



DATA GENERAL
CORPORATION

Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

Double Precision ASCII Decimal to Binary

TAPES

ASCII Source: 090-000035

ABSTRACT

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

1. REQUIREMENTS

1.1 Memory

1K or larger alterable memory.

1.2 Equipment

NOVA central processor.

1.3 External Subroutines

A user supplied "get a character" routine (see 2.2) and possibly a user supplied "accept a character" routine (see 2.1).

1.4 Other

None.

2. OPERATING PROCEDURE

2.1 Calling Sequence

Two entry points are provided. The first is a normal entry, while the second is for the user who needs an indication that a number is requested. The normal entry is called by

```
JSR .DDB  
return
```

If an indication is needed, entry is made by

```
JSR .DDBI  
return
```

If this second entry is used, the user must provide the address of a routine in location 41 of page zero that will accept the indication. Indication will be an ASCII "D" (for double precision integer) followed by a null character. These two characters are passed separately, right adjusted in ACØ (bit 8 = Ø), using two successive calls to the user. The user need not save any registers (except AC3 if used) or Carry. Return should be made by JMP Ø,3.

2.2 Input Format

Input characters will be requested by calling a user "get a character" routine. The address of this routine must be stored in location 4Ø of page zero. This user routine must be provided. This routine should return an ASCII character right adjusted in ACØ with bit 8 = Ø for each call. Input should be of the form:

SDD...DD(break)

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

2.3 Output Format

Upon exit, ACØ will contain the break character. AC1 and AC2 will contain the double precision, two's complement binary equivalent of the input. AC1 contains high order, AC2 low order.

2.4 Error Returns

None.

2.5 State of Active Registers upon Exit

All accumulators and Carry are destroyed.

2.6 Cautions to User

An indefinite stream of decimal digits will be accepted as input. However, the result will be

$$N \bmod 2^{**31}$$

where N represents the input number. For example, 2,147,483,655 ($2^{**31} + 7$) converts to 7.

3. DISCUSSION

3.1 Algorithms

The sign is processed (if given) and a flag set for later use in determining whether to negate the magnitude of the result. A character is requested and checked for a decimal digit. If not, the routine terminates with the character in ACØ as the break. Otherwise, the ASCII digit is reduced to the binary range

$$Ø \leq D \leq 9$$

and is added to a running Horner's sum (initially zero). If S_i is the former sum, this procedure is

$$S_{i+1} \leftarrow S_i * 1Ø + D$$

where S_{i+1} becomes the new sum. For example if six digits, D(5) to D(Ø), are returned before the break, the result is

$$((((D(5)*1Ø+D(4))*1Ø+D(3))*1Ø+D(2))*1Ø+D(1))*1Ø+D(Ø).$$

3.2 Limitations and Accuracy

The result is

$$N \text{ mod } 2^{**}31$$

where N represents the decimal input.

3.3 Size and Timing

The routine is 77 (octal) words in length.

Execution time is approximately

$$124.7 + I * 125.5 \mu \text{seconds}$$

where I is the number of decimal digits in the input. For example, -147,936 requires

$$124.7 + 6 * 125.5 = .877 \text{ milliseconds.}$$

3.4 References

Write-up 093-000025 describes single precision decimal to binary.

3.5 Flow Diagrams

None.

4. EXAMPLES AND APPLICATIONS

The ASCII source of double precision decimal to binary is provided with the NOVA software. If a user requires this program, the tape should be edited into the user source.

5. PROGRAM LISTING

A listing of .DDB follows. No origin is given in the source, enabling the user to edit the tape anywhere within his routine.

```

; CONVERT AN ASCII CHARACTER STRING TO A DOUBLE
;     PRECISION BINARY NUMBER
;
; 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 OF 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
;           AC1, AC2 CONTAIN THE BINARY RESULTS
;           OF THE CONVERSION
;
; CALLING SEQUENCE:
;           JSR     .DDB
;           RETURN
;
; IF AN INDICATION IS DESIRED TO SIGNAL CHARACTERS
; ARE REQUESTED, CALLING SEQUENCE IS:
;           JSR     .DDBI
;           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
;
; CAUTION:   THE ABSOLUTE VALUE OF THE RESULT
;           IS N MOD 2**31.
;
; DESTROYED: ALL ACS AND CARRY

```

```

00000 054066 .DDBI: STA 3,.FC03      ; SAVE RETURN
00001 020072      LDA 0,.FC20      ; SEND "D"
00002 006041      JSR @.FC41
00003 102400      SUB 0,0
00004 006041      JSR @.FC41      ; SEND NULL
00005 000007      JMP  .+2

00006 054066 .DDB:   STA 3,.FC03      ; SAVE RETURN
00007 102400      SUB 0,0
00010 040067      STA 0,.FC10      ; CLEAR SIGN WORD
00011 040070      STA 0,.FC11      ; CLEAR RESULT WORDS
00012 040071      STA 0,.FC11+1

```

```

00013 006040      JSR 0,FC40      ; GET A CHARACTER
00014 024073      LDA 1,.FC21     ; TEST FOR "+"
00015 106405      SUB 0,1,SNR
00016 000023      JMP .FC99      ; YES
00017 024074      LDA 1,.FC22     ; TEST FOR "-"
00020 106404      SUB 0,1,SZR
00021 000024      JMP .FC98      ; NEITHER
00022 010067      ISZ .FC10     ; SET SIGN FLAG TO 1 FOR MINUS
00023 006040      .FC99: JSR 0,FC40     ; GET A DIGIT
00024 024075      .FC98: LDA 1,.FC23     ; TEST ASCII "0"
00025 030076      LDA 2,.FC24     ; ASCII "9"
00026 142033      ADCZ# 2,0,SNC  ; TEST FOR A DIGIT
00027 106032      ADCZ# 0,1,SZC
00030 000054      JMP .FC97      ; NOT A DIGIT - THEREFORE A
                                ; BREAK
                                ; REDUCE TO 0-9 RANGE
00031 122400      SUB 1,0
00032 024070      LDA 1,.FC11     ; GET SUM WORDS
00033 030071      LDA 2,.FC11+1
00034 151120      MOVZL 2,2      ; SUM*4
00035 125100      MOVL 1,1
00036 151120      MOVZL 2,2
00037 125100      MOVL 1,1
00040 034071      LDA 3,.FC11+1
00041 173022      ADDZ 3,2,SZC
00042 125400      INC 1,1
00043 034070      LDA 3,.FC11
00044 167000      ADD 3,1      ; SUM*(4+1)
00045 151120      MOVZL 2,2
00046 125100      MOVL 1,1      ; 2*SUM*(4+1)
00047 113022      ADDZ 0,2,SZC  ; SUM*10+AC0
00050 125400      INC 1,1
00051 044070      STA 1,.FC11
00052 050071      STA 2,.FC11+1  ; STORE RESULTS

00053 000023      JMP .FC99      ; GET NEXT DIGIT

00054 024070      .FC97: LDA 1,.FC11     ; GET RESULTS
00055 030071      LDA 2,.FC11+1
00056 125120      MOVZL 1,1
00057 125220      MOVZL 1,1      ; SET SIGN +
00060 014067      DSZ .FC10     ; TEST SIGN OF RESULT
00061 002066      JMP 0,FC03     ; POSITIVE, RETURN
00062 150404      NEG 2,2,SZR      ; NEGATIVE
00063 124001      COM 1,1,SKP
00064 124400      NEG 1,1
00065 002066      JMP 0,FC03     ; RETURN

```

???

```
00066 000000 .FC03: 0 ; SAVE RETURN

00067 000000 .FC10: 0 ; SIGN OF RESULT WORD
      000002 .FC11: .BLK 2 ; RESULT WORDS

00072 000104 .FC20: "D ; SEND "D" FOR INDICATION
00073 000053 .FC21: "+" ; ASCII "+"
00074 000055 .FC22: "-" ; ASCII "-"
00075 000060 .FC23: "0" ; ASCII "0"
00076 000071 .FC24: "9" ; ASCII "9"

      000040 .FC40=40 ; PAGE 0 GET CHARACTER ADDRESS
      000041 .FC41=41 ; PAGE 0 PUT CHARACTER ADDRESS
```