



DATA GENERAL
CORPORATION

Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

Single Precision ASCII Decimal to Binary

TAPES

ASCII Source: 090-000029

ABSTRACT

This routine converts an ASCII character string of decimal digits to the single precision binary equivalent of the input.

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 "put character" routine (see 2.1) are required.

1.4 Other

None

2. OPERATING PROCEDURE

2.1 Calling Sequence

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

```
JSR .DBIN  
return
```

If an indication is needed, entry is made by

```
JSR .DBNI  
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 "S" (for single 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 routine. The user routine need not save any registers (except AC3 if used). 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 whose address must be stored in location 4Ø of page zero. This user routine must be provided.

ASCII characters should be returned, right adjusted in ACØ with bit 8 = Ø. This routine need not save any registers or Carry. Input should be in the form:

S D D . . . D D (break)

where S represents the sign ("- or optionally "+"), 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 ASCII break character and AC1 will contain the single precision, two's complement binary equivalent of the input.

2.4 Error Returns

None

2.5 State of Active Registers upon Exit

AC2 is unchanged. ACØ, AC1, AC3, 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

$$N \text{ mod } 2^{**}15$$

where N represents the input number. For example, 32769 converts to +1, -32768 converts to \emptyset , 96741 converts to 31205.

3. DISCUSSION

3.1 Algorithms

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

$$\emptyset \leq D \leq 9$$

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

$$S_{i+1} \leftarrow S_i * 1\emptyset + D$$

For example, if three digits, $D(2)$, $D(1)$, $D(\emptyset)$, are returned before a break, the result is

$$(D (2) * 1\emptyset + D (1)) * 1\emptyset + D (\emptyset) .$$

3.2 Limitations and Accuracy

The result is

$$N \text{ mod } 2^{**}15$$

where N represents the decimal input.

3.3 Size and Timing

The routine is 65 (octal) words in length.

Execution time is approximately

$$110 + I * 82.2 \mu \text{ seconds}$$

where I is the number of digits in the input.

For example, +5768 requires

$$110 + 4 * 82.2 = 438.8 \mu \text{ seconds.}$$

3.4 References

None

3.5 Flow Diagrams

None

4. EXAMPLES AND APPLICATIONS

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

5. PROGRAM LISTING

A listing of decimal to binary follows. No origin is given in the source, enabling the user to edit this tape anywhere within his routines.

```

; CONVERT AN ASCII CHARACTER STRING TO A SINGLE
; PRECISION BINARY NUMBER
; CONVERTS AN ASCII DECIMAL CHARACTER STRING TO A
; TWO'S COMPLEMENT, FIXED POINT, 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...D(BREAK)
;                 S IS THE SIGN, D A DECIMAL DIGIT
;                 THE BREAK CHARACTER IS ANY CHARACTER
;                 OTHER THAN A DIGIT.

; OUTPUT:         AC0 CONTAINS THE BREAK CHARACTER
;                 AC1 CONTAINS THE BINARY INTEGER

; CALLING SEQUENCE:
;     JSR     .DBIN
;     RETURN

; IF AN INDICATION IS DESIRED TO SIGNAL CHARACTERS ARE
; REQUESTED, CALLING SEQUENCE IS:
;     JSR     .DBNI
;     RETURN

; AN ASCII "S" 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**15.
;                 FOR EXAMPLE, +96741 CONVERTS TO +31205
;                 -2**15 CONVERTS TO 0

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

```

```

00000 054055 .DBNI: STA 3,.EC03 ; SAVE RETURN
00001 050054 STA 2,.EC02 ; SAVE AC2
00002 020064 LDA 0,.EC24 ; GET "S"
00003 006041 JSR 0,.EC41 ; SEND "S"
00004 102400 SUB 0,0
00005 006041 JSR 0,.EC41 ; SEND NULL
00006 000011 JMP .+3
00007 054055 .DBIN: STA 3,.EC03 ; SAVE AC3
00010 050054 STA 2,.EC02 ; SAVE AC2
00011 102400 SUB 0,0
00012 040056 STA 0,.EC10 ; CLEAR SIGN WORD
00013 040057 STA 0,.EC11 ; CLEAR SUM WORD
00014 006040 JSR 0,.EC40 ; GET A CHARACTER
00015 024060 LDA 1,.EC20 ; TEST FOR "+"
00016 106405 SUB 0,1,SNR
00017 000024 JMP .EC97 ; YES
00020 024061 LDA 1,.EC21 ; NO, TEST FOR "--"
00021 106404 SUB 0,1,SZR
00022 000025 JMP .EC96 ; NO EXPLICIT SIGN
00023 010056 ISZ .EC10 ; SET FLAG WORD FOR NEGATIVE
; NUMBER
00024 006040 .EC97: JSR 0,.EC40 ; GET ANOTHER CHARACTER
00025 024062 .EC96: LDA 1,.EC22 ; ASCII "0"
00026 030063 LDA 2,.EC23 ; ASCII "9"
00027 142033 ADCZ# 2,0,SNR ; SKIP IF > 9
00030 106032 ADCZ# 0,1,SZC ; SKIP IF >= 0
00031 000037 JMP .EC95 ; NOT A DIGIT, THEREFORE A BREAK
; CHARACTER
00032 122400 SUB 1,0 ; REDUCE DIGIT TO 0-9 BINARY
; RANGE
00033 024057 LDA 1,.EC11 ; SUM WORD
00034 004046 JSR .EC50 ; MULTIPLY BY 10 AND ADD
00035 044057 STA 1,.EC11 ; SAVE SUM
00036 000024 JMP .EC97 ; GET NEXT CHARACTER

00037 024057 .EC95: LDA 1,.EC11 ; RESULT TO AC1
00040 125120 MOVZL 1,1
00041 014056 DSZ .EC10 ; TEST SIGN
00042 125221 MOVZR 1,1,SKP ; POSITIVE
00043 124640 NEGOR 1,1 ; NEGATIVE
00044 030054 LDA 2,.EC02 ; RESTORE AC2
00045 002055 JMP 0,.EC03

```

↑↑↑

; ROUTINE TO MULTIPLY AC1 BY 10 AND ADD AC0

```
00046 131120 .EC50:  MOVZL 1,2      ; N*2
00047 151120      MOVZL 2,2      ; N*4
00050 147000      ADD 2,1        ; N*5
00051 125120      MOVZL 1,1      ; N*5*2 = N*10
00052 107000      ADD 0,1        ; ADD AC0
00053 001400      JMP 0,3        ; SUCCESS RETURN

00054 000000 .EC02:  0          ; SAVE AC2
00055 000000 .EC03:  0          ; SAVE AC3

00056 000000 .EC10:  0          ; FLAG WORD FOR SIGN OF RESULT
00057 000000 .EC11:  0          ; RUNNING SUM WORD

00060 000053 .EC20:  "+"        ; ASCII "+"
00061 000055 .EC21:  "-"        ; ASCII "-"
00062 000060 .EC22:  "0"        ; ASCII "0"
00063 000071 .EC23:  "9"        ; ASCII "9"
00064 000123 .EC24:  "S"        ; ASCII "S" FOR INDICATION
                                ; ENTRY

000040 .EC40=40      ; ADDRESS OF GET CHARACTER
                                ; ROUTINE
000041 .EC41=41      ; ADDRESS OF PUT CHARACTER
                                ; ROUTINE
```