



DATA GENERAL
CORPORATION

Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

ASCII Octal to Single Precision Binary

TAPES

ASCII Source: 090-000031

ABSTRACT

This routine converts an ASCII character string consisting of octal digits to a single precision binary 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 REQUIREMENTS

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. Normal entry is made by

```
JSR .OBIN  
return
```

If an indication is needed, entry is made by

```
JSR .OBNI  
return
```

If the 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 "0" (for octal) followed by a null character. These two characters are passed separately, right adjusted in AC \emptyset (bit 8 = \emptyset), using two consecutive calls to the user routine. The user routine need not save any registers (except AC3 if used) and return should be made by JMP $\emptyset, 3$.

2.2 Input Format

Input characters will be requested by calling a user "get a character" routine whose address must be stored in location 40 of page zero. This user routine must be provided. Upon call, this routine should return as ASCII character, right adjusted in AC0 with bit 8 = 0. This routine need not save any registers (except AC3 if used) or Carry. Return should be made by JMP 0,3.

Input should be of the form:

00...00(break)

where "0" represents an ASCII octal digit ($0 \leq 0 \leq 67$), and "break" is any ASCII character not in this range.

2.3 Output Format

Upon exit, AC0 will contain the ASCII break character and AC1 will contain the 16-bit binary equivalent of the input.

2.4 Error Returns

None.

2.5 State of Active Registers upon Exit

AC2 is unchanged. AC0, AC1, AC3, and Carry are destroyed.

2.6 Cautions to User

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

$N \bmod 20000$ (octal)

For example, 576452 converts to 176452.

3. DISCUSSION

3.1 Algorithms

A register for the result is initially cleared. An ASCII character is requested and checked for the range

$$6\emptyset \leq o \leq 67.$$

If out of range, the algorithm terminates. Otherwise, the character is reduced to the binary range

$$\emptyset \leq o \leq 7,$$

the result register is shifted three binary positions left, and the digit is added to the result. This procedure is repeated until a break character is encountered.

3.2 Limitations and Accuracy

The result is

$$N \bmod 2\emptyset\emptyset\emptyset\emptyset\emptyset \text{ (octal)}$$

where N represents the octal input.

3.3 Size and Timing

The routine is 42 (octal) words in length.

Execution time for .OBIN is

$$63.\emptyset + I * 7\emptyset.2 \mu \text{ seconds}$$

where I represents the number of digits in the input. For example, 576452 requires

$$63.\emptyset + 6 * 7\emptyset.2 = 484.2 \mu \text{ seconds.}$$

3.4 References

None.

3.5 Flow Diagrams

None.

4. EXAMPLES AND APPLICATIONS

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

5. PROGRAM LISTING

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

```

; CONVERT AN ASCII OCTAL CHARACTER STRING TO A 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

;                INPUT OF FORM:
;                00...0(BREAK)
;                WHERE "0" REPRESENTS AN OCTAL DIGIT AND
;                BREAK IS ANY OTHER CHARACTER

; OUTPUT:         AC0 CONTAINS THE BREAK CHARACTER
;                AC1 CONTAINS THE BINARY NUMBER (MOD
;                200000 OCTAL)

; CALLING SEQUENCE:
;     JSR     *OBIN
;     RETURN

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

; AN ASCII "0" FOLLOWED BY A NULL CHARACTER
; WILL BE TRANSMITTED VIA AC0
; TO USER PUT CHARACTER ROUTINE WHOSE
; ADDRESS MUST BE STORED IN LOCATION 41 OF PAGE 0

; CAUTION:       RESULT IS N MOD 200000 (OCTAL)
;                E.G. 576452* CONVERTS TO 176452

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

```

```

00000 054035 *OBNI:  STA 3,.EE03      ; SAVE RETURN
00001 050034          STA 2,.EE02      ; SAVE AC2
00002 020041          LDA 0,.EE22
00003 006041          JSR 0,.EE41      ; SEND "0"
00004 102400          SUB 0,0
00005 006041          JSR 0,.EE41      ; SEND NULL
00006 000011          JMP  +3
00007 054035 *OBIN:  STA 3,.EE03      ; SAVE RETURN
00010 050034          STA 2,.EE02      ; SAVE AC2
00011 126400          SUB 1,1         ; CLEAR RESULT WORD
00012 044036          STA 1,.EE10

```

↑↑↑

```
00013 006040 .EE98: JSR @.EE40 ; GET A DIGIT
00014 030037 LDA 2,.EE20 ; OCTAL 60
00015 034040 LDA 3,.EE21 ; OCTAL 67
00016 162033 ADCZ# 3,0,SNC ; TEST FOR 60 <=N<= 67
00017 112032 ADCZ# 0,2,SZC
00020 000031 JMP .EE99 ; NO - MUST BE BREAK CHARACTER
00021 142400 SUB 2,0 ; PUT N IN RANGE 0-7
00022 024036 LDA 1,.EE10
00023 125120 MOVZL 1,1 ; SHIFT SUM
00024 125120 MOVZL 1,1
00025 125120 MOVZL 1,1
00026 107000 ADD 0,1
00027 044036 STA 1,.EE10
00030 000013 JMP .EE98 ; LOOP TILL BREAK RECEIVED

00031 030034 .EE99: LDA 2,.EE02 ; RESTORE AC2
00032 024036 LDA 1,.EE10 ; ANSWER TO AC1
00033 002035 JMP @.EE03 ; AND RETURN

00034 000000 .EE02: 0 ; SAVE AC2
00035 000000 .EE03: 0 ; SAVE RETURN

00036 000000 .EE10: 0 ; STORAGE FOR RESULTS

00037 000060 .EE20: 60 ; ASCII "0"
00040 000067 .EE21: 67 ; ASCII "7"
00041 000117 .EE22: "0" ; ASCII "0"

000040 .EE40=40 ; PAGE 0 ADDRESS OF GET A
; CHARRACTER ROUTINE
000041 .EE41=41 ; PAGE 0 ADDRESS OF PUT A
; CHARACTER ROUTINE
```