1. IDENTIFICATION

1.1 Digital-8-10-U-Sym

1.2 Binary-Coded-Decimal to Binary Conversion Subroutine

1.3 March 1, 1965

2. ABSTRACT

A basic subroutine for converting binary-coded-decimal numbers to their equivalent binary value. Conversion is accomplished by "radix deflation."

3. REQUIREMENTS

3.1 Storage

This subroutine requires 23 (decimal) memory locations.

3.3 Equipment

Standard PDP-8.

4. USAGE

4.1 Loading

Load the subroutine with the Binary or RIM Loader.

4.2 Calling Sequence

Call with the number to be converted in the AC. Return will be to the location following the calling JMS with the result in the AC.

6. DESCRIPTION

6.1 Discussion

The method used is that of "radix deflation." Upon entry, the BCD number may be considered to be in the following form.

6.1.1.E $$D_2 16^2 + D_1 16 + D_o$$

What is desired is the number in the form.

6.1.2.E $$D_2 10^2 + D_1 10 + D_o$$

The PDP-8 can shift (rotate) and add. A right shift is equivalent to a division by a power of two. An appropriate series of shifts, additions, and subtractions is used to convert the number from the form of 6.1.1.E to that of 6.1.2.E.

6.2 Example

Consider the BCD number

0101	0001	1001

representing the decimal number 519.

First, the whole number is stored and then brought back into the AC. Next, the four most significant bits are masked out. At this point, the accumulator contains 16x16xA or

0101	0000	0000

A shift to the right of one bit yields

0010	1000	0000

This number is stored and then brought back to the AC, shifted right two bits, and the stored value added as follows

0000	1010	0000
0010	1000	0000
0011	0010	0000

Now the original number is added to this result

0011	0010	0000
0101	0001	1001
1000	0011	1001

and the most significant eight bits masked out as

1000	0011	0000

This is stored, brought back and shifted right once, and the stored value added.

0100	0001	1000
1000	0011	0000
1100	0100	1000

Next the result of this addition is shifted right two places dividing the number by four as follows

0011	0001	0010

negated and the original number added

1100	1110	1110
0101	0001	1001
0010	0000	0111

This result represents in binary 512 plus 4 plus 2 plus 1 or 519,the original number.

6.3 Scaling

This subroutine assumes an integral BCD number and yields an integral binary equivalent.

7. METHOD

7.2 Algorithm

The algorithm used is illustrated step by step in Section 10.4.

9. EXECUTION TIME

9.2 Maximum

The maximum (and invariant) execution time of this subroutine is 49.6 microseconds.

10. PROGRAM

10.4 Program Listing

A listing of the subroutine with BCDBIN located at 0200 is given below. To simplify mnemonics D_2, D_1, and D_0 have been replaced respectively with A, B, and C.

```
0200   0000   BCDBIN,   0              /ABC IN BCD CODE IN AC
0201   3223             DCA TEMPPP
0202   1223             TAD TEMPPP     /16 (16A + B) + C
0203   0225             AND MASKKA     /16 (16A)
0204   7110             CLL RAR        /8 (16A)
0205   3224             DCA TEMPPQ
0206   1224             TAD TEMPPQ     /8 (16A)
0207   7012             RTR            /2 (16A)
0210   1224             TAD TEMPPQ     /10 (16A)
0211   1223             TAD TEMPPP     /16 (26A + B) + C
0212   0226             AND MASKKB     /16 (26A + B)
0213   3224             DCA TEMPPQ
0214   1224             TAD TEMPPQ     /16 (26A + B)
0215   7110             CLL RAR        /8 (26A + B)
0216   1224             TAD TEMPPQ     /24 (26A + B)
0217   7012             RTR            /6 (26A + B)
0220   7041             CIA            /-6 (26A + B)
0221   1223             TAD TEMPPP     /16 (16A + B) + C -6 (26A + B)
                                       /= 16x16A - 6x26A + 16B - 6xB+C
                                       /= 100A + 10B + C
0222   5600             JMP I BCDBIN   /BINARY VALUE IN AC
0223   0000   TEMPPP;   0
0224   0000   TEMPPQ,   0
0225   7400   MASKKA,   7400           /MASK FOR MOST SIG. FOUR BITS
0226   7760   MASKKB,   7760           /MASK FOR MOST SIG. EIGHT BITS
```

12. REFERENCES

12.3 DECUS Programs

See DECUSCOPE January 1965, article entitled "Accelerated Radix Deflation on the PDP-7 and PDP-8."

14. ACKNOWLEDGEMENTS

Mr. Donald V. Weaver, Consultant, of New York City, who first described the algorithm used by this subroutine in reference 12.3 has granted his kind permission to include this subroutine in the PDP-8 library so that a detailed description may be available.