



DATA GENERAL
CORPORATION

Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

Unsigned Divide

TAPES

ASCII Source: 090-000021

ABSTRACT

This routine divides an unsigned, single precision divisor into an unsigned, double precision dividend to form a single precision quotient and a single precision remainder.

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

To divide a double precision dividend by a single precision divisor,

```
JSR .DIVU  
return
```

To divide a single precision dividend by a single precision divisor

```
JSR .DIVI  
return
```

2.2 Input Format

The dividend is passed in AC \emptyset (high order) and AC1 (low order). If .DIVI is called, the dividend is passed in AC1.

2.3 Output Format

The single precision quotient is returned in AC1. The remainder is returned in AC \emptyset .

2.4 Error Returns

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 \emptyset on return.

2.5 State of Active Registers upon Exit

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

2.6 Cautions to User

None

3. DISCUSSION

3.1 Algorithms

The subroutine initially compares the divisor to the high order of the dividend. If the divisor is less than or equal to the latter quantity, the result would be greater than $2^{16} - 1$ and could not be represented in 16 bits. In this case, Carry is set and return is made.

Otherwise, 16 iterations are performed. Each iteration determines whether the divisor will go into the most significant 16 bits of the dividend. If it will, the divisor is subtracted from the dividend and the dividend is shifted left one position. If it will not, no subtraction is performed - only the left shift. In both cases, Carry contains the latest quotient bit and is shifted in behind the low order of the dividend. Upon completion, the 16-bit quotient is entirely assembled in AC1, while the final adjusted dividend in AC \emptyset is the remainder.

3.2 Limitations and Accuracy

The routine is exact.

3.3 Size and Timing

The routine is 21 (octal) words in length.

Average execution time is 483 μ seconds.

3.4 References

Section 2.2 of "How to Use the NOVA" contains a further discussion of unsigned divide.

3.5 Flow Diagrams

None

4. EXAMPLES AND APPLICATIONS

The source tape of .DIVU is provided with the NOVA software. This tape can be directly edited into user programs that require unsigned divide.

This routine is called by a number of other programs in the Math Library.

5. PROGRAM LISTING

A listing of .DIVU follows. No origin is specified in the source, enabling the user to edit this subroutine anywhere within his program.

```

; UNSIGNED DIVIDE
; DIVIDES TWO UNSIGNED NUMBERS

; INPUT:          DIVIDEND IN AC0, AC1 ; HIGH ORDER, LOW
;                ORDER
;                DIVISOR IN AC2

; OUTPUT:         REMAINDER IN AC0
;                QUOTIENT IN AC1

; CALLING SEQUENCE
;     JSR     .DIVU
;     RETURN

; FOR INTEGER DIVIDE (SINGLE PRECISION DIVIDEND)
;     JSR     .DIVI
;     RETURN

; ERROR CONDITION:  AC0 >= AC2 ( INPLIES RESULT >
;                  2**16-1)
;                  SETS CARRY AND RETURNS
;                  OTHERWISE CARRY IS 0

; DESTROYED:      AC0, AC1 (AC1 UNCHANGED IF DIVIDE
;                  ERROR), AC3, CARRY
; UNCHANGED:      AC2

```

```

00000 102400 .DIVI:  SUB 0,0          ; INTEGER DIVIDE, CLEAR AC0
00001 054017 .DIVU:  STA 3,.CC03     ; SAVE AC3
00002 142432          SUBZ# 2,0,SEC  ; TEST FOR OVERFLOW
00003 000015          JMP .CC99      ; SET CARRY AND RETURN
00004 034020          LDA 3,.CC20    ; 16 ITERATIONS
00005 125120          MOVL 1,1      ; SHIFT LOW DIVIDEND
00006 101100 .CC98:  MOVL 0,0      ; SHIFT HIGH DIVIDEND
00007 142412          SUB# 2,0,SEC  ; DOES DIVISOR GO IN?
00010 142400          SUB 2,0      ; YES
00011 125100          MOVL 1,1      ; SHIFT LOW DIVIDEND
00012 175404          INC 3,3,SZR   ; CHECK COUNT
00013 000006          JMP .CC98     ; NOT DONE
00014 176441          SUBO 3,3,SKP  ; DONE , CLEAR CARRY
00015 176420 .CC99:  SUBZ 3,3      ; SET CARRY
00016 002017          JMP @.CC03    ; RETURN

00017 000000 .CC03:  0              ; SAVE AC3

00020 177760 .CC20:  -20           ; - 16 DECIMAL

```