DATA GENERAL
CORPORATION
Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

Single Precision BCD to Binary

TAPES

ASCII Source: 090-000027

ABSTRACT

This routine converts a four digit (16-bit) number in BCD to its binary equivalent.

093-000023-00

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

```
JSR  .BCDB
return
```

2.2 Input Format

A four digit BCD integer is passed AC1, most significant digit in bits Ø-3. Its maximum value is 9999.

2.3 Output Format

The binary equivalent (always positive) is returned in AC1.

2.4 Error Returns

If a digit greater than binary 1ØØ1 is encountered in the input, Carry will be set, AC1 will be unchanged, and ACØ will contain the bad digit.

3.5 Flow Diagrams

None

4. EXAMPLES AND APPLICATIONS

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

5. PROGRAM LISTING

A listing of .BCDB follows. No origin is given in the source, enabling the tape to be edited anywhere within a user routine.

```
                ; CONVERT A NUMBER IN BCD TO BINARY

                ; INPUT:          A BCD INTEGER IN AC1 (MAXIMUM VALUE
                ;                 9999 DECIMAL)

                ; OUTPUT:         BINARY EQUIVALENT OF BCD INPUT IN AC1

                ; CALLING SEQUENCE:
                ;       JSR     .BCDB
                ;       RETURN

                ; EXCEPTIONAL CONDITION:          IF A DIGIT >9 IS
                ;                                 ENCOUNTERED IN THE INPUT,
                ;                                 CARRY WILL BE SET, AC1 WILL
                ;                                 BE UNCHANGED, AND AC0 WILL
                ;                                 CONTAIN THE BAD DIGIT

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

00000 044043  .BCDB:  STA 1,.EA01       ; SAVE INPUT
00001 044047          STA 1,.EA11       ; SAVE IN CURRENT INPUT WORD
00002 050044          STA 2,.EA02       ; SAVE AC2
00003 054045          STA 3,.EA03       ; SAVE RETURN
00004 020051          LDA 0,.EA20       ; COUNT IS 4
00005 040050          STA 0,.EA12       ; SAVE IN LOOP COUNT WORD
00006 152400          SUB 2,2           ; CLEAR RESULT REGISTER

00007 004020  .EA98:  JSR .EA50         ; GET DIGIT
00010 000015          JMP .EA99         ; DIGIT IN ERROR, RETURN
00011 004035          JSR .EA51         ; MULTIPLY SUM BY 10 AND ADD
                                        ; DIGIT
00012 010050          ISZ .EA12         ; DONE YET?
00013 000007          JMP .EA98         ; NO
00014 145021          MOVZ 2,1,SKP      ; YES, RESULT TO AC1, CLEAR
                                        ; CARRY
00015 024043  .EA99:  LDA 1,.EA01       ; RESTORE INPUT IF ERROR
00016 030044          LDA 2,.EA02       ; RESTORE AC2
00017 002045          JMP @.EA03        ; RETURN
```

```
                ; GET NEXT DIGIT OF INPUT, CHECK FOR LEGAL BCD
                ; CALLING SEQUENCE:
                ;       JSR     .EA50
                ;       ERROR RETURN - BAD DIGIT IN AC0
                ;       SUCCESS RETURN - DIGIT IN AC0

00020 024051 .EA50:  LDA 1,.EA20      ; GET -4
00021 044046         STA 1,.EA10      ; SAVE IN COUNT WORD
00022 024047         LDA 1,.EA11      ; CURRENT INPUT WORD
00023 102400         SUB 0,0          ; CLEAR AC0
00024 125120         MOVZL 1,1
00025 101100         MOVL 0,0         ; BIT TO AC0
00026 010046         ISZ .EA10
00027 000024         JMP .-3          ; LOOP TILL A DIGIT SHIFTED
00030 044047         STA 1,.EA11      ; SAVE CURRENT INPUT WORD
00031 024052         LDA 1,.EA21      ; TEST CONSTANT (10 DECIMAL)
00032 106042         ADCO 0,1,SZC
00033 001400         JMP 0,3          ; ERROR: >9, CARRY IS SET
00034 001401         JMP 1,3          ; SUCCESS: <=9

                ; MULTIPLY AC2 BY 10 AND ADD AC0

00035 145120 .EA51:  MOVZL 2,1        ; N*2
00036 125120         MOVZL 1,1        ; N*4
00037 147000         ADD 2,1          ; N*5
00040 131120         MOVZL 1,2        ; N*10
00041 113000         ADD 0,2          ; N*10+AC0
00042 001400         JMP 0,3


00043 000000 .EA01:  0                ; SAVE INPUT
00044 000000 .EA02:  0                ; SAVE AC2
00045 000000 .EA03:  0                ; SAVE RETURN

00046 000000 .EA10:  0                ; LOOP COUNT FOR SHIFT
00047 000000 .EA11:  0                ; CURRENT INPUT
00050 000000 .EA12:  0                ; LOOP COUNT FOR MULTIPLY AND
                                      ; ADD

00051 177774 .EA20:  -4               ; COUNT
00052 000012 .EA21:  12               ; BCD ERROR TEST, DECIMAL 10
```