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

PROGRAM

# Single Precision Signed Multiply

TAPES

ASCII Source: 090-000013

ABSTRACT

This routine multiplies two fixed point, single precision, two's complement numbers to form a double precision, two's complement product.



093-000007-00

## 1. REQUIREMENTS

### 1.1 Memory

1K or larger alterable memory

### 1.2 Equipment

NOVA central processor

### 1.3 External Subroutines

Unsigned multiply (.MPYU)

### 1.4 Other

None

## 2. OPERATING PROCEDURE

### 2.1 Calling Sequence

```
JSR  .MPY
return
```

### 2.2 Input Format

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

### 2.3 Output Format

The double precision result is returned in AC0 (high order) and AC1 (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

None

# 3. DISCUSSION

## 3.1 Algorithms

The signed multiply routine calls the unsigned multiply and uses a correction factor to adjust the final result without resorting to a determination of the algebraic sign of the result.

Let N1 and N2 be the operands. Four cases must be examined.

If $N1 \geqslant \emptyset$, and $N2 \geqslant \emptyset$, no correction is necessary.

If $N1 \geqslant \emptyset$ and $N2 < \emptyset$, then N2 as an unsigned number is 2**16 - abs(N2).

The result of an unsigned multiply is thus

    2**16*abs(N1) - abs(N1)*abs(N2).

The true result is negative and should be

    2**32 - abs(N1)*abs(N2).

This can be obtained by adding

    2**32 - 2**16*abs(N1)

to the unsigned multiply result. But since

2**32 - 2**16*abs(N1) = 2**16*(2**16 - abs(N1))

this is equivalent to subtracting N1 from the high order result of the unsigned multiply.

If $N1 < \emptyset$ and $N2 \geqslant \emptyset$, a similar analysis gives us a subtraction of N2 from the high order result as the correction.

If N1 < Ø and N2 < Ø, the unsigned product is:

((2**16 - abs(N1)) * (2**16 - abs(N2)) mod 2**32

= -2**16*abs(N1) - 2**16*abs(N2) + abs(N1)*abs(N2).

To obtain the true result of

abs(N1)*abs(N2)

we merely subtract both N1 and N2 from the high order of the unsigned result.

The correction factor is determined before the unsigned multiply is performed and subtracted from the high order word of the result to give the correct two's complement result.

## 3.2 Limitations and Accuracy

The routine is exact.

## 3.3 Size and Timing

The routine is 16 (octal) words in length and requires the unsigned multiply subroutine (.MPYU).

Average execution time is 56.4 $\mu$ seconds in addition to the time for the unsigned multiply. Average execution time for the unsigned multiply is 34Ø $\mu$ seconds. Total average execution time is thus 396.4 $\mu$ seconds.

## 3.4 References

See the unsigned multiply write-up, 093-000015.

## 3.5 Flow Diagrams

Not applicable.

## 4. EXAMPLES AND APPLICATIONS

An ASCII source tape (090-000013) of .MPY is provided with the standard NOVA software. This tape should be edited into the user software that requires the signed multiply routine.

## 5. PROGRAM LISTING

A listing of .MPY follows. No origin is given in the source, enabling the user to edit the routine anywhere into his program.

```
                ; MULTIPLY
                ; MULTIPLIES TWO FIXED POINT, SINGLE PRECISION,
                ;       TWO'S COMPLEMENT NUMBERS

                ; INPUT:         N1 IN AC1, N2 IN AC2

                ; OUTPUT:        N1*N2, HIGH ORDER IN AC0, LOW ORDER IN
                ;                AC1

                ; CALLING SEQUENCE:
                ;       JSR     .MPY
                ;       RETURN

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

                ; REQUIRES:      .MPYU (UNSIGNED MULTIPLY)

00000 054013 .MPY:    STA 3,.AB03       ; SAVE RETURN
00001 176400          SUB 3,3           ; AC3 WILL CONTAIN CORRECTION
                                        ; FACTOR
00002 125112          MOVL# 1,1,SZC     ; TEST SIGN OF N1
00003 157000          ADD 2,3           ; ADD N2 TO CORRECTION
00004 151112          MOVL# 2,2,SZC     ; TEST SIGN OF N2
00005 137000          ADD 1,3           ; ADD N1 TO CORRECTION
00006 054014          STA 3,.AB10       ; SAVE CORRECTION FACTOR
00007 006015          JSR @.AB30        ; CALL UNSIGNED MULTIPLY
00010 034014          LDA 3,.AB10       ; GET CORRECTION
00011 162400          SUB 3,0           ; TRUE RESULT IN 2'S COMPLEMENT
                                        ; FORM
00012 002013          JMP @.AB03        ; RETURN

00013 000000 .AB03:   0                 ; SAVE AC3

00014 000000 .AB10:   0                 ; STORAGE FOR CORRECTION FACTOR

00015 000015 .AB30:   .MPYU             ; UNSIGNED MULTIPLY ADDRESS
```