

# A Low-Error and Area-Time Efficient Fixed-Width Booth Multiplier

Min-An Song, Lan-Da Van\*, Ting-Chun Huang, and Sy-Yen Kuo

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C

\*Chip Implementation Center (CIC), National Applied Research Laboratories, Taiwan, R.O.C.

E-mail: [sykuo@cc.ee.ntu.edu.tw](mailto:sykuo@cc.ee.ntu.edu.tw) and [ldvan@cic.org.tw](mailto:ldvan@cic.org.tw)

## Abstract

In this paper, we develop a new methodology for designing a lower-error and area-time efficient 2's-complement fixed-width Booth multiplier that receives two  $n$ -bit numbers and produces an  $n$ -bit product. By properly choosing the generalized index and binary thresholding, we derive a better error-compensation bias to reduce the truncation error. Since the proposed error-compensation bias is realizable, the constructing low-error fixed-width Booth multiplier is area-time efficient for VLSI implementation. Finally, we successfully apply the proposed fixed-width Booth multiplier to speech signal processing. The simulation results show that the performance is superior to that using the direct-truncation fixed-width Booth multiplier.

## 1. Introduction

In many digital signal processing (DSP) applications such as digital filters [6, 7] and wavelet transform, it is desirable to maintain fixed-width output word through the arithmetic operations. So, a low-error fixed-width multiplier [5-8] that receives an  $n$ -bit multiplier as well as an  $n$ -bit multiplicand and produces an  $n$ -bit output product is the most important processing element for digital computing engine. In view of the algorithm level, most fixed-width multipliers are based on either the Baugh-Wooley algorithm [1] or the Booth algorithm [2, 3]. The Baugh-Wooley based fixed-width multipliers [5-7] have been widely studied. King and Swartzlander [5] analyzed an adaptive error-compensation bias and proposed an  $n$ -bit fixed-width multiplier. In [6, 7], we generalized this kind of Baugh-Wooley based fixed-width multipliers by properly choosing the generalized index and binary thresholding. Thus, several lower-error and area-efficient fixed-width multipliers can be obtained. However, the area-time efficient fixed-width multiplier cannot be fully achieved by the Baugh-Wooley based fixed-width multipliers. Therefore, the fixed-width Booth algorithm is, currently one of the research topics.

The Booth algorithm is widely used in the implementation of ASIC-oriented products because a higher-speed and smaller multiplier can be obtained. The modified Booth algorithm was proposed by the MacSorley [3] in which a triplet of bits is scanned at a time. It is known that the recoding technique of the modified Booth algorithm has two main advantages. One is that almost half the partial products compared to the Baugh-Wooley multiplier can be saved. Hence, the number of rows of the subproduct array can be reduced by 2. The other is that, based on the first advantage, the critical delay time can be shorter than that of the Baugh-Wooley multiplier. Area saving of a fixed-width Booth multiplier can be achieved by directly truncating  $n$  least significant product bits and preserving  $n$  most significant product bits. With this method, significant truncation errors would be introduced since no error compensation is considered. Thus, the Booth-based scheme in [8] has been proposed to reduce the

truncation error; however, the proposed scheme [8] that lacks for systematic analysis and derivation does not be a systematic methodology. In this paper, we are motivated to propose a systematic design methodology for low-error area-time efficient Booth multipliers. The methodology includes the following steps in order: 1) Propose an error-compensation bias with a new binary thresholding; 2) simulate the value of  $K$  and error performance of the proposed error-compensation bias using our generalized index, and then select the best index having lower error and satisfying the same value of  $K$  for small width  $n$ ; and 3) construct a low-error Booth multiplier structure. Based on our methodology, the resulting error compensation circuit can be easily realized without any area overhead under the limited truncation error. The organization of this paper is as follows. The modified Booth algorithm is concisely reviewed in Section 2. In Section 3, we propose a better error-compensation bias and present the simulation results for small width  $n$ . The improved error-compensation bias can be mapped to a new structure. The performance of the proposed fixed-width Booth multiplier are described in Section 4. Finally, brief statements in section 5 conclude the presentation.

## 2. Modified Booth Multiplier

Considering the multiplication of two 2's-complement integers with  $n$ -bit multiplicand  $A$  and  $n$ -bit multiplier  $B$  as

$$P = AB = \sum_{i=0}^{2n-1} P_i 2^i \quad (1)$$

where  $A = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$ ,  $B = -b_{n-1}2^{n-1} + \sum_{j=0}^{n-2} b_j 2^j$ , and

$P_i$  denotes the  $i$ -th output product bit. Note that  $a_i$  and  $b_i$  indicate data bits of multiplicand and multiplier, respectively. Assume  $n$  is even and the  $n$ -bit multiplier  $B$  can be rewritten

$$B = \sum_{i=0}^{(n-2)/2} (b_{2i-1} + b_{2i} - 2b_{2i+1}) 2^{2i}, \quad (2)$$

where  $b_{-1} = 0$ . Note that the terms in the bracket in Eq (2) have values of  $\{-2, -1, 0, 1, 2\}$ . Each recoded value performs a certain operation on the multiplicand  $A$ , and then the multiple additions at each stage would be required in order to generate the correct partial product. It is worth mentioning that the operation of  $-A$  can be realized by the inversion of the multiplicand and addition of '1' at the least significant bit as illustrated in Table 1. Substituting Eq. (2) into Eq. (1), we can obtain Eq. (3) as

$$P = AB = \sum_{i=0}^{(n-2)/2} (b_{2i-1} + b_{2i} - 2b_{2i+1}) A 2^{2i} = \sum_{i=0}^{(n-2)/2} S_i, \quad (3)$$

where  $S_i = (b_{2i-1} + b_{2i} - 2b_{2i+1})A2^{2i}$ , and it is known that the scanning of triplets begins from  $b_{-1}$  to the MSB with one-bit overlapping. Thus, only the number of  $n/2$  partial-product rows needs to be computed.

Table 1: Modified Booth Encoding Table

$b_{2i+1}$	$b_{2i}$	$b_{2i-1}$	Operation on A	ADD to LSB
0	0	0	0	0
0	0	1	+A	0
0	1	0	+A	0
0	1	1	+2A	0
1	0	0	-2A	1
1	0	1	-A	1
1	1	0	-A	1
1	1	1	0	0

So as to simplify the representation of the bit-product of each row for the Booth algorithm, we define the following notation

$$S_i = S_{i,n-1}2^{2i+n-1} + S_{i,n-2}2^{2i+n-2} + \dots + S_{i,0}2^{2i}, \quad (4)$$

where  $S_{i,j}$  represents the bit product of the  $i$ -th row. Based on the conventional Booth arithmetic operations, sign extension of the partial products are required for each stage. However, the extended sign bits result in a larger power and area. According to the sign-generate sign extension scheme [4], for an 8 by 8 multiplier, the sign of the final result can be expressed as

$$\begin{aligned} S &= (S_{0,7} \sum_{j=8}^{15} 2^j)2^0 + (S_{1,7} \sum_{j=8}^{13} 2^j)2^2 + (S_{2,7} \sum_{j=8}^{11} 2^j)2^4 + (S_{3,7} \sum_{j=8}^9 2^j)2^6 \\ &= (2^9 + \overline{S_{0,7}}2^8) + (2^{11} + \overline{S_{1,7}}2^{10}) + (2^{13} + \overline{S_{2,7}}2^{12}) \\ &\quad + (2^{15} + \overline{S_{3,7}}2^{14}) + 2^8, \end{aligned} \quad (5)$$

where  $S$  is the final result sign. From Eq. (5), the partial products of the Booth algorithm only need to add two elements ( $1, \overline{S_{i,7}}$ ) for each row and add an extra '1' in the  $2^8$ -weight position as shown in Fig.1, where main and remain represent main part and remain part of the least significant bit (LSB), respectively. Thus, the sign-generate sign extension scheme can reduce a lot of redundant full adders compared to the conventional sign extension method. The architecture of the Booth Multiplier as shown in Fig.2 consists of Booth encoders, selectors (sel), full adders (FA) and half adders (HA). The Booth encoder generates  $Ctrl_i[0:2]$  signals to control the selector to choose -2A, -A, 0, 1A or 2A.

### 3. Design of Fixed-Width Booth Multiplier

The  $2n$  product for  $n$  by  $n$  2's-complement multiplication can be divided into two sections as

$$P = AB = MP + LP. \quad (6)$$

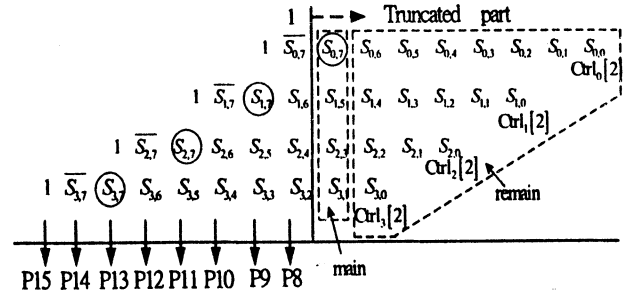


Fig. 1. Modified Booth partial-product diagram with sign-generate sign extension scheme for an 8x8 multiplier.

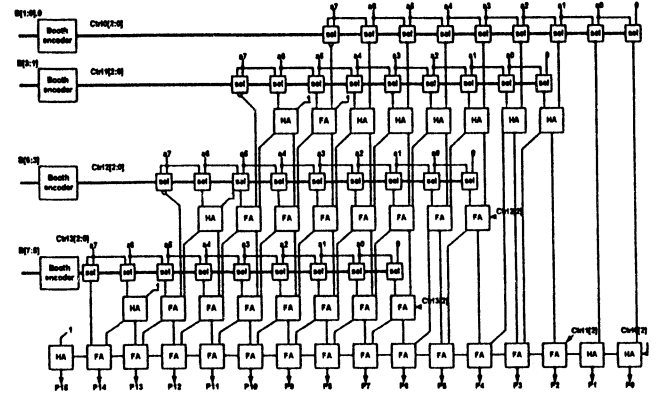


Fig. 2. An 8x8 modified Booth multiplier using sign-generate sign extension scheme.

The most accurate truncation product is given by

$$P \cong MP + \sigma_{Temp} \times 2^n, \quad (7)$$

$$\sigma_{Temp} = [LP]_r. \quad (8)$$

Without loss of generality, for  $n=8$ , Eq. (8) can be denoted as

$$\sigma_{Temp} = \left[ \begin{aligned} &\frac{1}{2}(S_{3,1} + S_{2,3} + S_{1,5} + S_{0,7}) + \frac{1}{2^2}(S_{3,0} + \dots + S_{0,6}) \\ &+ Ctrl_3[2] + \dots + \frac{1}{2^7}S_{0,1} + \frac{1}{2^8}(S_{0,0} + Ctrl_0[2]) \end{aligned} \right]_r. \quad (9)$$

Then we define the following terms

$$E_{main} = S_{3,1} + S_{2,3} + S_{1,5} + S_{0,7}, \quad (10)$$

$$\begin{aligned} E_{remain} &= \frac{1}{2}(S_{3,0} + S_{2,2} + S_{1,4} + S_{0,6} + Ctrl_3[2]) \\ &+ \dots + \frac{1}{2^7}(S_{0,0} + Ctrl_0[2]) \end{aligned} \quad (11)$$

Thus, we can rewrite Eq. (9) as

$$\sigma_{Temp} = \left[ \frac{1}{2}(E_{main} + E_{remain}) \right]_r. \quad (12)$$

It is convenient to perform exhaustive simulation if we define the generalized index. Here the generalized index for 8 by 8 multipliers is defined as

$$\theta_{index}(q_3, q_2, q_1, q_0) = \langle S_{3,1} \rangle^{q_3} + \langle S_{2,3} \rangle^{q_2} + \langle S_{1,5} \rangle^{q_1} + \langle S_{0,7} \rangle^{q_0} \quad (13)$$

where the binary parameters  $q_3, q_2, q_1$  and  $q_0 \in \{0,1\}$ , and the operator

$$\langle T \rangle^{q_i} = \begin{cases} T, & \text{if } q_i = 0 \\ \bar{T}, & \text{if } q_i = 1 \end{cases} \quad (14)$$

in which  $\bar{T}$  is the complement of binary  $T$ . Furthermore,  $\theta_{index}(q_3, q_2, q_1, q_0)$  is referred to as  $\theta_Q$ , where

$$Q = q_3 \times 2^3 + q_2 \times 2^2 + q_1 \times 2^1 + q_0 \times 2^0. \quad (15)$$

Note that  $Q$  has a range from 0 to 15. So as to evaluate the resulting performance, by applying Eqs. (13) to (15) into Eq. (12), we get

$$\sigma_{Temp} = \theta_Q + \left[ \frac{1}{2} E_{main} - \theta_Q + \frac{1}{2} E_{remain} \right]_r$$

$$= (\langle S_{3,1} \rangle^{q_3} + \langle S_{2,3} \rangle^{q_2} + \langle S_{1,5} \rangle^{q_1} + \langle S_{0,7} \rangle^{q_0}) + [K]_r, \quad (16)$$

where

$$K = \frac{1}{2} E_{main} - \theta_Q + \frac{1}{2} E_{remain}. \quad (17)$$

Based on the binary thresholding concept in [6, 7], Eq. (16) can be approximated as

$$\sigma_{Temp} = \begin{cases} \langle S_{3,1} \rangle^{q_3} + \langle S_{2,3} \rangle^{q_2} + \langle S_{1,5} \rangle^{q_1} + \langle S_{0,7} \rangle^{q_0} + [K_1]_r, & \text{if } \theta_Q < 4 \\ \langle S_{3,1} \rangle^{q_3} + \langle S_{2,3} \rangle^{q_2} + \langle S_{1,5} \rangle^{q_1} + \langle S_{0,7} \rangle^{q_0} + [K_2]_r, & \text{if } \theta_Q = 4 \end{cases} \quad (18)$$

where  $K_1$  and  $K_2$  are respectively the average of  $K$  for those satisfying  $\theta_Q < 4$  and  $\theta_Q = 4$ . In order to design a simple and realizable error-compensation circuit, we choose the indices which satisfy  $[K_1]_r \in \{-1, 0, 1\}$  and  $[K_2]_r \in \{-1, 0, 1\}$ . By exhaustive search simulation, we obtain values of  $K_1$  and  $K_2$  as shown in Fig. 3(a) for  $n = 8$ .

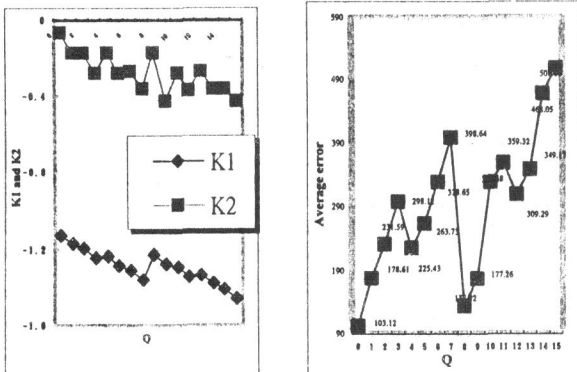


Fig. 3. (a) Values of  $K_1$  and  $K_2$  versus different  $Q$  of the binary thresholding. (b) Average errors by exhaustive search simulation versus different  $Q$  of the binary thresholding for  $n = 8$ .

In accordance with the rounding values of  $K_1$  and  $K_2$ , we simulate average error as shown in Fig. 3(b). Considering the goal of lower error and the restriction of  $K$ , we can find that  $\theta_{Q=0}$  index has better performance as shown in Fig. 3(b) with  $[K_1]_r = -1$  and  $[K_2]_r = 0$  for  $n = 8$ . Thus, the simple error-compensation circuit can be described as

$$\sigma_{Q=0} = \begin{cases} S_{3,1} + S_{2,3} + S_{1,5} + S_{0,7} - 1, & \text{if } \theta_{Q=0} < 4 \\ S_{3,1} + S_{2,3} + S_{1,5} + S_{0,7} + 0, & \text{if } \theta_{Q=0} = 4 \end{cases} \quad (19)$$

where  $\theta_{Q=0} = S_{3,1} + S_{2,3} + S_{1,5} + S_{0,7}$ . Eq. (19) has been completely simulated for  $n \leq 16$  and can be mapped to a new structure as shown in Fig. 4. Note that the error-compensation circuit only needs three AND gates.

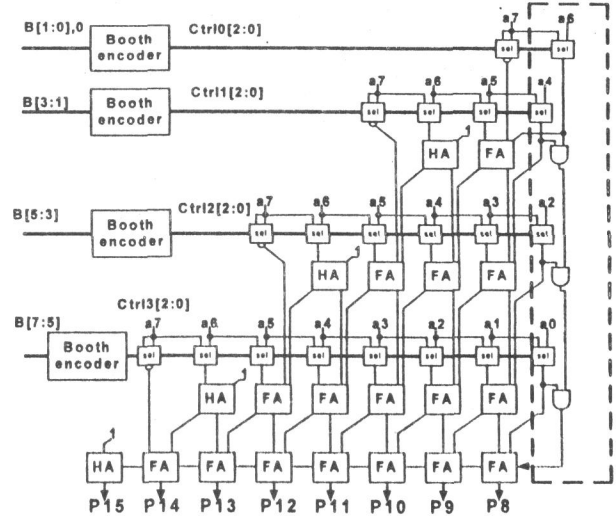


Fig. 4. Proposed low-error fixed-width  $8 \times 8$  Booth multiplier with  $\theta_{Q=0}$ .

#### 4. Performance and Application Discussion

In this section, we first simulate error performance in terms of maximum error, average error, and variance of error as listed in Table 2 between the direct-truncation multiplier and the proposed fixed-width Booth multiplier. It is clearly seen that the new structure can achieve better error performance than the direct-truncation Booth multiplier. Regarding the number of gates and the critical path delay time issues as listed in Table 3, comparison results show that the new Booth multiplier saves much area cost with respect to the full-precision Booth multiplier based on the sign-generate sign extension scheme. Most importantly, the gate count and critical delay time of the proposed structure are close to those of the direct truncation multiplier, respectively. Thus, the proposed fixed-width Booth multiplier has the area-time efficient feature with better error performance. On the other hand, we apply the proposed fixed-width multiplier to the 35-tap FIR filter for speech processing [9]. For convenience of comparison of various fixed-width multipliers, we take 1000 samples for the consonant part and vowel part of "Chicken" as shown in Fig. 5(a). We are concerned with whether the filtered waveform is accurate via our proposed

fixed-width multiplier, so the correct standard output is required. We use error-free output as a standard, which is used to compare the accuracy performances of fixed-width multipliers. Fig. 5(b) shows the standard filtering output signals and Figs. 5(c) shows the filtering output signals processed by the 35-tap low-pass FIR filter applying a direct truncation fixed-width multiplier. Using direct truncation multiplier, it is seen from Fig. 5(c) that there are large average error and variance of errors in the consonant part. The smaller average error and variance of the errors especially for consonant part is obtained by using our proposed fixed-width Booth multiplier as shown in Fig. 5(d).

## 5. Conclusions

This paper develops a new methodology for designing a low-error and area-time efficient fixed-width Booth multiplier. By properly choosing binary thresholding and the generalized index, we can derive a better error-compensation bias to improve the truncation error. Furthermore, this error-compensation bias can be easily constructed as a lower-error fixed-width Booth multiplier. It is very suitable for VLSI digital signal processing applications where the accuracy, area, and speed issues are crucial. Finally, we successfully apply the proposed fixed-width multiplier to a digital FIR filter for speech processing application.

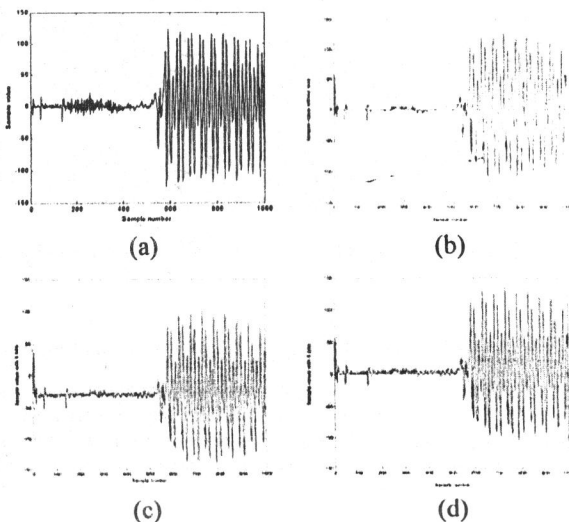


Fig. 5. (a) Original input voice signal, (b) standard output voice signal, (c) output signals using the direct truncation structure and (d) output signals using the proposed Booth multiplier structure.

## 6. References

- [1] C. R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithms", *IEEE Trans. Comput.*, vol. C-22, pp. 1045-1047, Dec. 1973.
- [2] A. D. Booth, "A signed binary multiplication technique," *Quarterly Journal of Mechanics and Applied Mathematics*, vol. 4, part 5, pp. 236-240, 1951.
- [3] O. L. MacSorley, "High-speed arithmetic in binary computer", *Proc. IRE*, vol. 49, pp. 67-91, 1961.

- [4] E. de Angel and E. E. Swartzlander, Jr., "Low power parallel multipliers," *IEEE VLSI Signal Processing IX*, 1996, pp. 199-208.
- [5] E. E. Swartzlander, Jr., "Truncated multiplication with approximate rounding," in *Proc. 33rd Asilomar Conference on Signals, Systems, and Computers*, 1999, vol. 2, pp. 1480-1483.
- [6] L. D. Van, S. S. Wang, and W. S. Feng, "Design of the lower-error fixed-width multiplier and its application", *IEEE Trans. Circuits Syst. II*, vol. 47, pp. 1112-1118, Oct. 2000.
- [7] L. D. Van and S. H. Lee, "A generalized methodology for lower-error area-efficient fixed-width multipliers," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2002, vol. 1, pp. 65-68, Phoenix, Arizona.
- [8] S. J. Jou and H. H. Wang, "Fixed-width multiplier for DSP application," *IEEE Int. Conf. Computer Design*, Sep. 2000, pp. 318-322.
- [9] M. E. Paul, and K. Bruce, *C Language Algorithms for Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

Table 2: Comparison Results of Three Kinds of Errors among Different Booth Multipliers

Multiplier	Width	Maximum Error	Average Error	Variance of Error
Direct Truncation Multiplier	4	32	10.88	67.20
	6	192	70.50	1465.86
	8	1024	384.25	28510.19
	16	524288	196608.25	3661149123
This Work	4	16	4.59	28.50
	6	85	21.60	716.86
	8	443	103.12	16376.65
	16	504268	62501.62	1486362529

Table 3: Comparison Results of Area and Critical Delay Time among Different Booth Multipliers for  $n = 8$

Multiplier	Area (# of gate counts)			Critical Delay Time
	FA	HA	Selector	
Full Precision Multiplier Based on Sign-Generate	28	12	36	$13T_{FA} + 3T_{HA}$
Direct Truncation Multiplier	11	8	16	$7T_{FA} + 3T_{HA}$
This work	16	4	20	$10T_{FA} + T_{HA}$