```
MMM         MMM  TTTTTTTTTTTTTTT  HHH         HHH  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
MMM         MMM  TTTTTTTTTTTTTTT  HHH         HHH  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
MMM         MMM  TTTTTTTTTTTTTTT  HHH         HHH  RRRRRRRRRRR     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  RRRRRRRRRRR           TTT        LLL
MMM         MMM        TTT        HHHHHHHHHHHHHHH  RRRRRRRRRRR           TTT        LLL
MMM         MMM        TTT        HHHHHHHHHHHHHHH  RRRRRRRRRRR           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
```

```
  OOOOOO    TTTTTTTTTT    SSSSSSSS  DDDDDDDD      IIIIII    VV      VV    CCCCCCCC
  OOOOOO    TTTTTTTTTT    SSSSSSSS  DDDDDDDD      IIIIII    VV      VV    CCCCCCCC
OO      OO      TT      SS          DD      DD      II      VV      VV  CC
OO      OO      TT      SS          DD      DD      II      VV      VV  CC
OO      OO      TT      SS          DD      DD      II      VV      VV  CC
OO      OO      TT      SS          DD      DD      II      VV      VV  CC
OO      OO      TT        SSSSSS    DD      DD      II      VV      VV  CC
OO      OO      TT        SSSSSS    DD      DD      II      VV      VV  CC
OO      OO      TT              SS  DD      DD      II      VV      VV  CC
OO      OO      TT              SS  DD      DD      II      VV      VV  CC
OO      OO      TT              SS  DD      DD      II        VV  VV    CC              ....
OO      OO      TT              SS  DD      DD      II        VV  VV    CC              ....
  OOOOOO        TT      SSSSSSSS    DDDDDDDD      IIIIII        VV        CCCCCCCC      ....
  OOOOOO        TT      SSSSSSSS    DDDDDDDD      IIIIII        VV        CCCCCCCC      ....


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  OTS$DIVC - COMPLEX / COMPLEX DIVISION ROUTINE
0000          .IDENT  /1-003/          ; File: OTSDIVC.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: MATH LIBRARY
0000  ;++
0000  ; ABSTRACT:
0000  ;
0000  ;       Perform COMPLEX division
0000  ;
0000  ;--
0000  ;
0000  ; VERSION: 0
0000  ;
0000  ; HISTORY:
0000  ;
0000  ; AUTHOR:
0000  ;       Jonathan Taylor, 28-Jun-77: Version 0
0000  ;
0000  ; MODIFIED BY:
0000  ;
0000  ;
0000  ;
```

```
0000          .SBTTL  HISTORY          ; Detailed Current Edit History
0000
0000
0000  ; Edit History for Version 0 of OTSDIVC
0000  ;
0000
0000  ; 0-04  - Change FOR$FLAG_JACKET to MTH$FLAG_JACKET.  TNH 17-July-78
0000  ; 1-001 - Update version number and copyright notice.  JBS 16-NOV-78
0000  ; 1-002 - Include MTHJACKET at assembly time.  JBS 07-DEC-78
0000  ; 1-003 - Add "_" to the PSECT directive.  JBS 22-DEC-78
```

```
                    .SBTTL  DECLARATIONS

;
; INCLUDE FILES:
;
;           MTHJACKET.MAR                       ; Math jacketing macro

;
; EXTERNAL SYMBOLS:
;
;           none

;
; MACROS:
;
;           none

;
; PSECT DECLARATIONS:
;
            .PSECT  _OTS$CODE       PIC, SHR, LONG, EXE, NOWRT
                                                ; Program section for OTS$ code

;
; EQUATED SYMBOLS:
;
            a       = 4                         ; real part of dividend
            b       = 8                         ; imag part of dividend
            c       = 12                        ; real part of divisor
            d       = 16                        ; imag part of divisor

;
; OWN STORAGE:
;
;           none
```

```
0000          .SBTTL  COMPLEX / COMPLEX giving COMPLEX result

0000    ;++
0000    ; FUNCTIONAL DESCRIPTION:
0000    ;
0000    ;       OTS$DIVC - COMPLEX / COMPLEX giving COMPLEX result
0000    ;
0000    ;
0000    ;       The COMPLEX result is computed as follows:
0000    ;
0000    ;       1)  Let (a, b) represent the COMPLEX dividend.
0000    ;       2)  Let (c, d) represent the COMPLEX divisor.
0000    ;       3)  Let (r, i) represent the COMPLEX quotient.
0000    ;
0000    ;       Then:
0000    ;
0000    ;       r = (ac + bd) / (cc + dd)
0000    ;       i = (bc - ad) / (cc + dd)
0000    ;
0000    ; CALLING SEQUENCE:
0000    ;
0000    ;       Complex_quotient.wfc.w = OTS$DIVC(dividend.rfc.v, divisor.rfc.v)
0000    ;
0000    ; INPUT PARAMETERS:
0000    ;
0000    ;       Dividend and divisor parameters are represented as standard
0000    ;       FORTRAN COMPLEX numbers and are CALL BY VALUE.
0000    ;
0000    ; IMPLICIT INPUTS:
0000    ;       NONE
0000    ;
0000    ; OUTPUT PARAMETERS:
0000    ;       NONE
0000    ;
0000    ; IMPLICIT OUTPUTS:
0000    ;       NONE
0000    ;
0000    ; FUNCTIONAL VALUE:
0000    ;
0000    ;       The COMPLEX value returned is (a, b) / (c, d)
0000    ;
0000    ; SIDE EFFECTS:
0000    ;
0000    ;       SS$ARITH is SIGNALed by hardware on floating zero divide
0000    ;       SS$ARITH is SIGNALed by hardware on floating overflow
0000    ;--
```

```
                    003C  0000   144          .ENTRY  OTS$DIVC, ^M<R2,R3,R4,R5>       ; disable integer overflow
                          0002   145          MTH$FLAG_JACKET                         ; establish math error handler
                          0002
        6D   00000000'GF   9E    0002          MOVAB   G^MTH$$JACKET_HND, (FP)
                          0009                                                        ; set handler address to jacket
                          0009                                                        ; handler
                          0009
             54   0C AC   7D    0009   146          MOVQ    c(AP), R4                       ; R4/R5 = divisor
                          000D   147
                          000D   148 ; Perform scaling of all operands before division
                          000D   149 ;
        52   54   08   07   EF    000D   150          EXTZV   #7, #8, R4, R2                  ; R2 = R4<exp>  0,1,...377
        51   55   08   07   EF    0012   151          EXTZV   #7, #8, R5, R1                  ; R1 = R5<exp>  0,1,...377
             52   51   B1    0017   152          CMPW    R1, R2                          ; R1 = MAX (c<exp>, d<exp>)
                  03   14    001A   153          BGTR    2$
             51   52   B0    001C   154          MOVW    R2, R1
             51   51   8E    001F   155 2$:      MNEGB   R1, R1                          ; R1 = scaling exponent  0,377,376,...,1
        51   51   07   9C    0022   156          ROTL    #7, R1, R1                      ; build a floating scale factor
                          0026   157                                                     ; scale all operands
             54   51   44    0026   158          MULF    R1, R4                          ; c
             55   51   44    0029   159          MULF    R1, R5                          ; d
        52   04 AC   51   45    002C   160          MULF3   R1, a(AP), R2                   ; a
             51   08 AC   44    0031   161          MULF    b(AP), R1                       ; b
                          0035   162
        50   54   52   45    0035   163          MULF3   R2,R4,R0                        ; R0 = ac
        53   55   51   45    0039   164          MULF3   R1, R5, R3                      ; R3 = bd
             50   53   40    003D   165          ADDF    R3, R0                          ; R0 = ac+bd
             52   55   44    0040   166          MULF    R5, R2                          ; R2 = ad
             51   54   44    0043   167          MULF    R4, R1                          ; R1 = bc
             51   52   42    0046   168          SUBF    R2, R1                          ; R1 = bc - ad
             54   54   44    0049   169          MULF    R4, R4                          ; R4 = cc
             55   55   44    004C   170          MULF    R5, R5                          ; R5 = dd
             55   54   40    004F   171          ADDF    R4, R5                          ; R5 = cc + dd
             50   55   46    0052   172          DIVF    R5, R0                          ; R0 = (ac+bd) / (cc+dd)
             51   55   46    0055   173          DIVF    R5, R1                          ; R1 = (bc-ad) / (cc+dd)
                          0058   174
                       04 0058   175          RET                                     ; (R0, R1) = (r, i)
                          0059   176
                          0059   177          .END
```

```
A               = 00000004
B               = 00000008
C               = 0000000C
MTH$$JACKET_HND   ********    X   01
OTS$DIVC          00000000 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
_OTS$CODE	00000059 (89.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:01.09
Command processing	123	00:00:00.63	00:00:04.95
Pass 1	82	00:00:00.65	00:00:03.63
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	46	00:00:00.55	00:00:01.97
Symbol table output	2	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	286	00:00:01.97	00:00:11.69

The working set limit was 750 pages.
3048 bytes (6 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 6 non-local and 1 local symbols.
237 source lines were read in Pass 1, producing 11 object records in Pass 2.
1 page of virtual memory was used to define 1 macro.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[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$:OTSDIVC/OBJ=OBJ$:OTSDIVC MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC$:
```

OTSMULCD
LIS

OTSPOWCGC
LIS

OTSDIVC
LIS

OTSPOWDD
LIS

OTSPOWCC
LIS

OTSPOWGG
LIS

OTSPOWGJ
LIS

OTSPOWCDJ
LIS

MTHTAN
LIS

MTHVECTOR
LIS

OTSDIVCG
LIS

OTSPOWCJ
LIS

OTSPOWDLU
LIS

MTHTANH
LIS

OTSMULCG
LIS

OTSPOWCGJ
LIS

OTSPOWDJ
LIS

OTSDIVCD
LIS

OTSPOWCDC
LIS