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

PROGRAM

Double Precision Addition

TAPES

ASCII Source: 090-000017 -01

ABSTRACT

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

093-000012-01

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  .DADD
address of higher order word of augend
return
```

2.2 Input Format

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 .DADD.

2.3 Output Format

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

2.4 Error Returns

None

2.5 State of Active Registers upon Exit

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

2.6 Cautions to User

No check is made for overflow. The magnitude of the result should not exceed 2**31-1 to insure accuracy.

3. DISCUSSION

3.1 Algorithms

Double addition performs an addition of low order terms followed by an addition of high order terms. Since the low order terms in two's complement notation can be considered as unsigned numbers, a low order add that produces a carry causes the high order result to be incremented by 1.

3.2 Limitations and Accuracy

The routine is exact for all results provided the magnitude does not exceed 2**31-1.

3.3 Size and Timing

.DADD is 15 (octal) words in length.

Execution time is 54.9 μ seconds.

3.4 References

See Section 2.2 of "How to Use the NOVA" for a further discussion of double precision addition.

3.5 Flow Diagrams

None

4. EXAMPLES AND APPLICATIONS

An ASCII source of .DADD is provided with the NOVA software. This tape can be edited directly into user software that requires double precision addition.

```
                ; DOUBLE ADDITION
                ; COMPUTES THE SUM OF TWO DOUBLE PRECISION TWO'S
                ;       COMPLEMENT INTEGERS

                ; INPUT:         D1 IN AC0, AC1 (HIGH AND LOW)
                ;                ADDRESS OF D2 IN WORD AFTER JSR

                ; OUTPUT:        D1+D2 IN AC0, AC1 (HIGH AND LOW)

                ; CALLING SEQUENCE:
                ;       JSR     .DADD
                ;       ADDRESS OF SECOND OPERAND
                ;       RETURN

                ; CAUTION:       NO CHECK IS MADE FOR OVERFLOW

                ; UNCHANGED:     AC2
                ; DESTROYED:     AC0, AC1, AC3, CARRY

00000 054014 .DADD: STA 3,.BD03     ; SAVE RETURN
00001 010014        ISZ .BD03       ; BUMP PAST ADDRESS CONSTANT
00002 050013        STA 2,.BD02     ; *SAVE AC2
00003 035400        LDA 3,0,3       ; ADDRESS OF D2
00004 031400        LDA 2,0,3       ; HIGH ORDER OF D2
00005 035401        LDA 3,1,3       ; LOW ORDER OF D2
00006 167022        ADDZ 3,1,SZC    ; LOW ORDER ADD
00007 101400        INC 0,0         ; CARRY TO HIGH ORDER
00010 143000        ADD 2,0         ; HIGH ORDER ADD
00011 030013        LDA 2,.BD02     ; *RESTORE AC2
00012 002014        JMP @.BD03      ; AND RETURN


00013 000000 .BD02: 0               ; *SAVE AC2
00014 000000 .BD03: 0               ; SAVE RETURN
```