DATA GENERAL
CORPORATION
Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

Random Number Generator

TAPES

ASCII Source: 090-000040

ABSTRACT

This routine generates a (pseudo) random sequence of integers in the range $0 \leqslant N \leqslant 2**16-1$.

093-000036-00

1. REQUIREMENTS

1.1 Memory

1K or larger alterable memory

1.2 Equipment

NOVA central processor

1.3 External Subroutines

None

1.4 Other

None

2. OPERATING PROCEDURE

2.1 Calling Sequence

```
JSR  .RAND
address of old value
return
```

2.2 Input Format

The address of the previous random value (or initially a starting value) must be provided in the word after the JSR .RAND.

2.3 Output Format

The new 16-bit random result will be returned in ACØ and will also replace the previous value in memory.

2.4 Error Returns

None

2.5 State of Active Registers upon Exit

AC1 and AC2 are unchanged. ACØ, AC3 and Carry are destroyed.

2.6 Cautions to User

If a K-bit number ($1 \leq K < 16$) is needed, use the most significant K bits (the least significant K bits are not as random). For example, to obtain random N mod 2, use the sign bit of the result.

3. DISCUSSION

3.1 Algorithms

This routine generates a linear, congruential sequence of the form

X(N + 1) = (X(N) * A + C) mod 2**16

where N = (2**11 + 2**2 + 1)

and C = 33031 (octal)

3.2 Limitations and Accuracy

The resulting sequence has the maximum period of 2**16.

3.3 Size and Timing

The routine is 36 (octal) words in length.

Execution time is 244.7 μ seconds.

3.4 References

Donald E. Knuth, The Art of Computer Programming, Volume 2, Addison-Wesley Publishing Company (1969), Chapter 3.

3.5 Flow Diagrams

None

4. EXAMPLES AND APPLICATIONS

Although this routine produces numbers in the range

$$\emptyset \leq N \leq 2^{**}16 - 1$$

fractional values in the range

$$Ø \leq F < 1$$

can be obtained by the simple computation

$$F \leftarrow N/2^{**}16$$

Random numbers are useful in a variety of applications - simulation of natural phenomena, random sampling, testing the effectiveness of computer algorithms, and probability theory to name a few.

The source of .RAND is provided with the NOVA software. If a user routine requires this program, the tape should be edited into the user source.

5. PROGRAM LISTING

A listing of .RAND follows. No origin is given in the source, enabling the user to edit this tape anywhere within his routines.

```
             ; RANDOM NUMBER GENERATOR
             ; GENERATES A (PSEUDO) RANDOM SEQUENCE OF INTEGERS
             ; IN THE RANGE  Ø<= N <= 2**16-1

             ; INPUT:        ADDRESS OF OLD VALUE POINTED TO BY WORD
             ;               AFTER JSR

             ; OUTPUT:       16-BIT NEW RANDOM NUMBER IN ACØ
             ;               AND IN STORAGE REPLACING OLD VALUE

             ; CALLING SEQUENCE:
             ;       JSR     .RAND
             ;       ADDRESS OF OLD VALUE
             ;       RETURN

             ; METHOD:       GENERATES A LINEAR CONGRUENTIAL
             ;               SEQUENCE OF THE FORM:
             ;                       X(N+1) = (X(N)*A+C)MOD 2**16

             ; CAUTION:      IF A K-BIT RANDOM NUMBER (AS OPPOSED
             ;               TO A 16- BIT NUMBER) IS NEEDED,
             ;               USE THE MOST SIGNIFIGANT
             ;               K-BITS (THE LEAST SIGNIFIGANT K-BITS
             ;               ARE NOT AS RANDOM).
             ;               FOR EXAMPLE, TO OBTAIN RANDOM N MOD 2,
             ;               USE THE SIGN BIT OF THE RESULT

             ; UNCHANGED:    AC1, AC2
             ; DESTROYED:    ACØ, AC3, CARRY

ØØØØØ Ø54Ø32 .RAND:  STA 3,.UDØ3     ; SAVE RETURN
ØØØØ1 Ø1ØØ32         ISZ .UDØ3       ; BUMP PAST ADDRESS CONSTANT
ØØØØ2 Ø44Ø3Ø         STA 1,.UDØ1     ; *SAVE AC1
ØØØØ3 Ø5ØØ31         STA 2,.UDØ2     ; *SAVE AC2
ØØØØ4 Ø314ØØ         LDA 2,Ø,3       ; GET ADDRESS OF OLD VALUE
ØØØØ5 Ø21ØØØ         LDA Ø,Ø,2       ; OLD VALUE TO ACØ
ØØØØ6 ØØ4Ø15         JSR .UD5Ø       ; N*A
ØØØØ7 Ø34Ø34         LDA 3,.UD2Ø     ; GET INCREMENT, C
ØØØ1Ø 163ØØØ         ADD 3,Ø         ; (N*A+C) MOD 2**16
ØØØ11 Ø41ØØØ         STA Ø,Ø,2       ; STORE IT
ØØØ12 Ø24Ø3Ø         LDA 1,.UDØ1     ; *RESTORE AC1
ØØØ13 Ø3ØØ31         LDA 2,.UDØ2     ; *RESTORE AC2
ØØØ14 ØØ2Ø32         JMP @.UDØ3      ; RETURN
```

```
                ; COMPUTE N*(2**11+2**2+1)

00015 024035 .UD50:   LDA 1,.UD21       ; GET COUNT, X
00016 044033          STA 1,.UD10       ; FOR ITERATION
00017 105120          MOVZL 0,1         ; N*2**(X+1)
00020 125120          MOVZL 1,1
00021 014033          DSZ .UD10
00022 000020          JMP .-2
00023 107000          ADD 0,1
00024 125120          MOVZL 1,1
00025 125120          MOVZL 1,1         ; 2**2*(N*2**(X+1)+N)
00026 123000          ADD 1,0
00027 001400          JMP 0,3           ; N+N*2**2+N*2**(X+3)



00030 000000 .UD01:   0                 ; *SAVE AC1
00031 000000 .UD02:   0                 ; *SAVE AC2
00032 000000 .UD03:   0                 ; SAVE RETURN

00033 000000 .UD10:   0                 ; ITERATION COUNT

00034 033031 .UD20:   33031             ; INCREMENT
00035 000010 .UD21:   10                ; ITERATION COUNT, X
```