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

PROGRAM

Relocatable Math Library File

TAPES

Library Binary: 099-000001

ABSTRACT

This document provides a brief description of all the routines available using Data General's math library tape 099-000001. These descriptions are in alphabetical order according to function. All the information necessary to "CALL" these routines is provided in this document. For further reference and a listing of a particular routine, the user should consult the appropriate DGC write-up.

093-000041-00

A brief description of all the relocatable math library subroutines available with tape number 099-000001, *ie*, given in this document. They appear in alphabetical order according to function. Each description will elaborate on the items explained below.

PURPOSE: Explains the function of the routine.

TITLE: Gives the name of the routine (necessary for editing the library tape).

INPUT: Describes necessary input format.

OUTPUT: Describes results of the routine.

CALLING SEQUENCE AND ENTRY POINTS:
The relocatable routines contained in the math library are called by declaring the appropriate entry point as a normal external within the user program. For example, to call double precision absolute value:

```
        EXTN      .DABS
          .
          .
          .
        JSR       @DUMMY
          .
          .
          .
DUMMY:  .DABS
```

The names of the entry point(s) are given for all routines, and all user calls must use the above method.

ERROR CONDITIONS:
Explains or cautions about the idiosyncrasies of the routine.

CARRY AND REGISTERS:
Gives state of active registers upon exit.

LENGTH AND TIME:
Gives the number of instructions used to write this routine and also gives an approximation of execution time.

REFERENCE:
Gives the number of the DGC manual to be referenced if more information is needed.

ABSOLUTE VALUE (DOUBLE PRECISION)

PURPOSE:
This routine computes the absolute value of a double precision, fixed point, two's complement number.

TITLE:
The title is .DABS.

INPUT:
A number in AC0 (high order), AC1 (low order).

OUTPUT:
The absolute value of the input returned in AC0, AC1 - high order in AC0, low order in AC1.

CALLING SEQUENCE AND ENTRY POINT:
Indirect at .DABS, with normal return to the instruction following the call.

ERROR CONDITIONS:
Caution: The absolute value of -2**31 cannot be represented and is returned unchanged.

CARRY AND REGISTERS:
AC0, AC1, and Carry are destroyed; AC2 and AC3 remain unchanged.

LENGTH AND TIME:
This routine consists of 6 instructions and is normally relocatable.
For $X \geq 0$, execution is 8.2 μ seconds.
For $X < 0$, execution is 19.4 μ seconds.

REFERENCE:
Manual #093-000009

ABSOLUTE VALUE (SINGLE PRECISION)

PURPOSE:

This routine computes the absolute value of a fixed point, single precision, two's complement number.

TITLE:

The title is .ABS.

INPUT:

The input is a single precision number in AC0.

OUTPUT:

The absolute value of the input is returned in AC0.

CALLING SEQUENCE AND ENTRY POINT:

Indirect to .ABS with normal return to the instruction following the call.

ERROR CONDITIONS:

The absolute value of -2**15 cannot be represented and will be returned unchanged.

CARRY AND REGISTERS:

The Carry and AC0 may be destroyed; AC1, AC2, and AC3 are unchanged.

LENGTH AND TIME:

This routine consists of 3 words and is normally relocatable.
For $X \geq 0$, execution time is 8.2 μ seconds.
For $X < 0$, execution time is 13.8 μ seconds.

REFERENCE:

Manual #093-000006

ADDITION (DOUBLE PRECISION)

PURPOSE:

This routine computes the sum of two double precision, two's complement integers.

TITLE:

The title is .DADD.

INPUT:

The first operand must be in AC0, AC1 (high order, low order). The second operand must be in storage, higher order word followed by lower order word. The address of the higher order word of the second operand must be given after the JSR @DUMMY.

OUTPUT:

The double precision sum will be returned in AC0, AC1 (high order, low order).

CALLING SEQUENCE AND ENTRY POINT:

Indirect at .DADD, then address of second operand with return to the instruction following the second operand address.

ERROR CONDITIONS:

Caution: No check is made for overflow.

CARRY AND REGISTERS:

AC0, AC1, AC3, and Carry are destroyed; AC2 remains unchanged.

LENGTH AND TIME:

This routine consists of 15 (octal) instructions and is normally relocatable.
Execution time is 54.9 μ seconds.

REFERENCE:

Manual #093-000012

BCD to BINARY (DOUBLE PRECISION)

PURPOSE:
This routine converts a double precision number in BCD to binary.

TITLE:
The title is .DBCB.

INPUT:
A double precision integer is passed in AC0, AC1 (high order, low order) of maximum value of 99999999 decimal.

OUTPUT:
The binary equivalent of the input is returned in AC0, AC1 (high order, low order).

CALLING SEQUENCE AND ENTRY POINT:
Indirect at .DBCB with normal return to the instruction following the call.

ERROR CONDITIONS:
If a digit greater than 9 is encountered in the input, Carry will be set and AC0 will contain the bad digit. Otherwise, Carry will be zero.

CARRY AND REGISTERS:
AC0, AC1, AC3, and Carry are destroyed; AC2 is unchanged.

LENGTH AND TIME:
This routine consists of 76 (octal) words and is normally relocatable.
Execution time is 2.174 milliseconds.

REFERENCE:
Manual #093-000029

BCD to BINARY (SINGLE PRECISION)

PURPOSE:
This routine converts a single precision number in BCD to binary.

TITLE:
The title is .BCDB.

INPUT:
A BCD integer in AC1 (maximum value 9999 decimal).

OUTPUT:
Binary equivalent of BCD integer is returned in AC1.

CALLING SEQUENCE AND ENTRY POINT:
Indirect at .BCDB with normal return to the instruction following the call.

ERROR CONDITIONS:
If a digit greater than binary 1001 is encountered in the input, Carry will be set, AC1 will be unchanged, and AC0 will contain the bad digit. Otherwise, Carry will be zero on return.

CARRY AND REGISTERS:
AC0, AC1, AC3, and Carry are destroyed; AC2 is unchanged.

LENGTH AND TIME:
This routine consists of 53 (octal) words and is normally relocatable.
Execution time is 1.034 milliseconds.

REFERENCE:
Manual #093-000023

BINARY to BCD (DOUBLE PRECISION)

PURPOSE:
This routine converts a double precision binary number to a BCD number.

TITLE:
The title is .DBBC.

INPUT:
A positive, double precision binary number in AC0, AC1 (high order, low order).

OUTPUT:
The BCD equivalent is in AC0, AC1 (high order, low order).

CALLING SEQUENCE AND ENTRY POINT:
Indirect to .DBBC with normal return to the instruction following the call.

ERROR CONDITIONS:
If AC0, AC1 contains a number greater than 99999999, no conversion will take place and Carry will be set. Otherwise, Carry will be reset.

CARRY AND REGISTERS:
AC0, AC1, AC3, and Carry are destroyed; AC2 is unchanged.

LENGTH AND TIME:
This routine consists of 57 (octal) words and is normally relocatable.

REFERENCE:
Manual #093-000030

BINARY to BCD (SINGLE PRECISION)

PURPOSE:
This routine converts a binary number to its BCD equivalent.

TITLE:
The title is .BBCD.

INPUT:
An unsigned binary number in AC1.

OUTPUT:
The BCD equivalent in AC1.

CALLING SEQUENCE AND ENTRY POINT:
Indirect to .BBCD with normal return to the instruction following the call.

ERROR CONDITIONS:
If a number greater than 9999 is input for conversion, no conversion will take place and Carry will be set. Otherwise, Carry will be zero.

CARRY AND REGISTERS:
AC1, AC3 and Carry are destroyed; AC0, and AC2 are unchanged.

LENGTH AND TIME:
This routine is 41 (octal) words and is normally relocatable. Execution time is 273.8 + N * 14.1 μ seconds, where N is the sum of the digits of the result.

REFERENCE:
Manual #093-000024

BINARY to DECIMAL (DOUBLE PRECISION)

PURPOSE:

This routine converts a double precision two's complement number to an ASCII decimal character string.

TITLE:

The title is .DBD.

INPUT:

A double precision, two's complement integer is passed in AC1, AC2 (high order, low order).

OUTPUT:

ASCII character string of the form:

```
          +DDDDDDDDDD(NULL)
    or    -DDDDDDDDDD(NULL)
```

is outputted. Characters are passed right adjusted, bit 8 = 0, in AC0 to a user routine whose address must be stored in location 41 of page 0.

CALLING SEQUENCE AND ENTRY POINT:

Indirect to .DBD with normal return to the instruction following the call.

CARRY AND REGISTERS:

AC1, AC2, AC3, and Carry are destroyed; AC0 remains unchanged.

LENGTH AND TIME:

This routine consists of 112 (octal) words and is normally relocatable.
Execution time is 1.061 + N * .047 milliseconds where N is the sum of the digits of the result.

REFERENCE:

Manual #093-000032

BINARY to DECIMAL (SINGLE PRECISION)

PURPOSE:

This routine converts a single precision two's complement number to an ASCII character string.

TITLE:

The title is .BIND.

INPUT:

A single precision, two's complement integer is passed in AC1.

OUTPUT:

An ASCII character string terminated by a null word. Characters are passed right adjusted in AC0 to the routine whose address must be in location 41 of page 0. The string is of the form:

```
        +DDDDD(NULL)
or      -DDDDD(NULL)
```

CALLING SEQUENCE AND ENTRY POINT:

Indirect at .BIND with normal return to the instruction following the call.

CARRY AND REGISTERS:

AC1, AC3 and Carry are destroyed; AC0 and AC2 remain unchanged.

LENGTH AND TIME:

This routine consists of 51 (octal) words and is normally relocatable.
Execution time is (378.3 + N * 14.1) μ seconds, where N is the sum of the digits of the result.

REFERENCE:

Manual #093-000026

BINARY to GRAY CODE

PURPOSE:
This routine computes the Gray Code equivalent of a 16-bit binary word.

TITLE:
The title is .BGRY.

INPUT:
A binary word in AC0.

OUTPUT:
Gray Code equivalent in AC0.

CALLING SEQUENCE AND ENTRY POINT:
Indirect to .BGRY with normal return to the instruction following the call.

CARRY AND REGISTERS:
AC0, AC3, and Carry are destroyed; AC1, AC2 are unchanged.

LENGTH AND TIME:
This routine consists of 13 (octal) words and is normally relocatable.
Execution time is 50.3 μ seconds.

REFERENCE:
Manual #093-000034

BINARY to OCTAL (SINGLE PRECISION)

PURPOSE:

This routine converts a 16-bit binary word to an octal ASCII character string.

TITLE:

The title is .BINO.

INPUT:

A 16-bit binary number is passed in AC1.

OUTPUT:

An ASCII character string terminated by a null character. Characters are passed right adjusted in AC0 to the user routine whose address must be stored in location 41 of page 0. The string is of the form:

000000(NULL)

where "0" represents octal digits.

CALLING SEQUENCE AND ENTRY POINT:

Indirect to .BINO with normal return to the instruction following the call.

CARRY AND REGISTERS:

AC0, AC1, AC3, and Carry are destroyed; AC2 remains unchanged.

LENGTH AND TIME:

This routine consists of 27 (octal) words and is normally relocatable.

Execution time is 367.6 + N * 20.0 μ seconds, where N is the sum of the digits of the result (the sum expressed in decimal).

REFERENCE:

Manual #093-000028

DECIMAL to BINARY (DOUBLE PRECISION)

PURPOSE:

This routine converts an ASCII character string to a double precision binary number.

TITLE:

The title is .DDB.

INPUT:

Calls a get character routine whose address must be stored in location 40 of page 0. Characters must be returned right adjusted in AC0 with bit 8 = 0. + is optional for positive numbers; - must be given for negative numbers. Input is of the form:

SDD...DD(break)

S is the sign (if given), D is a decimal digit. The break character is any character other than a decimal digit.

OUTPUT:

AC0 contains the break character, while AC1, AC2 contain the binary results of the conversion.

CALLING SEQUENCE AND ENTRY POINTS:

Indirect to .DDB with normal return to the instruction following the call.

If an indication is needed, entry is made by indirect to .DDBI with normal return. An ASCII "D" followed by a null word will be transmitted via AC0 to a put character routine whose address must be in location 41 of page 0.

ERROR CONDITIONS:

Caution: The absolute value of the result is N MOD 2**31.

CARRY AND REGISTERS:

All accumulators and Carry are destroyed.

LENGTH AND TIME:

This routine consists of 77 (octal) words and is normally relocatable.

Execution time is approximately 124.7 + I * 125.5 μ seconds.

REFERENCE:

Manual #093-000031

DECIMAL to BINARY (SINGLE PRECISION)

PURPOSE:

This routine converts ASCII characters to a single precision binary number.

TITLE:

The title is .DBIN.

INPUT:

Input characters will be requested by calling a user "get a character" routine whose address must be stored in location 40 of page zero. This user routine must be provided. ASCII characters should be returned, right adjusted in AC0 with bit 8 = 0. This routine need not save any registers or Carry. Input should be in the form:

S D D . . . D D (break)

where "S" represents the sign ("-" or optionally "+"), D represents an ASCII decimal digit, and "break" is any ASCII character other than a digit.

OUTPUT:

Upon exit, AC0 will contain the ASCII break character and AC1 will contain the single precision, two's complement binary equivalent of the input.

CALLING SEQUENCE AND ENTRY POINTS:

Indirect to .DBIN with normal return to the instruction following the call.

If an indication is desired to signal characters are requested, the calling sequence is indirect to .DBNI. An ASCII "S" followed by a null character will be transmitted via AC0 to a "put a character" routine whose address must be in location 41 of page 0.

ERROR CONDITIONS:

Caution: The absolute value of the result is N MOD 2**15.
For example: +96741 converts to +31205.
-2**15 converts to 0.

CARRY AND REGISTERS:

AC0, AC1, AC3, and Carry are destroyed; AC2 is unchanged.

LENGTH AND TIME:

This routine consists of 65 (octal) words and is normally relocatable.
Execution time is approximately 110 + I * 82.2 μ seconds, where I is the number of digits in the input.

REFERENCE:

Manual #093-000025

DIVIDE - SIGNED (DOUBLE PRECISION)

PURPOSE:

This routine divides two signed, multiple precision numbers.

TITLE:

The title is .DDIV.

INPUT:

The double precision divisor must be in AC0, AC1 (high order, low order). The quadruple precision dividend should be stored in four consecutive words, highest order to lowest order. AC2 must contain the address of highest order word of the dividend.

OUTPUT:

The double precision quotient is returned in AC0, AC1 (high order, low order). Its sign is determined by the algebraic rules for signed division. The double precision remainder is stored in two consecutive memory words, the higher order word first. AC2 will contain the address of the higher order remainder word. The remainder is signed as the dividend.

CALLING SEQUENCE AND ENTRY POINT:

Indirect at .DDIV with normal return to the instruction following the call.

ERROR CONDITIONS:

If the magnitude of the quotient would exceed 2**31-1, an error condition exists, Carry is set, and return is made with unpredictable results. For legal divisions, Carry will be zero on return.

CARRY AND REGISTERS:

All the accumulators and Carry are destroyed.

LENGTH AND TIME:

Double precision Multiply and Divide are in one program. Both consist of 213 (octal) words and are normally relocatable. Execution time for signed double divide is 2.98 milliseconds.

REFERENCE:

Manual #093-000011

DIVIDE - SIGNED (SINGLE PRECISION)

PURPOSE:
Divides two fixed points, two's complement numbers.

TITLE:
The title is .DIV.

INPUT:
Dividend in AC0 (high order) and AC1 (low order).
Divisor in AC2.

OUTPUT:
Quotient in AC1. Remainder in AC0 (same sign as dividend).

CALLING SEQUENCE AND ENTRY POINT:
Indirect to .DIV with normal return to the instruction following the call.

ERROR CONDITIONS:
If the magnitude of the quotient exceeds 2**15-1, Carry is set and the dividend remains unchanged. Otherwise, Carry will be zero.

CARRY AND REGISTERS:
AC0, AC1, AC3, and Carry are destroyed; AC2 remains unchanged.

LENGTH AND TIME:
This routine consists of 45 (octal) words and is normally relocatable.
Total average execution time is 605 μ seconds.

REFERENCE:
Manual #093-000008

DIVIDE - UNSIGNED

PURPOSE:
This routine divides two unsigned numbers.

TITLE:
The title is .DIVU.

INPUT:
The dividend is passed in AC0 (high order) and AC1 (low order). If .DIVI is called, the dividend is passed in AC1.

OUTPUT:
The single precision quotient is returned in AC1. The remainder is returned in AC0.

CALLING SEQUENCE AND ENTRY POINTS:
For a double precision dividend by a single precision divisor use indirect to .DIVU.
For a single precision dividend by a single precision divisor use indirect to .DIVI. In both cases, return is to the instruction following the call.

ERROR CONDITIONS:
If a quotient greater than 2**16-1 would result from the division, Carry is set and no division is attempted. If the division is successful, Carry will be zero on return.

CARRY AND REGISTERS:
AC0, AC1, AC3, and Carry are destroyed; AC2 remains unchanged.

LENGTH AND TIME:
This routine consists of 21 (octal) words and is normally relocatable.
Average execution time is 483 μ seconds.

REFERENCE:
Manual #093-000016

GRAY CODE to BINARY

PURPOSE:
This routine computes the binary equivalent of a 16-bit Gray Code word.

TITLE:
The title is .GRYB.

INPUT:
Input is a Gray Code word in AC0.

OUTPUT:
The binary equivalent is returned in AC0.

CALLING SEQUENCE AND ENTRY POINTS:
Indirect to .GRYB with normal return to the instruction following the call.

CARRY AND REGISTERS:
AC0, AC3 and Carry are destroyed; AC1, AC2 are unchanged.

LENGTH AND TIME:
This routine consists of 22 (octal) words and is normally relocatable.
Execution time is 536.4 μ seconds.

REFERENCE:
Manual #093-000035

LOGICAL EXCLUSIVE OR

PURPOSE:
This routine computes the exclusive OR of two unsigned numbers.

TITLE:
The title is .XOR.

INPUT:
One 16-bit quantity is passed in AC0, the second in AC1.

OUTPUT:
The exclusive OR of the two quantities is returned in AC0.

CALLING SEQUENCE AND ENTRY POINT:
Indirect to .XOR with normal return to the instruction following the call.

CARRY AND REGISTERS:
AC0, AC3, and Carry are destroyed; AC1 and AC2 are unchanged.

LENGTH AND TIME:
This routine is 7 words and is normally relocatable.
Execution time is 34.0 μ seconds.

REFERENCE:
Manual #093-000021

LOGICAL INCLUSIVE OR

PURPOSE:
This routine computes the logical inclusive OR of two unsigned numbers.

TITLE:
The title to reference this routine is .OR.

INPUT:
One 16-bit quantity is passed in AC0, the second in AC1.

OUTPUT:
The inclusive OR of the two quantities is returned in AC0.

CALLING SEQUENCE AND ENTRY POINT:
Indirect to .OR with normal return to the instruction following the call.

CARRY AND REGISTERS:
AC0 is destroyed; AC1, AC2, AC3, and Carry are unchanged.

LENGTH AND TIME:
This routine consists of 5 words and is normally relocatable. Execution time is 25.6 μ seconds.

REFERENCE:
Manual #093-000022

MULTIPLY - SIGNED (DOUBLE PRECISION)

PURPOSE:
Multiplies two signed, double precision numbers.

TITLE:
The title is .DPMD.

INPUT:
One double precision operand must be provided in AC0, AC1 (high order, low order). The address of the first word of the second operand must be supplied in the word following the JSR @DUMMY.

OUTPUT:
The quadruple precision product is stored in four consecutive memory locations, highest order to lowest order. AC2 will contain the address of the first and highest order word of the result.

CALLING SEQUENCE AND ENTRY POINT:
Indirect at .DMPY followed by the address of first word of two word multiplier. Return is to the instruction following the multiplier address.

CARRY AND REGISTERS:
All accumulators and Carry are destroyed.

LENGTH AND TIME:
Double precision multiply and divide are one program. Both consist of 213 (octal) words and are normally relocatable. Total average execution time is 1.62 milliseconds.

REFERENCE:
Manual #093-000010

MULTIPLY - SIGNED (SINGLE PRECISION)

PURPOSE:
This routine multiplies two, fixed point, single precision, two;s complement numbers.

TITLE:
The title is .MPY.

INPUT:
One fixed point, single precision operand is passed in AC1, the second in AC2.

OUTPUT:
The double precision result is returned in AC0 (high order) and AC1 (low order).

CALLING SEQUENCE AND ENTRY POINT:
Indirect to .MPY with normal return to the instruction following the call.

CARRY AND REGISTERS:
AC0, AC1, AC3, and Carry are destroyed, AC2 remains unchanged.

LENGTH AND TIME:
This routine consists of 16 (octal) words and is normally relocatable.
For execution time in addition to unsigned multiply, 56.4 μ seconds.
For execution time in unsigned multiply, 340 μ seconds.
Total average execution time is 396.4 μ seconds.

REFERENCE:
Manual #093-000007

MULTIPLY - UNSIGNED

PURPOSE:
This routine multiplies two unsigned, fixed point, single precision numbers.

TITLE:
The title is .MPYU.

INPUT:
The unsigned multiplicand is passed in AC2, the multiplier in AC1. If entry is made to .MPYA, the product is added to the contents of AC0 to form the final result.

OUTPUT:
The 32-bit result will be returned in AC0, AC1 (high order, low order).

CALLING SEQUENCE AND ENTRY POINTS:
To multiply AC1 by AC2, indirect to .MPYU.
To multiply AC1 by AC2 and add the result to AC0, indirect to .MPYA. In both cases, return is to the instruction following the call.

CARRY AND REGISTERS:
AC0, AC1, and AC3 are destroyed; AC2 and Carry remain unchanged.

LENGTH AND TIME:
This routine consists of 14 (octal) words and is normally relocatable.
Average execution time is 340 μ seconds.

REFERENCE:
Manual #093-000015

NEGATE (DOUBLE PRECISION)

PURPOSE:

This routine computes -D where D is a double precision two's complement integer.

TITLE:

The title is .DNEG.

INPUT:

A double precision, two's complement number in AC0, AC1 (high order, low order).

OUTPUT:

The negative of the input is returned in AC0 (high order), AC1 (low order).

CALLING SEQUENCE AND ENTRY POINT:

Indirect to .DNEG with normal return to the instruction following the call.

ERROR CONDITIONS:

Caution: The negative of -2**31 cannot be represented and is returned unchanged.

CARRY AND REGISTERS:

AC0, AC1, AC3, and Carry are destroyed; AC2 remains unchanged.

LENGTH AND TIME:

This routine consists of 4 words and is normally relocatable. Execution time is 13.8 μ seconds.

REFERENCE:

Manual #093-000014

OCTAL to BINARY (SINGLE PRECISION)

PURPOSE:

This routine converts an ASCII octal character string to a binary number.

TITLE:

The title is .OBIN.

INPUT:

Input characters will be requested by calling a user "get a character" routine whose address must be stored in location 40 of page 0. This user routine must be provided. Upon call, this routine should return an ASCII character, right adjusted in AC0 with bit 8 = 0. Input should be of the form:

00...00(break)

where the "0" represent octal digits.

OUTPUT:

AC0 contains the break character, and AC1 contains the binary number (MOD 200000 octal).

CALLING SEQUENCE AND ENTRY POINTS:

Indirect to .OBIN.

If an indication is desired to signal characters are requested, the calling sequence is indirect to .OBNI. An ASCII "0" followed by a null character will be transmitted via AC0 to a user put character routine whose address must be stored in location 41 of page 0. In both cases, return is to the first word after the call.

ERROR CONDITIONS:

Caution: Result is N MOD 200000 (octal), *eg*, 576452* converts to 176452.

CARRY AND REGISTERS:

AC0, AC1, AC3, and Carry are destroyed; AC2 is unchanged.

LENGTH AND TIME:

This routine consists of 42 (octal) words and is normally relocatable.

Execution time for .OBIN is 63.0 + I * 70.2 μ seconds, where I represents the number of digits in the input.

REFERENCE:

Manual #093-000027

PARITY GENERATOR

PURPOSE:
This routine computes the even parity bit over a 16-bit number and returns the bit in Carry.

TITLE:
The title is .PRTY.

INPUT:
A 16-bit number is passed in AC0.

OUTPUT:
The even parity bit over the contents of AC0 will be returned in Carry.

CALLING SEQUENCE AND ENTRY POINT:
Indirect to .PRTY with return to the word following the call.

CARRY AND REGISTERS:
AC3 and Carry are destroyed; AC0, AC1, AC2 remain unchanged.

LENGTH AND TIME:
This routine consists of 16 (octal) words and is normally relocatable.
Average execution time is 215.4 μ seconds.

REFERENCE:
Manual #093-000033

RANDOM NUMBER GENERATOR

PURPOSE:

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

TITLE:

The title is .RAND.

INPUT:

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

OUTPUT:

The new 16-bit random result will be returned in AC0 and will also replace the previous value in memory.

CALLING SEQUENCE AND ENTRY POINT:

Indirect to .RAND followed by the address of the old value. Return will be to the instruction after the address parameter.

ERROR CONDITIONS:

Caution: If a K-bit number ($1<K<16$) 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.

CARRY AND REGISTERS:

AC0, AC3, and Carry are destroyed; AC1 and AC2 are unchanged.

LENGTH AND TIME:

This routine consists of 36 (octal) words and is normally relocatable.

Execution time is 244.7 μ seconds.

REFERENCE:

Manual #093-000036

SUBTRACTION (DOUBLE PRECISION)

PURPOSE:

This routine computes the difference of two double precision two's complement integers.

TITLE:

The title is .DSUB.

INPUT:

The minuend is passed in AC0, AC1, (high order, low order). The subtrahend must be in two consecutive memory words, higher order followed by lower order. The word following the JSR should contain the address of the higher order word of the subtrahend.

OUTPUT:

The double precision difference is returned in AC0, AC1 (high order, low order).

CALLING SEQUENCE AND ENTRY POINT:

Indirect to .DSUB followed by the address of higher order word of subtrahend. Return will be to the instruction following the address parameter.

ERROR CONDITIONS:

Caution: no check is made for overflow.

CARRY AND REGISTERS:

AC0, AC1, AC3, and Carry are destroyed; AC2 remains unchanged.

LENGTH AND TIME:

This routine is 15 (octal) words and is normally relocatable. Execution time is 54.9 μ seconds.

REFERENCE:

Manual #093-000013