1.   IDENTIFICATION

1.1   Digital-8-11-U

1.2   Double Precision Binary Coded Decimal to Binary

1.3   March 18, 1966

2. ABSTRACT

This subroutine converts a 6-digit BCD number to its equivalent binary value in two computer words.

3. REQUIREMENTS

3.1 Storage

This subroutine requires 89 (decimal) memory locations.

3.2 Subprograms and/or Subroutines (None)

3.3 Equipment

Standard PDP-8

4. USAGE

4.1 Loading

The subroutine is loaded with the Binary Loader. The symbolic is either assembled with the user program or separately with the proper origin setting.

4.2 Calling Sequence

This subroutine is called with an effective JMS DOUBLE followed by the address of the high-order word of the double-precision BCD number. Control is returned to the following location with the high-order part of the result in C(AC) and with the low-order part of the result in C(LOW).

4.3 Switch Settings (None)

5. RESTRICTIONS (Not Applicable)

6. DESCRIPTION

6.1 Discussion

Upon entry, the BCD number is in the form:

$$(16^2 D_1 + 16 D_2 + D_3); (16^2 D_4 + 16 D_5 + D_6)$$

(each digit is 4 bits, $2^4 = 16$)

Using the single precision BCD to binary subroutine, then is reduced to

$$(10^2 D_1 + 10 D_2 + D_3); (10^2 D_4 + 10 D_5 + D_6)$$

The high order part of the BCD word is effectively multiplied by 1000 (= 8(128 − 3)) and the low-order part is added, giving

$$10^5 D_1 + 10^4 D_2 + 10^3 D_3 + 10^2 D_4 + 10 D_5 + D_6.$$

See Digital-8-10-U.

6.2 Examples

```
GO,     JMS I X
        HIGH
        HLT
X,      DOUBLE
HIGH,   1001    1001    1001    1    999,999
LOW,    1001    1001    1001
```

If this program were started at GO, the C(AC) at the halt would be 0364_8 and C(LOW) would be 1077_8, i.e., $03641077_8 = 999,999_{10}$.

7.   METHOD (Not Applicable)

8.   FORMAT (Not Applicable

9.   EXECUTION TIME

9.1   Minimum

        252 μsec

9.2   Maximum

        255 μsec

10.   PROGRAM

10.1   Core Map (Not Applicable)

10.2   Dimension List(s) (Not Applicable)

10.3   Macro, Parameter, and Variable Lists (None)

10.4   Program Listing

```
          /DIGITAL   8-11-U-SYM

          /DOUBLE PRECISION BCD TO BINARY CONVERSION
          /CALLING SEQUENCE:
          /        JMS DOUBLE
          /        ADDRESS OF HIGH ORDER ARGUMENT
          /        RETURN:    C(AC)=HIGH ORDER PART
          /                   C(LOW) = LOW ORDER PART

          /ALSO CONTAINS SINGLE PRECISION BCD TO BINARY
          /CALLING SEQUENCE:
          /        C(AC) = 3 BCD CHARACTERS
          /        JMS BCDBIN
          /        RETURN:    ANSWER IN C(AC)

0200 0000 DOUBLE,  0
0201 7300          CLA CLL
0202 1600          TAD I DOUBLE        / FETCH ADDRESS
0203 3271          DCA LOW1            / STORE
0204 2200          ISZ DOUBLE          / INCREMENT RETURN
0205 1671          TAD I LOW1 /FETCH HIGH ORDER
0206 4275          JMS BCDBIN /CONVERT IT
0207 3272          DCA HIGH1           / STORE
0210 2271          ISZ LOW1            / INCREMENT POINTER
0211 1671          TAD I LOW1 /FETCH LOW ORDER
0212 4275          JMS BCDBIN /CONVERT IT
0213 3271          DCA LOW1            / STORE IT
0214 1272          TAD HIGH1
0215 7112          CLL RTR
0216 7012          RTR
0217 7010          RAR                 /MULTIPLY HIGH ORDER
0220 3275          DCA BCDBIN /PART BY 128
0221 1275          TAD BCDBIN
0222 0327          AND K177
0223 3274          DCA HIGH
0224 1275          TAD BCDBIN
0225 7010          RAR
0226 0325          AND K7600
0227 3273          DCA LOW
0230 1272          TAD HIGH1           / MULTIPLY HIGH ORDER
0231 7104          CLL RAL             / BY THREE
```

```
0232    1272                TAD HIGH1           / FORM 128*HIGH-3*HIGH
0233    7141                CIA CLL
0234    1273                TAD LOW
0235    3273                DCA LOW
0236    7420                SNL
0237    7040                CMA
0240    1274                TAD HIGH
0241    3274                DCA HIGH            / 125*HIGH
0242    1274                TAD HIGH            / NOW MULTIPLY BY 8
0243    7106                CLL RTL
0244    7004                RAL
0245    0326                AND K7770           / MASK 9 BITS
0246    3274                DCA HIGH
0247    1273                TAD LOW
0250    7106                CLL RTL
0251    7004                RAL
0252    3273                DCA LOW
0253    1273                TAD LOW
0254    7004                RAL
0255    0324                AND K7              /3 BITS
0256    1274                TAD HIGH
0257    3274                DCA HIGH
0260    1273                TAD LOW
0261    0326                AND K7770           / 9 BITS
0262    7100                CLL
0263    1271                TAD LOW1            / ADD LOW ORDER PART
0264    3273                DCA LOW             / STORE LOW ORDER PART
0265    1274                TAD HIGH
0266    7430                SZL
0267    7001                IAC         /CARRY
0270    5600                JMP I DOUBLE

0271    0000    LOW1,   0
0272    0000    HIGH1,  0
0273    0000    LOW,    0
0274    0000    HIGH,   0
```

```
                    /SINGLE PRECISION CONVERSION
0275  0000   BCDBIN, 0
0276  3274          DCA HIGH
0277  1274          TAD HIGH
0300  0330          AND K7400        / LEFT DIGIT
0301  7112          CLL RTR
0302  3273          DCA LOW
0303  1273          TAD LOW
0304  7010          RAR
0305  1273          TAD LOW
0306  7041          CIA
0307  1274          TAD HIGH
0310  3274          DCA HIGH
0311  1274          TAD HIGH
0312  0323          AND K7760
0313  7112          CLL RTR
0314  3273          DCA LOW
0315  1273          TAD LOW
0316  7010          RAR
0317  1273          TAD LOW
0320  7041          CIA
0321  1274          TAD HIGH
0322  5675          JMP I BCDBIN
0323  7760   K7760,  7760
0324  0007   K7,     7
0325  7600   K7600,  7600
0326  7770   K7770,  7770
0327  0177   K177,   177
0330  7400   K7400,  7400
```

```
BCDBIN   0275
DOUBLE   0200
HIGH     0274
HIGH1    0272
K177     0327
K7       0324
K7400    0330
K7600    0325
K7760    0323
K7770    0326
LOW      0273
LOW1     0271
```