



DATA GENERAL
CORPORATION

Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

Single Precision Signed Divide

TAPES

ASCII source: 090-000014

ABSTRACT

This routine divides a double precision, two's complement, fixed point number by a single precision, two's complement, fixed point number. The result is a single precision, two's complement quotient and a single precision, two's complement remainder.

1. REQUIREMENTS

1.1 Memory

1K or larger alterable memory

1.2 Equipment

NOVA central processor

1.3 External Subroutines

Unsigned divide (.DIVU)

1.4 Other

None

2. OPERATING PROCEDURE

2.1 Calling Sequence

JSR .DIV
return

2.2 Input Format

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

2.3 Output Format

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

2.4 Error Returns

An error results if the magnitude of the quotient exceeds $2^{15}-1$. In this case, Carry will be set and no division takes place. If the division is correctly executed, Carry will be zero.

2.5 State of Active Registers upon Exit

AC \emptyset , AC1, AC3, and Carry are destroyed.
AC2 remains unchanged.

2.6 Cautions to User

None

3. DISCUSSION

3.1 Algorithms

The .DIV routine remembers the signs of the dividend and divisor and calls the unsigned divide routine using the absolute values of the operands. If either the unsigned divide routine indicates an error, or the quotient exceeds $2^{*}15-1$, Carry is set, the ACs restored, and control is returned. Otherwise, the signs of the results are determined from the operand signs. The remainder is the same sign as the dividend. The quotient is signed according to the algebraic rules for division.

3.2 Limitations and Accuracy

The routine is exact.

3.3 Size and Timing

The routine requires 45 (octal) words of storage in addition to the unsigned divide routine.

Average execution time is approximately 122 μ seconds in addition to the unsigned divide time. The average time for an unsigned divide is approximately 483 μ seconds for a total average execution time of 605 μ seconds.

3.4 References

See write-up 093-000016 for a description of the unsigned divide routine.

3.5 Flow Diagrams

None

4. EXAMPLES AND APPLICATIONS

None

5. PROGRAM LISTING

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

```

; SIGNED DIVIDE
; DIVIDES TWO FIXED POINT, TWO'S COMPLEMENT NUMBERS

; INPUT:          N1 IN AC0 AND AC1 (HIGH AND LOW), N2 IN
;                AC2

; OUTPUT:         N1/N2 ; REMAINDER IN AC0 (SAME SIGN AS
;                DIVIDEND)
;                QUOTIENT IN AC1

; CALLING SEQUENCE
;     JSR     .DIV
;     RETURN

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

; EXCEPTIONAL CONDITION:      IF THE MAGNITUDE OF THE
;                               QUOTIENT EXCEEDS
;                               2**15-1, CARRY IS SET AND
;                               THE DIVIDEND REMAINS UNCHANGED
;                               OTHERWISE, CARRY WILL BE 0

; REQUIRES:      .DIVU (UNSIGNED DIVIDE)

```

```

00000 054042 .DIV:   STA 3, .AC03.    ; SAVE RETURN
00001 050041      STA 2, .AC02    ; SAVE DIVISOR
00002 044040      STA 1, .AC01    ; SAVE DIVIDEND
00003 040037      STA 0, .AC00
00004 155102      MOVL 2, 3, SEC   ; CHECK SIGN OF DIVISOR
00005 150400      NEG 2, 2        ; FORM ABSOLUTE VALUE
00006 176560      SUBCL 3, 3      ; SAVE SIGN OF DIVISOR IN AC3
00007 175120      MOVZL 3, 3     ; POSITION IN BIT 14
00010 101113      MOVL# 0, 0, SNC ; TEST SIGN OF DIVIDEND
00011 000016      JMP .AC99      ; POSITIVE, BIT 15 OF AC3
                                ; CONTAINS SIGN OF DIVIDEND
                                ; SIGN OF DIVIDEND TO
                                ; AC3 BIT 15
00012 175400      INC 3, 3       ; FORM ABS. VALUE OF DIVIDEND

00013 124404      NEG 1, 1, S2R   ;
00014 100001      COM 0, 0, SKP   ;
00015 100400      NEG 0, 0       ;
00016 054043 .AC99: STA 3, .AC10   ; FLAG WORD FOR SIGNS OF
                                ; REMAINDER AND QUOTIENT
00017 006044      JSR 0, .AC30   ; CALL .DIVU (UNSIGNED DIVIDE)
00020 030041      LDA 2, .AC02   ; RESTORE AC2
00021 125103      MOVL 1, 1, SNC ; IF SIGN BIT SET,
                                ; QUOTIENT CAN'T
                                ; BE REPRESENTED IN 16 BITS
00022 125202      MOVR 1, 1, SEC ; IF CARRY SET, .DIVU GAVE
                                ; ERROR RETURN
00023 000034      JMP .AC98     ; ERROR, RETURN WITH CARRY SET
00024 034043      LDA 3, .AC10   ; GET FLAG WORD

```

```

;
; AC3 CONTAINS FOUR POSSIBLE COMBINATIONS
; THESE ARE: 00 0 POSITIVE, R POSITIVE
;            01 0 NEGATIVE, R NEGATIVE
;            10 0 NEGATIVE, R POSITIVE
;            11 0 POSITIVE, R NEGATIVE
;

00025 175203      MOVR 3,3,SNC      ; TEST REMAINDER SIGN
00026 174001      COM 3,3,SKP      ; POSITIVE
00027 100400      NEG 0,0          ; REMAINDER IS NEGATIVE
00030 175203      MOVR 3,3,SNC      ; TEST QUOTIENT SIGN
00031 124400      NEG 1,1          ; QUOTIENT IS NEGATIVE
00032 175020      MOVZ 3,3         ; CLEAR CARRY
00033 002042      JMP 0,AC03       ; RETURN

00034 020037 .AC98: LDA 0,.AC00    ; DIVIDE ERROR
00035 024040      LDA 1,.AC01      ; RESTORE DIVIDEND
00036 002042      JMP 0,AC03       ; RETURN, CARRY IS SET

00037 000000 .AC00: 0              ; SAVE DIVIDEND
00040 000000 .AC01: 0              ; SAVE DIVIDEND
00041 000000 .AC02: 0              ; SAVE AC2
00042 000000 .AC03: 0              ; SAVE AC3

00043 000000 .AC10: 0              ; SIGN OF QUOTIENT AND
; REMAINDER FLAG WORD

00044 000044 .AC30: .DIVU          ; UNSIGNED INTEGER DIVIDE
; ROUTINE ADDRESS

```