K 6

```
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
FFFFFFFFFFFF     000         000  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFFFFFFFFFFF     000         000  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFFFFFFFFFFF     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
```

```
FFFFFFFFFF    OOOOOO    RRRRRRRR    RRRRRRRR      AAAAAA    NN      NN  DDDDDDDD      OOOOOO    MM      MM
FFFFFFFFFF    OOOOOO    RRRRRRRR    RRRRRRRR      AAAAAA    NN      NN  DDDDDDDD      OOOOOO    MM      MM
FF          OO      OO  RR      RR  RR      RR  AA      AA  NN      NN  DD      DD  OO      OO  MMMM  MMMM
FF          OO      OO  RR      RR  RR      RR  AA      AA  NN      NN  DD      DD  OO      OO  MMMM  MMMM
FF          OO      OO  RR      RR  RR      RR  AA      AA  NNNN    NN  DD      DD  OO      OO  MM  MM  MM
FF          OO      OO  RR      RR  RR      RR  AA      AA  NNNN    NN  DD      DD  OO      OO  MM  MM  MM
FFFFFFFF    OO      OO  RRRRRRRR    RRRRRRRR    AA      AA  NN  NN  NN  DD      DD  OO      OO  MM      MM
FFFFFFFF    OO      OO  RRRRRRRR    RRRRRRRR    AA      AA  NN  NN  NN  DD      DD  OO      OO  MM      MM
FF          OO      OO  RR  RR      RR  RR      AAAAAAAAAA  NN    NNNN  DD      DD  OO      OO  MM      MM
FF          OO      OO  RR  RR      RR  RR      AAAAAAAAAA  NN    NNNN  DD      DD  OO      OO  MM      MM
FF          OO      OO  RR    RR    RR    RR    AA      AA  NN      NN  DD      DD  OO      OO  MM      MM      ....
FF          OO      OO  RR    RR    RR    RR    AA      AA  NN      NN  DD      DD  OO      OO  MM      MM      ....
FF            OOOOOO    RR      RR  RR      RR  AA      AA  NN      NN  DDDDDDDD      OOOOOO    MM      MM      ....
FF            OOOOOO    RR      RR  RR      RR  AA      AA  NN      NN  DDDDDDDD      OOOOOO    MM      MM      ....

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  FOR$RANDOM        ; random number generator and interfaces
0000             .IDENT  /1-003/           ; File: FORRANDOM.MAR  Edit: SBL1003
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: FORTRAN SYSTEM LIBRARY
0000    ;++
0000    ; ABSTRACT:
0000    ;
0000    ;       Provide entry points for:
0000    ;               FOR$IRAN
0000    ;               FOR$RANDU_W
0000    ;
0000    ;       The algorithm used is copied exactly from PDP-11 FORTRAN
0000    ;       library so the same sequences will be generated.
0000    ;--
0000    ;
0000    ; VERSION: 1-001
0000    ;
0000    ; HISTORY:
0000    ;
0000    ; AUTHOR:
0000    ;       Jonathan M. Taylor, 12-Aug-77: Version 0
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;
0000    ;
```

```
0000                .SBTTL  HISTORY         ; Detailed Current Edit History
0000    ; Edit History for Version 0 of FOR$RANDOM
0000    ;
0000    ; 0-3   - use word offset to call for$JRAN TNH 16-SEP-77
0000    ; 0-4   - add a bug from 11 routine to make compatible:
0000    ;       - now tests only second parameter for 0 (first call),
0000    ;       - instead of concatenated longword JMT 6-OCT-77
0000    ; 0-5   - JRAN is now passed only one longword arg.  JMT 9-Oct-77
0000    ; 0-6   - Copy back seed as 2 words or 1 long word.  TNH 14-Nov-77
0000    ; 0-9   - Remove FOR$JRAN which is no longer supported.
0000    ;         FORTRAN compiler now generates calls to MTH$RANDOM.  JMT 4-Jan-78
0000    ; 0-10  - Bug fix 0-4 didn't break my code enough to be
0000    ;         compatable with the 11.  JMT 16-Feb-78
0000    ; 0-11  - Remove FOR$FLAG_JACKET.  TNH 11-July-78
0000    ; 1-001 - Update version number and copyright notice.  JBS 16-NOV-78
0000    ; 1-002 - Add " " to the PSECT directive.  JBS 22-DEC-78
0000    ; 1-003 - Use .ENTRY.  SBL 1-Jul-1983
```

```
                  0000          .SBTTL  DECLARATIONS
                  0000
                  0000  ;
                  0000  ; INCLUDE FILES:
                  0000  ;       oerr.mar
                  0000
                  0000  ;
                  0000  ; EXTERNAL SYMBOLS:
                  0000  ;
                  0000
                  0000  ;
                  0000  ; MACROS:
                  0000  ;
                  0000
                  0000  ;
                  0000  ; PSECT DECLARATIONS:
              00000000          .PSECT  _FOR$CODE       PIC, SHR, EXE, LONG, NOWRT
                  0000
                  0000  ;
                  0000  ; EQUATED SYMBOLS:
        00000004  0000          a1      =       4       ; offset into AP of address of arg1
        00000008  0000          a2      =       8       ; offset into AP of address of arg2
        0000000C  0000          a3      =       12      ; (optional) offset into AP of add-
                  0000                                  ; ress of output
                  0000
                  0000  ;
                  0000  ; OWN STORAGE:
                  0000  ;       NONE
```

```
                         0000          .SBTTL  FOR$RANDU and FOR$RANDU_W       return number as parameter
                         0000
                         0000   ;++
                         0000   ; FUNCTIONAL DESCRIPTION:
                         0000   ;
                         0000   ;      CALLs FOR$IRAN to get a random number and returns it in
                         0000   ;      third parameter.
                         0000   ;
                         0000   ; CALLING SEQUENCE:
                         0000   ;      CALL FOR$RANDU   (gen_base_1.ml.r, gen_base_2.ml.r,
                         0000   ;                       random_fraction.wf.r)
                         0000   ;
                         0000   ;      CALL FOR$RANDU_W (gen_base_1.mw.r, gen_base_2.mw.r,
                         0000   ;                       random_fraction.wf.r)
                         0000   ;
                         0000   ; INPUT PARAMETERS:
                         0000   ;      gen_base_1               seed1 for algorithm
                         0000   ;      gen_base_2               seed2 for algorithm
                         0000   ;
                         0000   ; IMPLICIT INPUTS:
                         0000   ;      NONE
                         0000   ;
                         0000   ; OUTPUT PARAMETERS:
                         0000   ;      random_fraction          floating point result is
                         0000   ;                               between 0 and 1
                         0000   ;
                         0000   ; IMPLICIT OUTPUTS:
                         0000   ;      NONE
                         0000   ;
                         0000   ; COMPLETION CODES:
                         0000   ;      NONE
                         0000   ;
                         0000   ; SIDE EFFECTS:
                         0000   ;      NONE
                         0000   ;
                         0000   ;--
                         0000
                         0000
                         0000
                         0000   FOR$RANDU_W::
                   0000  0000          .ENTRY  FOR$RANDU, ^M<>
  000C'CF     6C     FA  0002          CALLG   (AP), W^FOR$IRAN             ; R0 = floating result
    0C BC     50     50  0007          MOVF    R0, @a3(AP)             ; return as third parameter
                     04  000B          RET
```

```
                         000C              .SBTTL  FOR$IRAN          result in R0
                         000C
                         000C      ;++
                         000C      ; FUNCTIONAL DESCRIPTION:
                         000C      ;
                         000C      ;       SEED = arg1,arg2
                         000C      ;       if arg2 = 0 then SEED = 1        ; first call only
                         000C      ;       SEED = SEED * (2**16 + 3)
                         000C      ;       arg1,arg2 = SEED                 ; return for later calls
                         000C      ;       R0 = SEED normalized to floating point
                         000C      ;
                         000C      ; CALLING SEQUENCE:
                         000C      ;       Random_fraction.wf.v = FOR$IRAN (gen_base_1.mw.r,
                         000C      ;                                         gen_base_2.mw.r)
                         000C      ;
                         000C      ; INPUT PARAMETERS:
                         000C      ;       gen_base_1              seed1 for algorithm
                         000C      ;       gen_base_2              seed2 for algorithm
                         000C      ;
                         000C      ; IMPLICIT INPUTS:
                         000C      ;       NONE
                         000C      ;
                         000C      ; OUTPUT PARAMETERS:
                         000C      ;       NONE
                         000C      ;
                         000C      ; IMPLICIT OUTPUTS:
                         000C      ;       NONE
                         000C      ;
                         000C      ; COMPLETION CODES:
                         000C      ;       NONE
                         000C      ;
                         000C      ; SIDE EFFECTS:
                         000C      ;       NONE
                         000C      ;
                         000C      ; FUNCTIONAL VALUE:
                         000C      ;       A floating-point value between 0 and 1
                         000C      ;--
                         000C
                         000C
                         000C
                   0000  000C              .ENTRY  FOR$IRAN, ^M<>
         50   04 BC  B0  000E              MOVW    @a1(AP), R0          ; R0 = arg1
      50   50    10  9C  0012              ROTL    #16, R0, R0          ; build a longword value
         50   08 BC  B0  0016              MOVW    @a2(AP), R0          ; in R0
                         001A                                           ; NOTE: PDP-11 algorithm only checks
                         001A                                           ; bits 15:0 for 0, so VAX is compatibile
                 23  13  001A              BEQL    20$                  ; branch if first call
51   50  00010003 8F  C5  001C              MULL3   #^X10003, R0, R1     ; R1 = R0 *((2**16)+3) = SEED
         00 51   1F  E5  0024              BBCC    #31, R1, 15$         ; make sure SEED positive
         50   51 4E      0028      15$::   CVTLF   R1, R0               ; R0 = floating (SEED) binary point
                         002B                                           ; to right of bit 0
     50  00003100 8F  44  002B              MULF    #^X3100, R0          ; R0 = R0 * 2**-31 = normalized, binary poin
                         0032                                           ; to right of bit 31
                         0032                                           ; R0 = floating point result
                         0032                                           ; R1 = new seed
         08 BC   51  B0  0032              MOVW    R1, @a2(AP)          ; return bits 15:0 of seed
      51   51    10  9C  0036              ROTL    #16, R1, R1          ; seed<31:16> to R1<15:0>
```

```
         04 BC    51   B0  003A   202          MOVW    R1, @4(AP)             ; return bits 31:16 of seed as first arg
                       04  003E   203          RET                            ; return with R0 = floating random number
   50   00010000 8F    C0  003F   204 20$:     ADDL    #^X10000, R0           ; this is what the 11 did!
              50  03   B0  0046   205          MOVW    #3, R0
              51  50   D0  0049   206          MOVL    R0, R1
                  DA   11  004C   207          BRB     15$
                           004E   208
                           004E   209
                           004E   210          .END
```

```
A1               = 00000004
A2               = 00000008
A3               = 0000000C
FOR$IRAN           0000000C RG    01
FOR$RANDU          00000000 RG    01
FOR$RANDU_W        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
_FOR$CODE                       0000004E  (    78.)  01 (  1.)    PIC    USR    CON    PEL    LCL   SHR   EXE   RD   NOWRT NOVEC LONG

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

Phase                     Page faults   CPU Time       Elapsed Time
-----                     -----------   --------       ------------
Initialization                    29    00:00:00.08    00:00:00.35
Command processing               120    00:00:00.50    00:00:02.42
Pass 1                            68    00:00:00.59    00:00:01.29
Symbol table sort                  0    00:00:00.00    00:00:00.00
Pass 2                            50    00:00:00.44    00:00:01.94
Symbol table output                2    00:00:00.01    00:00:00.01
Psect synopsis output              3    00:00:00.02    00:00:00.02
Cross-reference output             0    00:00:00.00    00:00:00.00
Assembler run totals             274    00:00:01.64    00:00:06.03

The working set limit was 900 pages.
2400 bytes (5 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 2 local symbols.
210 source lines were read in Pass 1, producing 14 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.

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

Macro library name                             Macros defined
------------------                             --------------
_$255$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$:FORRANDOM/OBJ=OBJ$:FORRANDOM MSRC$:FORRANDOM/UPDATE=(ENH$:FORRANDOM)
```

0182 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY