

Multiplication of Signed-Digit Numbers

Çetin Kaya Koç and Scott Johnson
Department of Electrical & Computer Engineering
Oregon State University
Corvallis, Oregon 97331

Abstract

A recently proposed technique for common-multiplicand multiplication of binary numbers is shown to be applicable to signed-digit numbers. We prove that multiplication of a single k -bit multiplicand by n k -bit multipliers can be performed using $0.306nk$ additions for canonically recoded signed-digit numbers, while the binary case requires $0.375nk$ additions.

Keywords: Computer arithmetic, multiplication, signed-digit numbers.

1 Multiplication of Binary Numbers

Let X and Y_1, Y_2, \dots, Y_n be k -bit 2's complement or unsigned binary numbers such that $n \geq 2$. We want to compute the numbers P_1, P_2, \dots, P_n such that $P_i = X \times Y_i$ for all $1 \leq i \leq n$. Applications of this computation is found in cryptography; for example, the RSA algorithm [6] requires computation of modular exponentiations. The exponentiation operation is broken into a series of squaring and multiplication operations by the use of the binary method [3]. The right-to-left binary method performs a series of multiplication operations, in which a common multiplicand is multiplied by several multipliers.

The standard algorithm computes $P_i = X \times Y_i$ separately for each i , which takes n multiplications. Assuming that each Y_i is a k -bit quantity, each multiplication requires on average $k/2$ additions, since randomly distributed k -bit binary numbers will have a Hamming weight of $k/2$. Thus, the standard algorithm requires $nk/2$ additions in the average case.

A more efficient method is given in [7]. Let $t \leq n$. We first compute $Y_c = Y_1 \wedge Y_2 \wedge \dots \wedge Y_t$, where \wedge is the bit-wise AND operation. Then, we compute $Y_{i,c}$ for all $i \leq t$, such that $Y_{i,c} = Y_i \oplus Y_c$ where \oplus is the bit-wise XOR operation. It can be easily seen that $Y_i = Y_{i,c} + Y_c$. Thus, $P_i = X \times Y_i = X \times (Y_{i,c} + Y_c) = (X \times Y_{i,c}) + (X \times Y_c)$. It was shown in [7] that, using this technique, only $3nk/8 = 0.375nk$ additions will be required in the average case to perform n k -bit common-multiplicand multiplications.

2 Signed-Digit Numbers

Recoding techniques (Booth recoding, bit-pair recoding, etc.) for sparse representations of binary numbers have been effectively used in multiplication algorithms [4]. For example, the original Booth recoding technique scans the bits of the multiplier one bit at a time, and adds or subtracts the multiplicand to or from the partial product, depending on the value of the current bit and the previous bit. The modified versions of the Booth algorithm scan the bits of the multiplier two bits at a time or three bits at a time. These techniques are equivalent in the sense that the identity

$2^{i+j} - 2^i = 2^{i+j-1} + 2^{i+j-2} \dots + 2^{i+1} + 2^i$ is used to collapse blocks of 1's appearing in a binary representation. In a signed-digit number with radix 2, three symbols $\{\bar{1}, 0, 1\}$ are allowed for the digit set, in which 1 and $\bar{1}$ in bit position i represent $+2^i$ and -2^i , respectively.

The recoding is called canonical if it contains no adjacent nonzero digits. The canonical signed-digit vector can be constructed by the algorithm of Reitwiesner [5]. Reitwiesner's algorithm computes the recoded number starting from the least significant digit and proceeding to the left. First the auxiliary carry variable C_0 is set to 0 and subsequently the binary number A is scanned two bits at a time. The canonically recoded digit B_i and the next value of the auxiliary binary variable C_{i+1} for $i = 0, 1, 2, \dots, n$ are generated using Table 1.

Table 1: Canonical recoding.

A_{i+1}	A_i	C_i	B_i	C_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	$\bar{1}$	1
1	1	0	$\bar{1}$	1
1	1	1	0	1

As an example, when $A = 3038$, we compute the canonical signed-digit vector B as

$$A = (0101111011110) = (10\bar{1}0000\bar{1}000\bar{1}0) = B .$$

Note that in this example the number A contains 9 nonzero bits, while its canonically recoded version contains only 4 nonzero digits. It has been shown [1, 2] that the average Hamming weight of a k -bit canonically recoded binary number approaches $k/3$ as $k \rightarrow \infty$.

3 Multiplication Algorithm

Let X be a k -bit binary number, and Y_1, Y_2, \dots, Y_n be k -digit canonically recoded numbers. We will assume that k is sufficiently large so that the average Hamming weight of Y_i is approximately equal to $k/3$. Common-multiplicand multiplication using the standard method requires n multiplications, each of which requires $k/3$ additions on the average. Thus, a total of $nk/3$ additions will be required. In order to apply the technique of [7], we first define the \wedge and \oplus operators over the set $\{0, 1, -1\}$ as follows:

Table 2: Operators \wedge and \oplus .

\wedge	0	1	-1	\oplus	0	1	-1
0	0	0	0	0	0	1	-1
1	0	1	0	1	1	0	0
-1	0	0	-1	-1	-1	0	0

These operators are commutative and associative, which can be proven by checking all combinations. With these definitions, the multiplication of canonically recoded numbers is quite simple. First we compute $Y_c = Y_1 \wedge Y_2 \wedge \dots \wedge Y_t$, using the new definition for \wedge , and compute $Y_{i,c} = Y_i \oplus Y_c$ for all $i \leq t$, using the new definition for \oplus . As in the case of binary numbers, $Y_i = Y_{i,c} + Y_c$. Thus,

we can compute $P_i = P_{i,c} \times P_c = X \times (Y_{i,c} + Y_c)$. We compute P_i for all $1 \leq i \leq n$ by breaking the set Y_1, Y_2, \dots, Y_n up into $\lfloor n/t \rfloor$ subsets with t -element each, and 1 subset with $n \bmod t$ elements.

We assume that the two possible non-zero digits, 1 and -1 , occur with equal probability. Furthermore, since $Pr(0) = 2/3$, we have $Pr(1) = Pr(-1) = 1/6$. Now, note the behavior of the new \wedge operator, as defined above. Given Q_1, Q_2, \dots, Q_t with $Q_i \in \{0, 1, -1\}$ for all $1 \leq i \leq t$, we compute $Q_c = Q_1 \wedge Q_2 \wedge \dots \wedge Q_t$. Q_c will be equal to 1 if and only if $Q_i = 1$ for all $1 \leq i \leq t$. Similarly, Q_c will be equal to -1 if and only if $Q_i = -1$ for all $1 \leq i \leq t$. In all other cases, Q_c will be equal to zero. As a result, $Pr(Q_c = 1) = (Pr(1))^t = 6^{-t}$ and $Pr(Q_c = -1) = (Pr(-1))^t = 6^{-t}$. Thus, the average Hamming weight of Y_c is equal to $2 \times 6^{-t} \times k$, and the average Hamming weight for each of the $Y_{i,c}$ terms is equal to

$$k/3 - 2 \times 6^{-t} \times k = (1 - 6^{-t+1}) \times (k/3) .$$

Thus, the total number of additions needed to perform common-multiplicand multiplication on t numbers is found as

$$2 \times 6^{-t} \times k + (1 - 6^{-t+1}) \times (k/3) \times t .$$

Ignoring the additions required to compute $P_i = P_{i,c} + P_c$, we compute the performance improvement over the standard algorithm as

$$\frac{t/3}{2 \times 6^{-t} + (1 - 6^{-t+1}) \times (1/3) \times t} .$$

By inserting appropriate values of t , we can determine the increase in performance for common-multiplicand multiplication of canonically recoded numbers. As was the case for binary numbers, it is easily shown that the performance improvement is maximized when $t = 2$. Larger arrays can be dealt with by breaking the array up into pairs. By substituting 2 for t into the above formula, we calculate that the performance improvement for the canonically recoded numbers as $12/11$. Thus, the common-multiplicand multiplication of n k -digit canonically recoded numbers takes $nk \times (1/3) \div (12/11) = 11nk/36 \approx 0.306nk$ additions.

References

- [1] S. Arno and F. S. Wheeler. Signed digit representations of minimal Hamming weight. *IEEE Transactions on Computers*, 42(8):1007–1010, August 1993.
- [2] Ö. Eğecioğlu and Ç. K. Koç. Exponentiation using canonical recoding. *Theoretical Computer Science*, 129(2):407–417, 1994.
- [3] D. E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*, volume 2. Reading, MA: Addison-Wesley, Second edition, 1981.
- [4] I. Koren. *Computer Arithmetic Algorithms*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [5] G. W. Reitwiesner. Binary arithmetic. *Advances in Computers*, 1:231–308, 1960.
- [6] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
- [7] S.-M. Yen and C.-S. Lai. Common-multiplicand multiplication and its applications to public key cryptography. *Electronics Letters*, 29(17):1583–1584, 19th August 1993.