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

PROGRAM

Double Precision Binary to BCD

TAPES

ASCII Source: 090-000034

ABSTRACT

This routine converts a double precision binary number to its eight digit BCD equivalent.

 093-000030-00

1. REQUIREMENTS

1.1 Memory

1K or larger alterable memory.

1.2 Equipment

NOVA central processor.

1.3 External Subroutines

Unsigned divide (.DIVU) is required.

1.4 Other

None.

2. OPERATING PROCEDURE

2.1 Calling Sequence

```
JSR  .DBBC
return
```

2.2 Input Format

A positive, double precision binary number in ACØ (high order), AC1 (low order).

2.3 Output Format

The eight digit (32-bit) BCD equivalent in ACØ (high order), AC1 (low order).

2.4 Error Returns

If a number greater than 99,999,999 is input for conversion, Carry is set to indicate an error (only eight digits of BCD information can be accomodated in 32 bits). If the conversion is performed, Carry will be zero on return.

2.5 State of Active Registers upon Exit

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

2.6 Cautions to User

None.

3. DISCUSSION

3.1 Algorithms

The binary input is compared to 1ØØ,ØØØ,ØØØ. If greater than or equal to, Carry is set to indicate an error and return is made. Otherwise, consider the input as

$$B = H * 10,000 + L$$

where H and L are integers in the range

$$0 \leq \begin{Bmatrix} H \\ L \end{Bmatrix} \leq 9,999.$$

Since only four BCD digits can be accomodated in a 16-bit register, H and L, represented in BCD, would be precisely the results desired. Further, H and L satisfy the algebraic definition of division of B by 1Ø,ØØØ, with H = quotient and L = remainder. The algorithm thus consists of a division of the input by 1Ø,ØØØ and single precision binary to BCD conversions on the quotient (to produce the high order result) and the remainder (to produce the low order result). The single precision conversion is performed by successively subtracting appropriate powers of ten from the binary input (see References).

3.2 Limitations and Accuracy

The routine is exact for all binary values less than 1ØØ,ØØØ,ØØØ decimal.

3.3 Size and Timing

The routine is 57 (octal) words in length.

Average execution time (including the divide) is

$$1018 + N * 14.1 \ \mu \text{ seconds}$$

where N is the sum of the digits of the result. For example, the worst case is 99,999,999 which gives an execution time of

$$1018 + 9 * 8 * 14.1 = 2.033 \text{ milliseconds.}$$

3.4 References

A description of the single precision binary to BCD conversion algorithm is described by write-up 093-000024.

3.5 Flow Diagrams

None.

4. EXAMPLES AND APPLICATIONS

The ASCII source of .DBCB is provided with the NOVA software. This tape should be edited into the user source that requires this type conversion.

5. PROGRAM LISTING

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

```
                          ; CONVERT A DOUBLE PRECISION BINARY NUMBER
                          ;         TO A DOUBLE PRECISION BCD NUMBER

                          ; INPUT:          AN UNSIGNED DOUBLE PRECISION BINARY
                          ;                 NUMBER IN ACØ, AC1 (HIGH, LOW)

                          ; OUTPUT:         THE BCD EQUIVALENT IN ACØ, AC1 (HIGH,
                          ;                 LOW)

                          ; CALLING SEQUENCE:
                          ;         JSR     .DBBC
                          ;         RETURN

                          ; ERROR CONDITION:        IF ACØ, AC1 CONTAIN A NUMBER
                          ;                 GREATER THAN 99999999, NO CONVERSION
                          ;                 WILL TAKE PLACE AND CARRY WILL BE SET

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

                          ; REQUIRES:       .DIVU (UNSIGNED DIVIDE)

ØØØØØ Ø54Ø42 .DBBC:  STA 3,.FBØ3        ; SAVE RETURN
ØØØØ1 Ø3ØØ52         LDA 2,.FB2Ø        ; GET 1ØØ,ØØØ,ØØØ
ØØØØ2 Ø34Ø53         LDA 3,.FB2Ø+1
ØØØØ3 112Ø32         ADCZ# Ø,2,SZC
ØØØØ4 ØØØØ1Ø         JMP .FB99          ; INPUT OK
ØØØØ5 112445         SUBO Ø,2,SNR
ØØØØ6 136Ø42         ADCO 1,3,SZC
ØØØØ7 ØØ2Ø42         JMP @.FBØ3         ; ERROR CARRY IS SET

ØØØ1Ø Ø3ØØ54 .FB99:  LDA 2,.FB21
ØØØ11 ØØ6Ø56         JSR @.FB31         ; DIVIDE BY 1Ø,ØØØ
ØØØ12 Ø4ØØ47         STA Ø,.FB1Ø        ; SAVE REMAINDER
ØØØ13 ØØ4Ø22         JSR .FB5Ø          ; CONVERT HIGH ORDER
ØØØ14 Ø4ØØ5Ø         STA Ø,.FB11        ; SAVE RESULTS HIGH
ØØØ15 Ø24Ø47         LDA 1,.FB1Ø
ØØØ16 ØØ4Ø22         JSR .FB5Ø          ; CONVERT LOW ORDER
ØØØ17 1Ø5Ø2Ø         MOVZ Ø,1
ØØØ2Ø Ø2ØØ5Ø         LDA Ø,.FB11        ; RESULTS TO ACØ, AC1
ØØØ21 ØØ2Ø42         JMP @.FBØ3
```

; CONVERT BINARY IN AC1 TO BCD IN AC0

```
00022 030055 .FB50:   LDA 2,.FB30       ; ADDRESS OF POWER OF TEN TABLE
00023 050051          STA 2,.FB12       ; SAVE IT
00024 102400          SUB 0,0           ; CLEAR AC0

00025 032051 .FB52:   LDA 2,@.FB12      ; GET POWER OF TEN
00026 010051          ISZ .FB12         ; BUMP TO NEXT ENTRY
00027 101120          MOVZL 0,0
00030 101120          MOVZL 0,0
00031 101120          MOVZL 0,0
00032 101120          MOVZL 0,0
00033 146422 .FB51:   SUBZ 2,1,SZC      ; DOES POWER OF 10 GO IN?
00034 101401          INC 0,0,SKP       ; YES, BUMP BDC OUTPUT
00035 147001          ADD 2,1,SKP       ; NO, RESTORE AC1
00036 000033          JMP .FB51         ; SUBTRACT AGAIN
00037 151224          MOVZR 2,2,SZR     ; DONE IF 10**0
00040 000025          JMP .FB52         ; NO, CONTINUE
00041 001400          JMP 0,3           ; YES, RETURN



00042 000000 .FB03:   0                 ; SAVE RETURN

      000012          .RDX 10
00043 001750 .FB05:   1000              ; 10**3
00044 000144          100               ; 10**2
00045 000012          10                ; 10**1
00046 000001          1                 ; 10**0
      000010          .RDX 8

00047 000000 .FB10:   0                 ; SAVE REMAINDER
00050 000000 .FB11:   0                 ; SAVE HIGH ORDER RESULTS
00051 000000 .FB12:   0                 ; POINTER TO CURRENT POWER OF
                                        ; TEN

00052 002765 .FB20:   2765              ; 10**9 (ERROR TEST)
00053 160400          160400
00054 023420 .FB21:   23420             ; 10**4

00055 000043 .FB30:   .FB05             ; ADDRESS OF POWER OF TEN TABLE
00056 000057 .FB31:   .DIVU             ; UNSIGNED DIVIDE REQUIRED
```