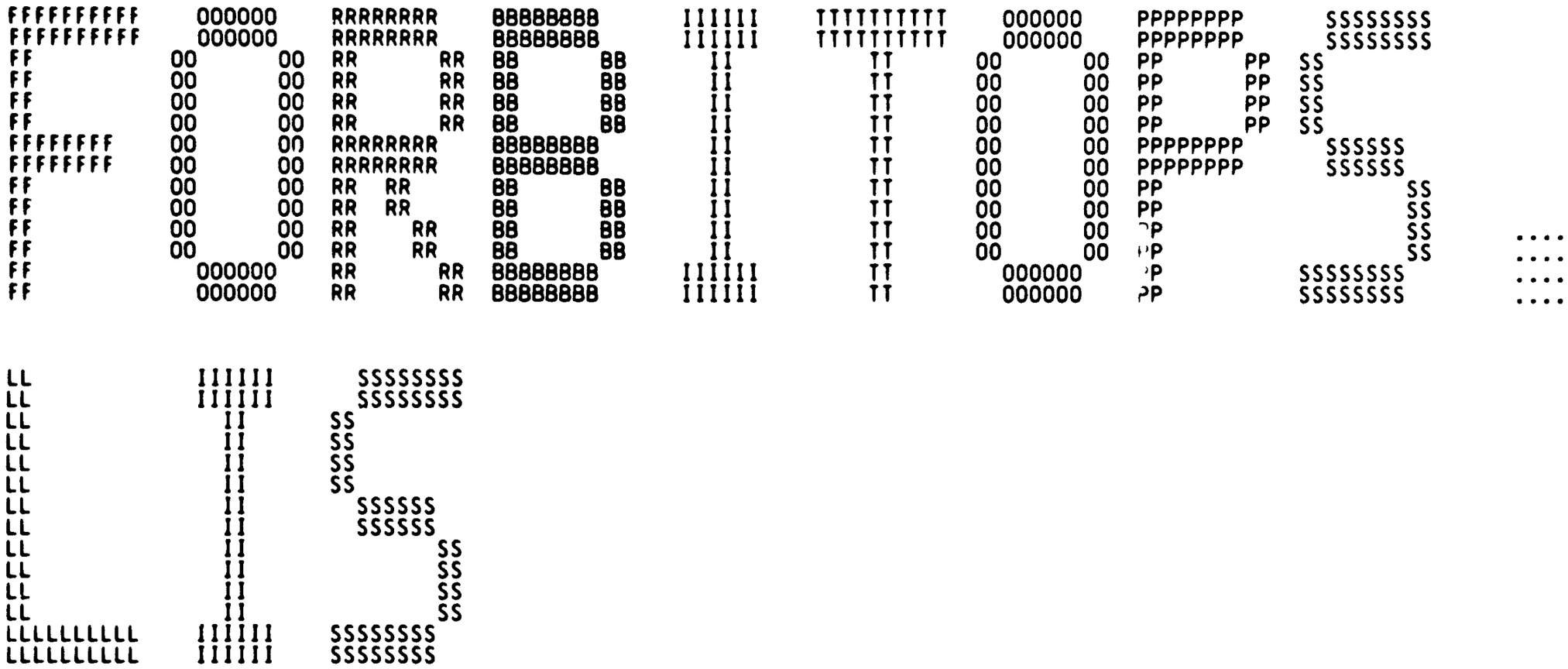
```
FFFFFFFFFFFFFFF     000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     000000000     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFF               000         000 RRR         RRR RRR         RRR       TTT        LLL
FFFFFFFFFFF       000         000 RRRRRRRRRRRR    RRRRRRRRRRRR          TTT        LLL
FFFFFFFFFFF       000         000 RRRRRRRRRRRR    RRRRRRRRRRRR          TTT        LLL
FFFFFFFFFFF       000         000 RRRRRRRRRRRR    RRRRRRRRRRRR          TTT        LLL
FFF               000         000 RRR   RRR       RRR   RRR             TTT        LLL
FFF               000         000 RRR   RRR       RRR   RRR             TTT        LLL
FFF               000         000 RRR   RRR       RRR   RRR             TTT        LLL
FFF               000         000 RRR     RRR     RRR     RRR           TTT        LLL
FFF               000         000 RRR     RRR     RRR     RRR           TTT        LLL
FFF               000         000 RRR     RRR     RRR     RRR           TTT        LLL
FFF                 000000000     RRR         RRR RRR         RRR       TTT        LLLLLLLLLLLLLLL
FFF                 000000000     RRR         RRR RRR         RRR       TTT        LLLLLLLLLLLLLLL
FFF                 000000000     RRR         RRR RRR         RRR       TTT        LLLLLLLLLLLLLLL
```

FILEID**FORBITOPS



```
0000     1          .TITLE  FOR$BITOPS      ; MIL-STD 1753 bit operations
0000     2          .IDENT  /1-002/         ; File: FORBITOPS.MAR  Edit: JAW1002
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: Fortran Support Library - user callable
0000    30 ;++
0000    31 ;
0000    32 ; ABSTRACT:
0000    33 ;       This module contains routines for operations on individual
0000    34 ;       bits of arguments.
0000    35 ;
0000    36 ;--
0000    37 ;
0000    38 ; VERSION: 1
0000    39 ;
0000    40 ; HISTORY:
0000    41 ;
0000    42 ; AUTHOR:
0000    43 ;       John A. Wheeler, 5-Jun-1981: Version 1
0000    44 ;
0000    45 ; MODIFIED BY:
0000    46 ;
0000    47 ;
0000    48 ;
```

```
                        50          .SBTTL  HISTORY                 ; Detailed Current Edit History
                        51
                        52
                        53 ; Edit History for Version 1 of FOR$BITOPS
                        54 ;
                        55 ; 1-001 - Original.  JAW 05-Jun-1981
                        56 ; 1-002 - Interpret count modulo length in FOR$IISHFTC and FOR$JISHFTC.
                        57 ;         JAW 08-Jun-1981
```

```
                        .SBTTL  DECLARATIONS

;
; INCLUDE FILES:
;
;       NONE
;
; EXTERNAL SYMBOLS:
;
;       NONE
;
; MACROS:
;
;       NONE
;
; PSECT DECLARATIONS:
;
00000000                .PSECT  _FOR$CODE       PIC, SHR, LONG, EXE, NOWRT
;
; EQUATED SYMBOLS:
;
;       NONE
;
; OWN STORAGE:
;
;       NONE
;
```

```
                                        .SBTTL  FOR$IMVBITS - Move bit field to bit field (word)

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       This routine moves a bit field contained in the first argument
                                ;       to a bit field contained in the fourth argument.  FOR$IMVBITS
                                ;       and FOR$JMVBITS implement the Fortran MIL-STD 1753 subroutine
                                ;       MVBITS.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       CALL FOR$IMVBITS(M1.rw.r, POS1.rw.r, LEN.rw.r, M2.ww.r, POS2.rw.r)
                                ;
                                ; FORMAL PARAMETERS:
                                ;
00000004                        ;       m1      = 4                     ; Address of source word
00000008                        ;       pos1    = 8                     ; Address of bit position in source
0000000C                        ;       len     = 12                    ; Address of field length
00000010                        ;       m2      = 16                    ; Address of destination word
00000014                        ;       pos2    = 20                    ; Address of bit position in destination
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       NONE
                                ;
                                ; COMPLETION STATUS:
                                ;
                                ;       NONE
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;       NONE
                                ;
                                ;--

                        0004    .ENTRY  FOR$IMvBITS, ^M<R2>     ; Entry mask
      50    08 BC     3C      MOVZWL  @pos1(AP), R0           ; R0 = source bit position
      51    0C BC     3C      MOVZWL  @len(AP), R1            ; R1 = length
      52    14 BC     3C      MOVZWL  @pos2(AP), R2           ; R2 = destination bit position
50    04 BC    51     50  EF  EXTZV   R0, R1, @m1(AP), R0     ; Extract desired bits.
10 BC    51    52     50  F0  INSV    R0, R2, R1, @m2(AP)     ; Store in destination.
                        04      RET                             ; Return to caller.
```

```
                     001B   135          .SBTTL  FOR$JMVBITS - Move bit field to bit field (longword)
                     001B   136
                     001B   137 ;++
                     001B   138 ; FUNCTIONAL DESCRIPTION:
                     001B   139 ;
                     001B   140 ;       This routine moves a bit field contained in the first argument
                     001B   141 ;       to a bit field contained in the fourth argument.  FOR$JMVBITS
                     001B   142 ;       and FOR$IMVBITS implement the Fortran MIL-STD 1753 subroutine
                     001B   143 ;       MvBITS.
                     001B   144 ;
                     001B   145 ; CALLING SEQUENCE:
                     001B   146 ;
                     001B   147 ;       CALL FOR$JMVBITS(M1.rl.r, POS1.rl.r, LEN.rl.r, M2.wl.r, POS2.rl.r)
                     001B   148 ;
                     001B   149 ; FORMAL PARAMETERS:
                     001B   150 ;
00000004             001B   151          m1      = 4                     ; Address of source longword
00000008             001B   152          pos1    = 8                     ; Address of bit position in source
0000000C             001B   153          len     = 12                    ; Address of field length
00000010             001B   154          m2      = 16                    ; Address of destination longword
00000014             001B   155          pos2    = 20                    ; Address of bit position in destination
                     001B   156 ;
                     001B   157 ; IMPLICIT INPUTS:
                     001B   158 ;
                     001B   159 ;       NONE
                     001B   160 ;
                     001B   161 ; IMPLICIT OUTPUTS:
                     001B   162 ;
                     001B   163 ;       NONE
                     001B   164 ;
                     001B   165 ; COMPLETION STATUS:
                     001B   166 ;
                     001B   167 ;       NONE
                     001B   168 ;
                     001B   169 ; SIDE EFFECTS:
                     001B   170 ;
                     001B   171 ;       NONE
                     001B   172 ;
                     001B   173 ;--
                     001B   174
                0000 001B   175          .ENTRY  FOR$JMVBITS, ^M<>       ; Entry mask
          50    0C BC  D0   001D   176          MOVL    @len(AP), R0            ; R0 = length
51  04 BC   50    08 BC  EF   0021   177          EXTZV   @pos1(AP), R0, @m1(AP), R1 ; Extract desired bits.
10 BC  50   14 BC   51   F0   0028   178          INSV    R1, @pos2(AP), R0, @m2(AP) ; Store in destination.
                  04   002F   179          RET                             ; Return to caller.
```

```
                                0030   181          .SBTTL  FOR$IIBITS - Extract bit field (word)
                                0030   182
                                0030   183 ;++
                                0030   184 ; FUNCTIONAL DESCRIPTION:
                                0030   185 ;
                                0030   186 ;       This function extracts and returns a bit field contained in the
                                0030   187 ;       first argument.  FOR$IIBITS and FOR$JIBITS implement the Fortran
                                0030   188 ;       MIL-STD 1753 function IBITS.
                                0030   189 ;
                                0030   190 ; CALLING SEQUENCE:
                                0030   191 ;
                                0030   192 ;       result.ww.v = FOR$IIBITS(M.rw.r, POS.rw.r, LEN.rw.r)
                                0030   193 ;
                                0030   194 ; FORMAL PARAMETERS:
                                0030   195 ;
00000004                        0030   196          m       = 4                     ; Address of source word
00000008                        0030   197          pos     = 8                     ; Address of bit position in source
0000000C                        0030   198          len     = 12                    ; Address of field length
                                0030   199 ;
                                0030   200 ; IMPLICIT INPUTS:
                                0030   201 ;
                                0030   202 ;       NONE
                                0030   203 ;
                                0030   204 ; IMPLICIT OUTPUTS:
                                0030   205 ;
                                0030   206 ;       NONE
                                0030   207 ;
                                0030   208 ; ROUTINE VALUE:
                                0030   209 ;
                                0030   210 ;       The specified bit field is returned in R0.
                                0030   211 ;
                                0030   212 ; SIDE EFFECTS:
                                0030   213 ;
                                0030   214 ;       NONE
                                0030   215 ;
                                0030   216 ;--
                                0030   217
                          0000  0030   218          .ENTRY  FOR$IIBITS, ^M<>        ; Entry mask
                    50    08 BC   3C  0032   219          MOVZWL  @pos(AP), R0            ; R0 = source bit position
50    04 BC   0C BC   50    EF  0036   220          EXTZV   R0, @len(AP), @m(AP), R0 ; R0 = m<pos, len>
                            04  003D   221          RET                             ; Return to caller.
```

```
                     003E   223          .SBTTL  FOR$JIBITS - Extract bit field (longword)
                     003E   224
                     003E   225 ;++
                     003E   226 ; FUNCTIONAL DESCRIPTION:
                     003E   227 ;
                     003E   228 ;       This function extracts and returns a bit field contained in the
                     003E   229 ;       first argument.  FOR$JIBITS and FOR$IIBITS implement the Fortran
                     003E   230 ;       MIL-STD 1753 function IBITS.
                     003E   231 ;
                     003E   232 ; CALLING SEQUENCE:
                     003E   233 ;
                     003E   234 ;       result.wl.v = FOR$JIBITS(M.rl.r, POS.rl.r, LEN.rl.r)
                     003E   235 ;
                     003E   236 ; FORMAL PARAMETERS:
                     003E   237 ;
00000004             003E   238          m       = 4                     ; Address of source longword
00000008             003E   239          pos     = 8                     ; Address of bit position in source
0000000C             003E   240          len     = 12                    ; Address of field length
                     003E   241 ;
                     003E   242 ; IMPLICIT INPUTS:
                     003E   243 ;
                     003E   244 ;       NONE
                     003E   245 ;
                     003E   246 ; IMPLICIT OUTPUTS:
                     003E    47 ;
                     003E   248 ;       NONE
                     003E   249 ;
                     003E   250 ; ROUTINE VALUE:
                     003E   251 ;
                     003E   252 ;       The specified bit field is returned in R0.
                     003E   253 ;
                     003E   254 ; SIDE EFFECTS:
                     003E   255 ;
                     003E   256 ;       NONE
                     003E   257 ;
                     003E   258 ;--
                     003E   259
                0000 003E   260          .ENTRY  FOR$JIBITS, ^M<>        ; Entry mask
50   04 BC   0C BC   08 BC   EF 0040   261          EXTZV   @pos(AP), @len(AP), @m(AP), R0
                     0048   262                                          ; R0 = m<pos, len>
                       04 0048   263          RET                             ; Return to caller.
```

```
                     0049   265          .SBTTL  FOR$IISHFTC - Circular shift of low-order bits (word)
                     0049   266
                     0049   267 ;++
                     0049   268 ; FUNCTIONAL DESCRIPTION:
                     0049   269 ;
                     0049   270 ;       This function returns the first argument after shifting the low-
                     0049   271 ;       order len bits by cnt positions.  FOR$IISHFTC and FOR$JISHFTC
                     0049   272 ;       implement the Fortran MIL-STD 1753 function ISHFTC.  The count
                     0049   273 ;       is taken modulo length if !cnt! > len.
                     0049   274 ;
                     0049   275 ;
                     0049   276 ; CALLING SEQUENCE:
                     0049   277 ;
                     0049   278 ;       result.ww.v = FOR$IISHFTC(M.rw.r, CNT.rw.r, LEN.rw.r)
                     0049   279 ;
                     0049   280 ; FORMAL PARAMETERS:
                     0049   281 ;
00000004             0049   282          m       = 4                     ; Address of source word
00000008             0049   283          cnt     = 8                     ; Address of shift count
0000000C             0049   284          len     = 12                    ; Address of field length
                     0049   285 ;
                     0049   286 ; IMPLICIT INPUTS:
                     0049   287 ;
                     0049   288 ;       NONE
                     0049   289 ;
                     0049   290 ; IMPLICIT OUTPUTS:
                     0049   291 ;
                     0049   292 ;       NONE
                     0049   293 ;
                     0049   294 ; ROUTINE VALUE:
                     0049   295 ;
                     0049   296 ;       The first argument with its low-order len bits shifted by cnt
                     0049   297 ;       positions is returned in R0.
                     0049   298 ;
                     0049   299 ; SIDE EFFECTS:
                     0049   300 ;
                     0049   301 ;       NONE
                     0049   302 ;
                     0049   303 ;--
                     0049   304
                001C 0049   305          .ENTRY  FOR$IISHFTC, ^M<R2, R3, R4> ; Entry mask
50   04 BC     3C   004B   306          MOVZWL  @m(AP), R0              ; R0 = word containing field
51   0C BC     3C   004F   307          MOVZWL  @len(AP), R1            ; R1 = field length
52   08 BC     32   0053   308          CVTWL   @cnt(AP), R2            ; R2 = shift count
        0E     11   0057   309          BRB     COMM1                   ; Join common code.
                    0059   310
```

```
                     0059   312          .SBTTL  FOR$JISHFTC - Circular shift of low-order bits (longword)
                     0059   313
                     0059   314 ;++
                     0059   315 ; FUNCTIONAL DESCRIPTION:
                     0059   316 ;
                     0059   317 ;       This function returns the first argument after shifting the low-
                     0059   318 ;       order len bits by cnt positions.  FOR$JISHFTC and FOR$IISHFTC
                     0059   319 ;       implement the Fortran MIL-STD 1753 function ISHFTC.  The count
                     0059   320 ;       is taken modulo length if |cnt| > len.
                     0059   321 ;
                     0059   322 ; CALLING SEQUENCE:
                     0059   323 ;
                     0059   324 ;       result.wl.v = FOR$JISHFTC(M.rl.r, CNT.rl.r, LEN.rl.r)
                     0059   325 ;
                     0059   326 ; FORMAL PARAMETERS:
                     0059   327 ;
00000004             0059   328          m       = 4                     ; Address of source longword
00000008             0059   329          cnt     = 8                     ; Address of shift count
0000000C             0059   330          len     = 12                    ; Address of field length
                     0059   331 ;
                     0059   332 ; IMPLICIT INPUTS:
                     0059   333 ;
                     0059   334 ;       NONE
                     0059   335 ;
                     0059   336 ; IMPLICIT OUTPUTS:
                     0059   337 ;
                     0059   338 ;       NONE
                     0059   339 ;
                     0059   340 ; ROUTINE VALUE:
                     0059   341 ;
                     0059   342 ;       The first argument with its low-order len bits shifted by cnt
                     0059   343 ;       positions is returned in R0.
                     0059   344 ;
                     0059   345 ; SIDE EFFECTS:
                     0059   346 ;
                     0059   347 ;       NONE
                     0059   348 ;
                     0059   349 ;--
                001C 0059   350          .ENTRY  FOR$JISHFTC, ^M<R2, R3, R4> ; Entry mask
50   04 BC     D0   005B   351          MOVL    @m(AP), R0              ; R0 = longword containing field
51   0C BC     D0   005F   352          MOVL    @len(AP), R1            ; R1 = field length
52   08 BC     D0   0063   353          MOVL    @cnt(AP), R2            ; R2 = shift count
                     0067   354 ;+
                     0067   355 ; Enter here from FOR$IISHFTC
                     0067   356 ;-
     03   18     0067   357 COMM1:  BGEQ    10$                     ; If cnt < 0,
52   51   C0     0069   358          ADDL    R1, R2                  ;  add len to cnt.
51   52   D1     006C   359 10$:    CMPL    R2, R1                  ; Is 0 <= cnt <= len?
     13   1B     006F   360          BLEQU   30$                     ; Branch if yes.
                     0071   361 ;+
                     0071   362 ; Here if the count is (still) negative, or greater than len.  Reduce it
                     0071   363 ; to the range [0, len].
                     0071   364 ;-
54   52   D0     0071   365          MOVL    R2, R4                  ; R4 = tmp = cnt
     05   18     0074   366          BGEQ    20$                     ; If tmp < 0,
54   51   C2     0076   367          SUBL    R1, R4                  ;  decrease it by len - 1
     54   D6     0079   368          INCL    R4                      ;  so division rounds downward.
```

```
                54   51   C6  007B   369 20$:    DIVL    R1, R4                  ; R4 = tmp/len
                54   51   C4  007E   370         MULL    R1, R4                  ; R4 = (tmp/len)*len
                52   54   C2  0081   371         SUBL    R4, R2                  ; R2 = cnt-(tmp/len)*len)
                                0084   372 ;+
                                0084   373 ; The reduced count is now in R2.  Do the actual shift by locating the
                                0084   374 ; boundary between the "high part" and the "low part" of the field, and
                                0084   375 ; exchanging the two parts.
                                0084   376 ;-
           53   51   52   C3  0084   377 30$:    SUBL3   R2, R1, R3              ; R3 = boundary position
      51   50   53   00   EF  0088   378         EXTZV   #0, R3, R0, R1          ; Extract low-order part.
      54   50   52   53   EF  008D   379         EXTZV   R3, R2, R0, R4          ; Extract high-order part.
      50   53   52   51   F0  0092   380         INSV    R1, R2, R3, R0          ; Make low-order part high.
      50   52   00   54   F0  0097   381         INSV    R4, #0, R2, R0          ; Make high-order part low.
                                04  009C   382         RET                     ; Return to caller.
```

```
                     009D    384          .SBTTL  FOR$BITEST - Test single bit (word)
                     009D    385
                     009D    386 ;++
                     009D    387 ; FUNCTIONAL DESCRIPTION:
                     009D    388 ;
                     009D    389 ;       This function returns .TRUE. if the specified bit in the first
                     009D    390 ;       argument is 1 and .FALSE. if it is 0.  FOR$BITEST and FOR$BJTEST
                     009D    391 ;       implement the Fortran MIL-STD 1753 function BTEST.
                     009D    392 ;
                     009D    393 ; CALLING SEQUENCE:
                     009D    394 ;
                     009D    395 ;       result.ww.v = FOR$BITEST(M.rw.r, POS.rw.r)
                     009D    396 ;
                     009D    397 ; FORMAL PARAMETERS:
                     009D    398 ;
00000004             009D    399          m       = 4                     ; Address of source word
00000008             009D    400          pos     = 8                     ; Address of bit position
                     009D    401 ;
                     009D    402 ; IMPLICIT INPUTS:
                     009D    403 ;
                     009D    404 ;       NONE
                     009D    405 ;
                     009D    406 ; IMPLICIT OUTPUTS:
                     009D    407 ;
                     009D    408 ;       NONE
                     009D    409 ;
                     009D    410 ; ROUTINE VALUE:
                     009D    411 ;
                     009D    412 ;       .TRUE. or .FALSE.
                     009D    413 ;
                     009D    414 ; SIDE EFFECTS:
                     009D    415 ;
                     009D    416 ;       NONE
                     009D    417 ;
                     009D    418 ;--
                     009D    419
                0000 009D    420          .ENTRY  FOR$BITEST, ^M<>        ; Entry mask
          50    08 BC   3C   009F    421          MOVZWL  @pos(AP), R0            ; R0 = source bit position
50   04 BC    01    50   EE   00A3    422          EXTV    R0, #1, @m(AP), R0      ; R0 = SEXT(specified bit)
                04         00A9    423          RET                             ; Return to caller.
```

```
                        00AA   425          .SBTTL  FOR$BJTEST - Test single bit (longword)
                        00AA   426
                        00AA   427 ;++
                        00AA   428 ; FUNCTIONAL DESCRIPTION:
                        00AA   429 ;
                        00AA   430 ;       This function returns .TRUE. if the specified bit in the first
                        00AA   431 ;       argument is 1 and .FALSE. if it is 0.  FOR$BJTEST and FOR$BITEST
                        00AA   432 ;       implement the Fortran MIL-STD 1753 function BTEST.
                        00AA   433 ;
                        00AA   434 ; CALLING SEQUENCE:
                        00AA   435 ;
                        00AA   436 ;       result.wl.v = FOR$BJTEST(M.rl.r, POS.rl.r)
                        00AA   437 ;
                        00AA   438 ; FORMAL PARAMETERS:
                        00AA   439 ;
00000004                00AA   440          m       = 4                     ; Address of source longword
00000008                00AA   441          pos     = 8                     ; Address of bit position
                        00AA   442 ;
                        00AA   443 ; IMPLICIT INPUTS:
                        00AA   444 ;
                        00AA   445 ;       NONE
                        00AA   446 ;
                        00AA   447 ; IMPLICIT OUTPUTS:
                        00AA   448 ;
                        00AA   449 ;       NONE
                        00AA   450 ;
                        00AA   451 ; ROUTINE VALUE:
                        00AA   452 ;
                        00AA   453 ;       .TRUE. or .FALSE.
                        00AA   454 ;
                        00AA   455 ; SIDE EFFECTS:
                        00AA   456 ;
                        00AA   457 ;       NONE
                        00AA   458 ;
                        00AA   459 ;--
                        00AA   460
                  0000  00AA   461          .ENTRY  FOR$BJTEST, ^M<>        ; Entry mask
50   04 BC   01   08 BC   EE   00AC   462          EXTV    @pos(AP), #1, @m(AP), R0 ; R0 = SEXT(specified bit)
                        04   00B3   463          RET                             ; Return to caller.
                             00B4   464
```

```
                        00B4    466          .SBTTL  FOR$IIBSET - Set single bit (word)
                        00B4    467
                        00B4    468 ;++
                        00B4    469 ; FUNCTIONAL DESCRIPTION:
                        00B4    470 ;
                        00B4    471 ;       This function returns the first argument with the specified bit
                        00B4    472 ;       set.  FOR$IIBSET and FOR$JIBSET implement the Fortran MIL-STD
                        00B4    473 ;       1753 function IBSET.
                        00B4    474 ;
                        00B4    475 ; CALLING SEQUENCE:
                        00B4    476 ;
                        00B4    477 ;       result.ww.v = FOR$IIBSET(M.rw.r, POS.rw.r)
                        00B4    478 ;
                        00B4    479 ; FORMAL PARAMETERS:
                        00B4    480 ;
00000004                00B4    481          m       = 4                     ; Address of source word
00000008                00B4    482          pos     = 8                     ; Address of bit position
                        00B4    483 ;
                        00B4    484 ; IMPLICIT INPUTS:
                        00B4    485 ;
                        00B4    486 ;       NONE
                        00B4    487 ;
                        00B4    488 ; IMPLICIT OUTPUTS:
                        00B4    489 ;
                        00B4    490 ;       NONE
                        00B4    491 ;
                        00B4    492 ; ROUTINE VALUE:
                        00B4    493 ;
                        00B4    494 ;       The first argument with the specified bit set is returned in R0.
                        00B4    495 ;
                        00B4    496 ; SIDE EFFECTS:
                        00B4    497 ;
                        00B4    498 ;       NONE
                        00B4    499 ;
                        00B4    500 ;--
                        00B4    501
                  0000  00B4    502          .ENTRY  FOR$IIBSET, ^M<>        ; Entry mask
50    04 BC       3C    00B6    503          MOVZWL  @m(AP), R0              ; R0 = word containing bit
51    08 BC       3C    00BA    504          MOVZWL  @pos(AP), R1            ; R1 = source bit position
00 50    51       E2    00BE    505          BBSS    R1, R0, 10$             ; Set specified bit.
                   04    00C2    506 10$:     RET                             ; Return to caller
                         00C3    507
```

```
                                .SBTTL  FOR$JIBSET - Set single bit (longword)

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;       This function returns the first argument with the specified bit
                                ;       set.  FOR$JIBSET and FOR$IIBSET implement the Fortran MIL-STD
                                ;       1753 function IBSET.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       result.wl.v = FOR$JIBSET(M.rl.r, POS.rl.r)
                                ;
                                ; FORMAL PARAMETERS:
                                ;
                00000004        m       = 4                     ; Address of source longword
                00000008        pos     = 8                     ; Address of bit position
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       NONE
                                ;
                                ; ROUTINE VALUE:
                                ;
                                ;       The first argument with the specified bit set is returned in R0.
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;       NONE
                                ;
                                ;--

                        0000    .ENTRY  FOR$JIBSET, ^M<>        ; Entry mask
    50    04 BC       D0        MOVL    @m(AP), R0              ; R0 = longword containing bit
00  50    08 BC       E2        BBSS    @pos(AP), R0, 10$       ; Set specified bit.
                        04      RET                             ; Return to caller
```

```
                     00CF    551          .SBTTL  FOR$IIBCLR - Clear single bit (word)
                     00CF    552
                     00CF    553 ;++
                     00CF    554 ; FUNCTIONAL DESCRIPTION:
                     00CF    555 ;
                     00CF    556 ;       This function returns the first argument with the specified bit
                     00CF    557 ;       cleared.  FOR$IIBCLR and FOR$JIBCLR implement the Fortran
                     00CF    558 ;       MIL-STD 1753 function IBCLR.
                     00CF    559 ;
                     00CF    560 ; CALLING SEQUENCE:
                     00CF    561 ;
                     00CF    562 ;       result.ww.v = FOR$IIBCLR(M.rw.r, POS.rw.r)
                     00CF    563 ;
                     00CF    564 ; FORMAL PARAMETERS:
                     00CF    565 ;
00000004             00CF    566          m       = 4                     ; Address of source word
00000008             00CF    567          pos     = 8                     ; Address of bit position
                     00CF    568 ;
                     00CF    569 ; IMPLICIT INPUTS:
                     00CF    570 ;
                     00CF    571 ;       NONE
                     00CF    572 ;
                     00CF    573 ; IMPLICIT OUTPUTS:
                     00CF    574 ;
                     00CF    575 ;       NONE
                     00CF    576 ;
                     00CF    577 ; ROUTINE VALUE:
                     00CF    578 ;
                     00CF    579 ;       The first argument with the specified bit cleared is returned in
                     00CF    580 ;       R0.
                     00CF    581 ;
                     00CF    582 ; SIDE EFFECTS:
                     00CF    583 ;
                     00CF    584 ;       NONE
                     00CF    585 ;
                     00CF    586 ;--
                     00CF    587
                0000 00CF    588          .ENTRY  FOR$IIBCLR, ^M<>        ; Entry mask
50    04 BC     3C   00D1    589          MOVZWL  @m(AP), R0              ; R0 = word containing bit
51    08 BC     3C   00D5    590          MOVZWL  @pos(AP), R1            ; R1 = source bit position
00 50   51      E5   00D9    591          BBCC    R1, R0, 10$             ; Clear specified bit.
                04   00DD    592 10$:     RET                             ; Return to caller
                     00DE    593
```

```
                     00DE    595          .SBTTL  FOR$JIBCLR - Clear single bit (longword)
                     00DE    596
                     00DE    597 ;++
                     00DE    598 ; FUNCTIONAL DESCRIPTION:
                     00DE    599 ;
                     00DE    600 ;       This function returns the first argument with the specified bit
                     00DE    601 ;       cleared.  FOR$JIBCLR and FOR$IIBCLR implement the Fortran
                     00DE    602 ;       MIL-STD 1753 function IBCLR.
                     00DE    603 ;
                     00DE    604 ; CALLING SEQUENCE:
                     00DE    605 ;
                     00DE    606 ;       result.wl.v = FOR$JIBCLR(M.rl.r, POS.rl.r)
                     00DE    607 ;
                     00DE    608 ; FORMAL PARAMETERS:
                     00DE    609 ;
00000004             00DE    610          m       = 4                     ; Address of source longword
00000008             00DE    611          pos     = 8                     ; Address of bit position
                     00DE    612 ;
                     00DE    613 ; IMPLICIT INPUTS:
                     00DE    614 ;
                     00DE    615 ;       NONE
                     00DE    616 ;
                     00DE    617 ; IMPLICIT OUTPUTS:
                     00DE    618 ;
                     00DE    619 ;       NONE
                     00DE    620 ;
                     00DE    621 ; ROUTINE VALUE:
                     00DE    622 ;
                     00DE    623 ;       The first argument with the specified bit cleared is returned in
                     00DE    624 ;       R0.
                     00DE    625 ;
                     00DE    626 ; SIDE EFFECTS:
                     00DE    627 ;
                     00DE    628 ;       NONE
                     00DE    629 ;
                     00DE    630 ;--
                     00DE    631
                0000 00DE    632          .ENTRY  FOR$JIBCLR, ^M<>        ; Entry mask
    50    04 BC    D0 00E0    633          MOVL    @m(AP), R0              ; R0 = longword containing bit
00  50    08 BC    E5 00E4    634          BBCC    @pos(AP), R0, 10$       ; Clear specified bit.
                04  00E9    635 10$:     RET                             ; Return to caller
                    00EA    636
                    00EA    637          .END                            ; End of module FOR$BITOPS
```

```
CNT             =  00000008
COMM1              00000067  R      01
FOR$BITEST         0000009D  RG     01
FOR$BJTEST         000000AA  RG     01
FOR$IIBCLR         000000CF  RG     01
FOR$IIBITS         00000030  RG     01
FOR$IIBSET         000000B4  RG     01
FOR$IISHFTC        00000049  RG     01
FOR$IMVBITS        00000000  RG     01
FOR$JIBCLR         000000DE  RG     01
FOR$JIBITS         0000003E  RG     01
FOR$JIBSET         000000C3  RG     01
FOR$JISHFTC        00000059  RG     01
FOR$JMVBITS        0000001B  RG     01
LEN             =  0000000C
M               =  00000004
M1              =  00000004
M2              =  00000010
POS             =  00000008
POS1            =  00000008
POS2            =  00000014
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_FOR$CODE	000000EA (234.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.10	00:00:00.74
Command processing	136	00:00:00.49	00:00:04.40
Pass 1	81	00:00:01.18	00:00:03.28
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	109	00:00:01.25	00:00:04.04
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	363	00:00:03.08	00:00:12.53

```
The working set limit was 1050 pages.
8031 bytes (16 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 21 non-local and 7 local symbols.
637 source lines were read in Pass 1, producing 43 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.
```

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$:FORBITOPS/OBJ=OBJ$:FORBITOPS MSRC$:FORBITOPS/UPDATE=(ENH$:FORBITOPS)
```

COMRS0WD
LIS

FORDATEDS
LIS

FORDECOMO
LIS

FORCLB
LIS

COMSETST
LIS

FORASSOC
LIS

FORCLOSEF
LIS

FORDATE
LIS

FORCLOSE
LIS

FORDECOMF
LIS

FORDELETE
LIS

COMRAD50
LIS

COMUSEREX
LIS

FORBITOPS
LIS

FORDEFINE
LIS

FORBACKSP
LIS

FORDISPA
LIS

FORCUTRT
LIS