cycle for summing the partial sum and partial carry vectors, the above process needs C cycles to complete, where

$$C = \left\lceil \frac{n+4}{m} \right\rceil + 2 \tag{5.22}$$

Because the CPA requires a constant time delay, which is usually lower than that of the CSA tree, the cycle period is determined primarily by the delays through the path leading to the output level of the CSA tree. These may include the delays in the multiplier recoder, the multiple complementer and gate control, the first three levels of CSAs, and the last two levels of feedback loop plus the shifter delay. With today's IC devices, these delays can be added as low as 30 nsec. Therefore, one can use the above hardware multiply unit to complete the multiplication of two 56-bit fractions in less than $\{\lceil(56+4)/(12-1)\rceil + 2\} \times 30 \simeq 225$ nsec, which is quite satisfactory to most of today's computers with memory cycle time in the neighborhood of one μsec.

5.7 Canonical Multiplier Recoding

One of the more recent advances in machine arithmetic is the use of redundancy in SD code to replace the conventional multiplier digits, such that the add-type operations can be reduced in a multiplication with the increase of average shift length across the zeros in the multiplier.

Canonical Signed-Digit Code

In an SD number with radix 2, the allowed digit set is $\{\bar{1}, 0, 1\}$. Among all the possible redundant representations of an n-digit number with a prespecified value x, we are particularly interested in the minimal SD vector, which has minimal weight as described in section 1.5. There may be more than one n-digit SD vectors that have equal minimal weight for a given value x. For example, given $n = 6$ and $x = 3$, there are eight distinct radix-2 SD vectors that have value 3. The weights of these vectors are listed below. Among the eight, the top two vectors are both minimal with weight two.

Signed-Digit Vectors	Value	Weight
$(0\ 0\ 0\ 0\ 1\ 1)_2$	$2 + 1 = 3$	2
$(0\ 0\ 0\ 1\ 0\ \bar{1})_2$	$4 - 1 = 3$	2
$(0\ 0\ 1\ \bar{1}\ 0\ \bar{1})_2$	$8 - 4 - 1 = 3$	3
$(0\ 1\ \bar{1}\ \bar{1}\ 0\ \bar{1})_2$	$16 - 8 - 4 - 1 = 3$	4
$(1\ \bar{1}\ \bar{1}\ \bar{1}\ 0\ \bar{1})_2$	$32 - 16 - 8 - 4 - 1 = 3$	5
$(0\ 0\ 1\ \bar{1}\ \bar{1}\ 1)_2$	$8 - 4 - 2 + 1 = 3$	4
$(0\ 1\ \bar{1}\ \bar{1}\ \bar{1}\ 1)_2$	$16 - 8 - 4 - 2 + 1 = 3$	5
$(1\ \bar{1}\ \bar{1}\ \bar{1}\ \bar{1}\ 1)_2$	$32 - 16 - 8 - 4 - 2 + 1 = 3$	6

A minimal SD vector $\mathbf{D} = D_{n-1} \cdots D_1 D_0$ that contains no adjacent nonzero digits is called a *canonical signed-digit vector*. This means that all nonzero digits in a canonical SD vector are separated by zeros as formally defined by

$$D_i \times D_{i-1} = 0 \quad \text{for} \quad 1 \le i \le n-1 \tag{5.23}$$

Reitweisner [17] has proved that there exists a "unique" canonical SD form D for any digital number with a fixed value α and a fixed vector length n, provided the product of the two leftmost digits in D does not equal one, that is

$$D_{n-1} \times D_{n-2} \neq 1 \tag{5.24}$$

for $\mathbf{D} = D_{n-1} D_{n-2} \cdots D_0$. This property can be always satisfied by imposing an additional digit $D_n = 0$ to the left end of the vector **D**. Without loss of generality, we shall consider $(n+1)$-digit SD vectors with a leading digit *zero* in the following discussions.

A procedure to transform a conventional binary vector to a canonical SD vector is described below.

Canonical Recoding Algorithm

Given an $(n+1)$-digit binary vector $\mathbf{B} = B_n B_{n-1} \cdots B_1 B_0$ with $B_n = 0$ and $B_i \in \{0, 1\}$ for $0 \le i \le n-1$. We wish to obtain the $(n+1)$-digit canonical SD vector $\mathbf{D} = D_n D_{n-1} \cdots D_1 D_0$ with $D_i = \{\bar{1}, 0, 1\}$ expect $D_n = 0$ such that both vectors **D** and **B** represent the same value

$$\alpha = \sum_{i=0}^{n} B_i \times 2^i = \sum_{i=0}^{n} D_i \times 2^i \tag{5.25}$$

Step 1. Start with the low-order end of **B** by setting the index $i = 0$ and initial carry $C_0 = 0$.

Step 2. Examine two adjacent bits B_{i+1} and B_i of vector **B** conditioned by the carry-in C_i and generate the next carry C_{i+1} according to the same rule of conventional binary arithmetic, that is, $C_{i+1} = 1$ if and only if there are two or three 1's among the three inputs B_{i+1}, B_i, and C_i.

Step 3. Generate the ith digit D_i of vector **D** by the following arithmetic equation

$$D_i = B_i + C_i - 2C_{i+1} \tag{5.26}$$

Step 4. Increment the index i by one and check if $i = n$. Go to Step 2 if no, and halt otherwise.

The above procedure is summarized in the Table 5.5. Let us use a numerical example to straighten out any ambiguity. Given the following 9-bit binary vector

$$\mathbf{B} = (0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1)_2 \tag{5.27}$$

Table 5.5 The Canonical Multiplier Recoding Algorithm With Radix $r = 2$.

Conventional Multiplier Bits B_{i+1}	B_i	Assumed Carry-in C_i	Recoded Bit[a] D_i	Carry-out[b] C_{i+1}
0	0	0	0	0
0	1	0	1	0
0	0	0	0	0
1	1	0	$\bar{1}$	1
1	0	1	1	0
0	1	1	0	1
0	0	1	$\bar{1}$	1
1	1	1	0	1

[a] Use Eq. 5.26 to obtain D_i.
[b] $C_{i+1} = B_{i+1}B_i + B_iC_i + B_{i+1}C_i$.

The corresponding canonical **SD** vector is obtained below using the above procedure.

$$\mathbf{D} = (0\ 1\ 0\ \bar{1}\ 0\ \bar{1}\ 0\ 0\ \bar{1})_{SD} \tag{5.28}$$

D has a value $128 - 32 - 8 - 1 = 87$ equal to that of **B**. Note that all the nonzero digits (1's or $\bar{1}$'s) in **D** are separated by zeros. Furthermore, vector **D** has a weight of 4 and **B** has a weight 5. In fact, the conventional binary representation of a digital number can be considered as a special case of the **SD** numbers.

The canonical recoding can be used in multiplier design. Only the additions of the multiples, A or $-$A, are required per each cycle. All the zeros, which separate the nonzero digits, in the recoded multiplier causes more "shifts" to be performed. The method can be extended to generating two or more signed-digits at a time. For instance, two digits D_i, D_{i+1} of **D** and the next carry C_{i+2} can be generated in one step upon the examination of three adjacent digits B_{i+2}, B_{i+1}, B_i of **B** and the incoming carry C_i. Because **D** must be canonical, adjacent digits D_{i+1} and D_i can not be both nonzero. This renders the following five possible choices of the pair, D_{i+1}, D_i, excluding 11, $\bar{1}\bar{1}$, $\bar{1}1$, or $1\bar{1}$.

$$\{\bar{1}0,\ 0\bar{1},\ 00,\ 01,\ 10\} \tag{5.29}$$

This process of generating two signed digits at a time can be visualized as a radix-4 **SD** vector with the following digit set

$$\{\bar{2},\ \bar{1},\ 0,\ 1,\ 2\} \tag{5.30}$$

corresponding to the five pairs in Eq. 5.29, respectively.

The radix-4 canonical vector can be used to design fast recoded multipliers with two-digit shifting per cycle, provided the multiples 0, $\pm$A, $\pm$2A are generated in the arithmetic processor. The use of redundancy and higher radix methods have been applied to the design of arithmetic units in the series of **ILLIAC** computers.