A Generalized Methodology for Low-Error and Area-Time Efficient Fixed-Width Booth Multipliers

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
*National Chip Implementation Center (CIC), National Applied Research Laboratories, Taiwan, R.O.C.
E-mail: sykuo@cc.ee.ntu.edu.tw and ldvan@cic.org.tw

Abstract

In this paper, we extend our generalized 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. The generalized methodology involving three steps results in several better error-compensation biases. These better error-compensation biases can be mapped to low-error fixed-width Booth multipliers suitable for VLSI implementation. Finally, we successfully apply the proposed fixed-width Booth multipliers 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 [1, 2] and wavelet transform, it is desirable to maintain fixed-width output word through the arithmetic operations. The Baugh-Wooley based fixed-width multipliers [1-5] have been widely studied. King and Swartzlander [3] analyzed an adaptive error-compensation bias and proposed an n-bit fixed-width multiplier. In [1, 2], we generalized this kind of Baugh-Wooley based fixed-width multipliers by properly choosing the generalized index and binary thresholding. The above scheme is based on keeping n+w colums of the subproduct array, where w is a nonnegative integer between 0 and n-1. 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 Bauth-Wooley based fixed-width multipliers. Therefore, the fixed-width Booth algorithm is currently one of the research topics.

The modified Booth algorithm proposed by the MacSorley [6] 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. 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 for a fixed value of w; 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; 3) construct a low-error Booth multiplier structure. Based on our methodology, while w=1, the proposed fixed-width Booth multiplier also operates lower error than those in [7] at the expense of slightly increased area-ratio with respect to each value of w. 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 with respect to each value of w. The performance of the proposed is 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

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

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 as

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

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. Substituting Eq. (2) into Eq. (1), we can obtain Eq. (3) as

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

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.

0-7803-8346-X/04/$20.00 ©2004 IEEE

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

$$S_i = S_{i,n-1}2^{2i+n-1} + S_{i,n-2}2^{2i+n-2} + \ldots + S_{i,0}2^{2i}, \tag{4}$$

where $S_{i,j}$ represents the bit product of the i-th row. According to the sign-generate sign extension scheme [8], for an 8 by 8 multiplier, the sign of the final result can be expressed as

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

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 many redundant full adders compared to the conventional sign extension method.

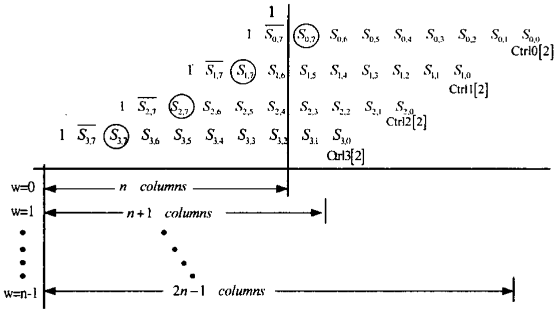

Fig. 1. Modified Booth partial-product diagram with sign-generate sign extension scheme for an 8x8 multiplier.

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.

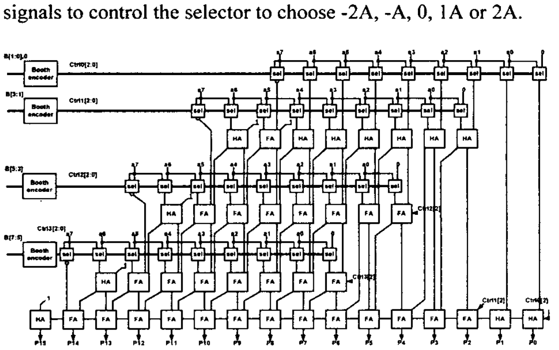

Fig. 2. An 8×8 modified Booth multiplier using sign-generate sign extension scheme.

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

$$P = AB = MP + LP. \tag{6}$$

The most accurate truncation product is given by

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

$$\sigma_{Temp} = [LP]_r. \tag{8}$$

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

$$\sigma_{Temp} = \begin{bmatrix} \frac{1}{2}(S_{3,1} + S_{2,3} + S_{1,5} + S_{0,7}) + \frac{1}{2^2}(S_{3,0} + \ldots + S_{0,6} \\ + Ctrl_3[2]) + \ldots + \frac{1}{2^7}S_{0,1} + \frac{1}{2^8}(S_{0,0} + Ctrl_0[2]) \end{bmatrix}_r. \tag{9}$$

Then we define the following terms

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

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

Thus, we can rewrite Eq. (9) as

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

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

$$\begin{aligned} \theta_{index,w}(q_3,q_2,q_1,q_0) &= <S_{3,1-w}>^{q_3} + <S_{2,3-w}>^{q_2} + <S_{1,5-w}>^{q_2} \\ &\quad + <S_{0,7-w}>^{q_0} \end{aligned} \tag{13}$$

where the binary parameters $q_{3-w}, q_{2-w}, \ldots, q_0 \in \{0,1\}$, and the operator

$$<T>^{q_i} = \begin{cases} T & ,\text{if } q_i = 0 \\ \overline{T}, & ,\text{if } q_i = 1 \end{cases}, \tag{14}$$

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

$$Q = q_3 \times 2^3 + q_2 \times 2^2 + q_1 \times 2^1 + q_0 \times 2^0. \tag{15}$$

For example, the value of Q has a range from 0 to15 for w=0 and 1. Note that if the value of the second index of $S_{i,j}$ in Eq. (13) is less than zero, the $S_{i,j}$ can be neglected. In [4,5], they show that lower truncation error can be obtained if larger $n+w$ columns are kept in hardware. However, more area cost could be increased. Since the reduction and rounding errors do not own the same weight position, we adopt S-Ss' method [8] to concurrently treat reduction and rounding error. By applying Eqs. (13) to (15) into Eq. (12), we get

$$\sigma_{Temp} = \begin{bmatrix} \frac{1}{2}E_{main} + \frac{1}{2}E_{remain} + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} + \\ E_{reduct,w} - \frac{1}{2^w}\theta_{Q,w} \end{bmatrix}_r$$

$$\cong \left\lfloor \begin{pmatrix} \frac{1}{2}E_{main} + \frac{1}{2}E_{remain} + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} \\ + [K]_r / 2^w \end{pmatrix} \right\rfloor, \quad (16)$$

where

$$K = 2^w \left(E_{reduct,w} - \frac{1}{2^w}\theta_{Q,w} + E_{round,w} \right), \quad (17)$$

$$\begin{aligned} E_{reduct,w} &= \frac{1}{2^{w+1}}(S_{3,1-w} + S_{2,3-w} + S_{1,5-w} + S_{0,7-w}) \\ &+ \ldots + \frac{1}{2^n}S_{0,0} + (\frac{1}{2^2}Ctrl_{3,1-w}[2] + \frac{1}{2^4}Ctrl_{2,3-w}[2] \\ &+ \frac{1}{2^6}Ctrl_{1,5-w}[2] + \frac{1}{2^8}Ctrl_{0,7-w}[2]), \end{aligned} \quad (18)$$

$$E_{round,w} = 2^{-1}\left(1 - 2^{-w}\right). \quad (19)$$

Herein, the second index of the control signal in Eq. (18) denotes whether the control signal exists. In case the value of the second index is less than zero, the control signal can be neglected. Note that the least significant weight of K must be limited to the $n+w$ weight position. Concurrently treating method for reduction and rounding errors of Eq. (16). In the first step, to design a realizable error-compensation bias, two types of binary thresholding for the error-compensation bias can be change to

Type 1:

$$\sigma_{Type1,Q,w\geq 1} = \begin{cases} \left\lfloor \frac{1}{2}(E_{main} + E_{remain}) + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} + [K_1]/2^w \right\rfloor, if \quad \theta_{Q,w} = 0 \\ \left\lfloor \frac{1}{2}(E_{main} + E_{remain}) + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} + [K_2]/2^w \right\rfloor, if \quad \theta_{Q,w} > 0 \end{cases} \quad (20)$$

Type 2:

$$\sigma_{Type2,Q,w\geq 1} = \begin{cases} \left\lfloor \frac{1}{2}(E_{main} + E_{remain}) + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} + [K_3]/2^w \right\rfloor, if \quad \theta_{Q,w} = 4 \\ \left\lfloor \frac{1}{2}(E_{main} + E_{remain}) + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} + [K_4]/2^w \right\rfloor, if \quad \theta_{Q,w} < 4 \end{cases} \quad (21)$$

where K_1, K_2, K_3 and K_4 are defined as those of [1] but for w=1. The restriction of the value of K can be modified as $[K_i]_r \in \{0, 1, 2^{w-1} - 1, 2^{w-1}\}$ for i=1,2,3 and 4. For w=1, since using the same simulation procedures as mentioned in [1], we only introduce the analysis for $w = 1$ and construct the structure. In Type 1 binary thresholding, by exhaustive search we can find that one good index, shown in Fig. 3(a). We observe that the specific index, $\theta_{Q=0,w=1}$ achieves better error performance where the chosen index satisfies $[K_1]_r = 1$ and $[K_2]_r = 0$, we simulate average error as shown in Fig. 3(b). On the other hand, for Type 2 binary thresholding, the error simulation in terms of average errors are large than what find error resulted from the best index in Type 1 thresholding, so we ignore the discussion in Type 2. So far, the second step is achieved. Hence, a new lower error fixed-width Booth multiplier under w=1 can be described and simplified as:

$$\sigma_{Type1,Q=0,w=1} = \begin{cases} \left\lfloor \frac{1}{2}(E_{main} + \theta_{Q=0,w=1}) + \frac{1}{2} \right\rfloor, \text{if } \theta_{Q=0,w=1} = 0 \\ \left\lfloor \frac{1}{2}(E_{main} + \theta_{Q=0,w=1}) + 0 \right\rfloor, \text{if } \theta_{Q=0,w=1} > 0 \end{cases} \quad (22)$$

where $\theta_{Q=0,w=1} = S_{3,0} + S_{2,2} + S_{1,4} + S_{0,6}$. In the third step, Eq. (22) can be mapped to a new structure as shown in Fig. 4. From simulation results, $\theta_{Q=0,w=1}$ in Type 1 binary thresholding is still the best index for w=1. Note that the error-compensation circuit only needs three basic element gates.

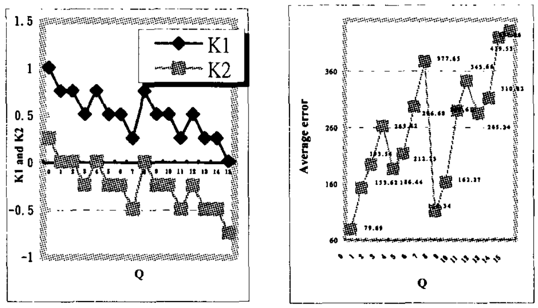

Fig. 3. (a) Values of K1 and K2 versus different Q of the binary thresholding. (b) Average errors by exhaustive search simulation versus different Q of the binary threshoding for $n = 8$.

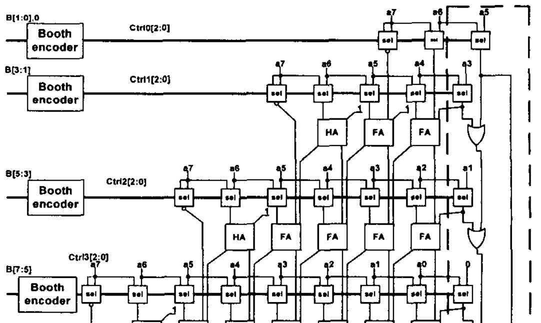

Fig. 4. Proposed low-error fixed-width 8 × 8 Booth multiplier with $\theta_{Q=0,w=1}$.

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 1 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 2, 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" .We are concerned with whether the filtered waveform is accurate via our proposed fixed-width Booth 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 Booth multipliers. From comparison results obtain with four fixed-width Booth multipliers as show in Fig. 5 for speech processing application, we observed that Type 1 multiplier with $\theta_{Q=0,w=1}$ shows better performance in the consonant and vowel parts.

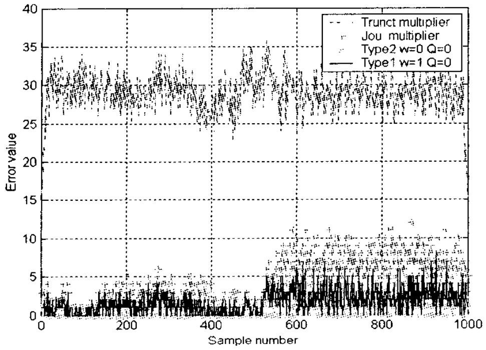

Fig. 5. Comparison results of error signals obtain with four kind of fixed-width multipliers.

5. Conclusions

This paper develops a new methodology for designing two low-error and area-time efficient fixed-width Booth multipliers. By properly choosing binary thresholding and the generalized index, we can derive several better error-compensation biases to improve the truncation error. Furthermore, these error-compensation biases can be easily constructed as lower-error fixed-width Booth multipliers. 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.

6. References

[1] 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.

[2] 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.

[3] 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.

[4] Y. P. Lim, "Single-precision multiplier with reduced circuit complexity for signal processing applications", *IEEE Tran. Comput.* vol. 41, no. 10, pp. 1333-1336,Oct. 1992.

[5] M. J. Schulte and E.E. Swartzlander, Jr. "Truncated multiplication with correction constant", *VLSI Signal Processing, VI* New York: IEEE Press, 1993, pp. 338-396.

[6] O. L. MacSorley, "High-speed arithmetic in binary computer", *Proc. IRE*, vol. 49, pp. 67-91, 1961.

[7] S. J. Jou and H. H. Wang, "Fixed-width multiplier for DSP application," *IEEE Int. Conf. Computer Design*, Sep. 2000, pp. 318-322.

[8] E. de Angel and E. E. Swartzlander, Jr., "Low power parallel multipliers," *IEEE VLSI Signal Processing IX*, 1996, pp. 199-208.

[9] M. E. Paul, and K. Bruce, *C Language Algorithms for Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1991.

Table 1: 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
Jou multiplier	4	15	4.69	36.19
	6	85	23.24	846.19
	8	443	107.1	17806.19
	16	524288	68501.62	1686362529
Type 2 with Q=0, *w*=0	4	16	4.59	28.50
	6	85	21.60	716.86
	8	443	103.12	16376.65
	16	524268	62501.62	1486362529
Type 1 with Q=0, *w*=1	4	8	3.28	9.93
	6	41	12.25	175.88
	8	298	79.69	7534.44
	16	394248	42533.53	1186761423

Table 2: 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	28	12	36	$13T_{FA} + 3T_{HA}$
Direct-Truncation Multiplier	11	8	16	$7T_{FA} + 3T_{HA}$
Type 2 with Q=0, *w*=0	16	4	20	$10T_{FA} + T_{HA}$
Type 1 with Q=0, *w*=1	20	4	24	$11T_{FA} + T_{HA}$