1166      :   The one parameter is a F floating-point value
00E3 1167      :   and is call-by-reference.
00E3 1168      :
00E3 1169      :
00E3 1170      : IMPLICIT INPUTS:
00E3 1171      :   NONE
00E3 1172      :
00E3 1173      : OUTPUT PARAMETERS:
00E3 1174      :   NONE
00E3 1175      :
00E3 1176      : IMPLICIT OUTPUTS:
00E3 1177      :   NONE
00E3 1178      :
00E3 1179      : COMPLETION CODES:
00E3 1180      :   NONE
00E3 1181      :
00E3 1182      : SIDE EFFECTS:
00E3 1183      :   Reserved Operand exception can occur.
00E3 1184      :
00E3 1185      :
00E3 1186      :--
00E3 1187
00E3 1188
00E3 1189      .ENTRY  MTH$GDBLE,      ^M<>
50  04 BC 0000 00E5 1190      CVTFG  @4(AP), R0      ; REAL*4 to G REAL*8 in R0/R1
00E3 1191      RET
00E3 1192
00E3 1193
00E3 1194      .END
```

MTH\$CONVER
Symbol table

type conversions (FLOAT, FIX, INT) L 9

16-SEP-1984 01:11:23
6-SEP-1984 11:21:17VAX/VMS Macro V04-00
[MTHRTL.SRC]MTHCONVER.MAR;1Page 31
(30)

MTH\$AIMAG	00000031	RG	01
MTH\$CMPLX	0000003C	RG	01
MTH\$DBLE	00000047	RG	01
MTH\$DCMPLX	000000C1	RG	01
MTH\$DFLOTI	00000000	RG	01
MTH\$DFLOTJ	00000007	RG	01
MTH\$DIMAG	000000AA	RG	01
MTH\$DREAL	0000009B	RG	01
MTH\$FLOATI	0000005D	RG	01
MTH\$FLOATJ	00000064	RG	01
MTH\$GCMPLX	000000D1	RG	01
MTH\$GDBLE	000000E3	RG	01
MTH\$GFLOTI	0000006B	RG	01
MTH\$GFLOTJ	00000073	RG	01
MTH\$GIMAG	000000B5	RG	01
MTH\$GREAL	000000A2	RG	01
MTH\$IIDINT	0000001C	RG	01
MTH\$IIFIX	0000000E	RG	01
MTH\$IIGINT	0000007B	RG	01
MTH\$IIHINT	0000008B	RG	01
MTH\$IINT	0000000E	RG	01
MTH\$JIDINT	00000023	RG	01
MTH\$JIFIX	00000015	RG	01
MTH\$JIGINT	00000083	RG	01
MTH\$JIHINT	00000093	RG	01
MTH\$JINT	00000015	RG	01
MTH\$REAL	0000002A	RG	01
MTH\$SNGL	0000004E	RG	01
MTH\$SNGLG	00000055	RG	01

+-----+
! 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	000000EB (235.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG					

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:01.43
Command processing	118	00:00:00.48	00:00:02.08
Pass 1	88	00:00:01.62	00:00:05.23
Symbol table sort	0	00:00:00.02	00:00:00.10
Pass 2	190	00:00:02.07	00:00:06.89
Symbol table output	5	00:00:00.04	00:00:00.11
Psect synopsis output	4	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	436	00:00:04.32	00:00:15.86

The working set limit was 900 pages.
11441 bytes (23 pages) of virtual memory were used to buffer the intermediate code.

There were 10 pages of symbol table space allocated to hold 29 non-local and 0 local symbols.
1194 source lines were read in Pass 1, producing 89 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

_S255\$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\$:MTHCONVER/OBJ=OBJ\$:MTHCONVER MSRC\$:MTHCONVER/UPDATE=(ENH\$:MTHCONVER)

0258

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY