PAPER

Adaptive Low-Error Fixed-Width Booth Multipliers

Min-An SONG[†a)], *Student Member*, Lan-Da VAN[††], *and* Sy-Yen KUO[†††], *Nonmembers*

SUMMARY In this paper, we propose two 2's-complement fixed-width Booth multipliers that can generate an n-bit product from an n-bit multiplicand and an n-bit multiplier. Compared with previous designs, our multipliers have smaller truncation error, less area, and smaller time delay in the critical paths. A four-step approach is adopted to search for the best error-compensation bias in designing a multiplier suitable for VLSI implementation. Last but not least, we show the superior capability of our designs by inscribing it in a speech signal processor. Simulation results indicate that this novel design surpasses the previous fixed-width Booth multiplier in the precision of the product. An average error reduction of 65–84% compared with a direct-truncation fixed-width multiplier is achieved by adding only a few logic gates.

key words: *digital signal processing, fixed-width Booth multiplier, VLSI*

1. Introduction

In digital signal processing (DSP) applications (e.g., digital filters and wavelet transformers), it is desirable for the width of arithmetic data to remain fixed throughout the entire computation. To achieve this goal, a fixed-width multiplier [1]–[5] capable of receiving an n-bit multiplier and an n-bit multiplicand and producing an n-bit output is essential. In practice, fixed-width multipliers are based on the CSD algorithm [6], the Baugh-Wooley algorithm or the Booth algorithm. For decades, the Baugh-Wooley fixed-width multipliers have been widely studied. In [8], King and Swartzlander proposed a fixed-width multiplier by analyzing its adaptive error-compensation bias. We generalized the low-error fixed-width multiplier via indexing and *binary thresholding* in [3], [4]. In [5] we have applied the binary thresholding algorithm in the compensating circuit for fixed-width Booth multipliers with small width. In Sect. 3.3 of this paper a statistical technique will be used to verify that the compensating bias is as good for large width too. We have also optimized the compensating circuit, which reduces about one critical path of HA or FA.

The proposed scheme is based on keeping $n + w$ most significant columns of the partial products intact, where w is a nonnegative integer between 0 and $n-1$. When w becomes larger the error is smaller, more gates are required in the compensating circuit. On the contrary, when w is smaller the error becomes larger, fewer gates in the compensating circuit are needed. Therefore, this algorithm allows users to adjust the value of w according to their need in the design of the compensating circuit, and is thus called adaptive. Because an area-time efficient fixed-width multiplier cannot be achieved using the Baugh-Wooley algorithm, most researchers pay attention to the fixed-width Booth algorithm.

The Booth multiplier is widely used in ASIC-oriented products due to the higher computing speed and smaller area. This encoding technique has two advantages: a) only about half of the partial products are needed during the computation, that is, the number of partial products is reduced by a factor of 2; b) delay on the critical path is less than that of the Baugh-Wooley multiplier. Moreover, in a fixed-width Booth multiplier, area saving can be further achieved by truncating n least significant columns and preserving n most significant columns of the partial product. However, since errors are not compensated, significant errors are introduced due to truncation. In [1], a means of reducing the truncation errors was proposed. Unfortunately, it lacks systematic analysis and error estimation.

Therefore, our goal is to introduce a systematic design methodology for low-error area-time efficient Booth multiplier by a) using an error-compensating bias according to a new binary threshold; b) simulating the K value and error performance of the proposed error-compensating bias with various indices; then c) selecting the index resulting in lowest error and satisfying identical K value for limited width of n; and finally; d) constructing a low-error Booth multiplier. Under the limited truncation error, applying this methodology allows additional error-compensating circuit to be easily realized with little area overhead induced by the bias generation circuit. This leads to high-speed area-time efficient multiplication suitable for modern high-performance VLSI and SOC (system on a chip) applications in which area, speed, throughput, and power consumption are critical parameters.

2. Modified Booth Multiplier

Consider the multiplication of two 2's-complement integers, an n-bit multiplicand A and an n-bit multiplier B, as

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

Manuscript received December 1, 2005.
Manuscript revised November 29, 2006.
Final manuscript received March 8, 2007.

[†]The author is with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

[††]The author is with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan.

[†††]The author is with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan.

a) E-mail: dennis@lion.ee.ntu.edu.tw

DOI: 10.1093/ietfec/e90–a.6.1180

where $A = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$, $B = -b_{n-1}2^{n-1} + \sum_{j=0}^{n-2} b_j 2^j$. If n is even, 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 $b_{-1} = 0$. The bracketed term in (2) has a value of $\{-2, -1, 0, 1, 2\}$. Each recoded value performs a certain operation on the multiplicand A; the multiple additions at each stage are required to generate the product. Substituting (2) into (1), we obtain

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

where $S_i = (b_{2i-1} + b_{2i} - 2b_{2i+1}) \cdot A \cdot 2^{2i}$. Triplet scanning takes place from b_{-1} to the MSB with a one-bit overlap, and thus only that is floor $((n+2)/2) - 1$ for signed numbers and defiantly floor $((n+2)/2)$ for unsigned numbers of partial-product rows need to be computed. To simplify the representation of each partial product, we define the notation.

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

where $S_{i,j}$ represents the j-th bit product of the i-th row. In conventional 2's complement Booth arithmetic operations, partial product sign extensions are required for each stage, but these extended sign bits result in large amount of power and area overhead. The sign S in an 8 by 8 multiplier can be expressed as

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

Figure 1 shows the subproduct Booth multiplier for 8 × 8 multiplication according to Eqs. (4) and (5).

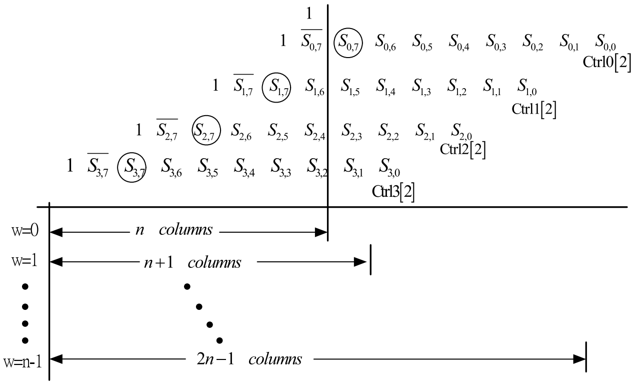
1
1 $\overline{S_{0,7}}$ $S_{0,7}$ $S_{0,6}$ $S_{0,5}$ $S_{0,4}$ $S_{0,3}$ $S_{0,2}$ $S_{0,1}$ $S_{0,0}$
Ctrl0[2]
1 $\overline{S_{1,7}}$ $S_{1,7}$ $S_{1,6}$ $S_{1,5}$ $S_{1,4}$ $S_{1,3}$ $S_{1,2}$ $S_{1,1}$ $S_{1,0}$
Ctrl1[2]
1 $\overline{S_{2,7}}$ $S_{2,7}$ $S_{2,6}$ $S_{2,5}$ $S_{2,4}$ $S_{2,3}$ $S_{2,2}$ $S_{2,1}$ $S_{2,0}$
Ctrl2[2]
1 $\overline{S_{3,7}}$ $S_{3,7}$ $S_{3,6}$ $S_{3,5}$ $S_{3,4}$ $S_{3,3}$ $S_{3,2}$ $S_{3,1}$ $S_{3,0}$
Ctrl3[2]
w=0
n columns
w=1
n+1 columns
w=n-1
2n−1 columns

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

3. Design of Fixed-Width Booth Multiplier

The $2n$-bit product of the n by n 2-complement multiplication can be divided into two sections:

$$P = AB = MP + LP \qquad (6)$$

It is known that the most accurate truncated product is given by

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

$$\sigma_{Temp} = [LP/2^n]_r \qquad (8)$$

where $[\bullet]_r$ denotes the rounding integer of $\bullet$.

Figure 2, without loss of generality we assume $n = 8$, Eq. (8) can be denoted as

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

From Eq. (9), it can be observed that σ_{Temp} is mainly affected by $S_{3,1} + S_{2,3} + S_{1,5} + S_{0,7}$ because the most significant weight is occupied. Thus, it is convenient to define the main-error compensation term E_{main} and the remain-error compensation term E_{remain} as

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

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

From Eqs. (10) and (11), we can rewrite Eq. (8) as

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

It must be emphasized that σ_{Temp} is the most accurate error-compensation bias (also called the true rounding approach). It is apparent that this true rounding approach results in larger area (i.e., higher cost) than the direct-truncted Booth

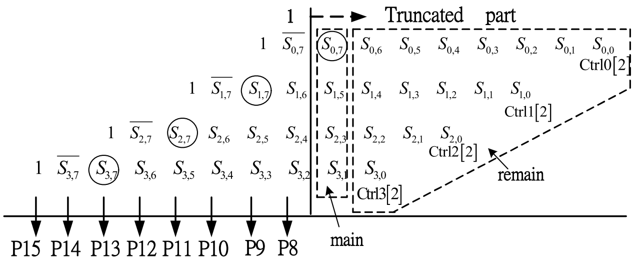
Truncated part
1
1 $\overline{S_{0,7}}$ $S_{0,7}$ $S_{0,6}$ $S_{0,5}$ $S_{0,4}$ $S_{0,3}$ $S_{0,2}$ $S_{0,1}$ $S_{0,0}$
Ctrl0[2]
1 $\overline{S_{1,7}}$ $S_{1,7}$ $S_{1,6}$ $S_{1,5}$ $S_{1,4}$ $S_{1,3}$ $S_{1,2}$ $S_{1,1}$ $S_{1,0}$
Ctrl1[2]
1 $\overline{S_{2,7}}$ $S_{2,7}$ $S_{2,6}$ $S_{2,5}$ $S_{2,4}$ $S_{2,3}$ $S_{2,2}$ $S_{2,1}$ $S_{2,0}$
Ctrl2[2]
remain
1 $\overline{S_{3,7}}$ $S_{3,7}$ $S_{3,6}$ $S_{3,5}$ $S_{3,4}$ $S_{3,3}$ $S_{3,2}$ $S_{3,1}$ $S_{3,0}$
Ctrl3[2]
P15 P14 P13 P12 P11 P10 P9 P8
main

Fig. 2 Modified Booth diagram with "main" and "remain" representing the main and the remaining parts of the (LSB).

multiplier. On the contrary, to obtain the smallest area fixed-width two's-complement multiplier involves truncating the *LP* section directly, but doing so results in large truncation error. Before proposing a new systematic methodology, some useful terminologies are defined first. To explore the influence of the index in the proposed binary *thresholding*, we first define a generalized index. Here the generalized index for 8 by 8 multipliers is defined as

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

where $n + w$ is the number of columns kept in the subproduct, the binary parameters $q_{3-w}, q_{2-w,...,}q_0 \in \{0, 1\}$, 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 binary complement of T. Furthermore, $\theta_{index,w}(q_3, q_2, q_1, q_0)$ is referred to as as $\theta_{Q,w}$, where

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

Note that Q ranges from 0 to $2^{(n/2)-w}-1$; for example, the 9th index is $\theta_{index,w=0}\,(1, 0, 0, 1)$, which is denoted as $\theta_{Q=9,w=0}$.

3.1 Realizable Error-Compensation Bias by Keeping n Most Significant Columns (w=0)

The concept of binary *thresholding* multipliers and methods for choosing their indices have been discussed in [5]. In this work, we propose a generalized methodology to further develop an adaptive multiplier with smaller truncation error. We can rewrite Eq. (12) into

$$\begin{aligned}\sigma_{Temp} &= \theta_{Q,w=0} + \left[\frac{1}{2}E_{main} - \theta_{Q,w=0} + \frac{1}{2}E_{remain}\right]_r \\ &= (< S_{3,1} >^{q_3} + < S_{2,3} >^{q_2} + < S_{1,5} >^{q_1} \\ &\quad + < S_{0,7} >^{q_0}) + [K]_r\end{aligned} \tag{16}$$

Where

$$K = \frac{1}{2}E_{main} - \theta_{Q,w=0} + \frac{1}{2}E_{remain} \tag{17}$$

In Eq. (16), the first term is referred to as the *rough-adjustment term* and the second term, $[K]_r$, is referred to as the fine-adjustment term. The rough-adjustment term can be easily realized in hardware once the index is determined. On the other hand, the fine-adjustment term can be approximated with the expected value in rounding operation after analyzing the statistics [3]. In order to design a simple and realizable error-compensation circuit, we propose two types of binary *thresholding* for error-compensation bias and impose one restriction on the value of K under $w = 0$. Both types of binary *thresholding* of the generalized index, $\theta_{Q,w=0}$ are described as follows:

- Type 1:

$$\sigma_{Type1,w=0} = \begin{cases} (< S_{3,1} >^{q_3} + < S_{2,3} >^{q_2} + < S_{1,5} >^{q_1} \\ + < S_{0,7} >^{q_0}) + [K_1]_r\,, \\ \quad \text{if } \theta_{Q,W=0} = 0 \\ (< S_{3,1} >^{q_3} + < S_{2,3} >^{q_2} + < S_{1,5} >^{q_1} \\ + < S_{0,7} >^{q_0}) + [K_2]_r\,, \\ \quad \text{if } \theta_{Q,W=0} > 0 \end{cases} \tag{18}$$

- Type 2:

$$\sigma_{Type2,w=0} = \begin{cases} (< S_{3,1} >^{q_3} + < S_{2,3} >^{q_2} + < S_{1,5} >^{q_1} \\ + < S_{0,7} >^{q_0}) + [K_3]_r\,, \\ \quad \text{if } \theta_{Q,W=0} = n/2 \\ (< S_{3,1} >^{q_3} + < S_{2,3} >^{q_2} + < S_{1,5} >^{q_1} \\ + < S_{0,7} >^{q_0}) + [K_4]_r\,, \\ \quad \text{if } \theta_{Q,W=0} < n/2 \end{cases} \tag{19}$$

where K_1, K_2, K_3 and K_4 are the average values of K that satisfies $\theta_{Q,w=0} = 0$, $\theta_{Q,w=0} > 0$, $\theta_{Q,w=1} < n/2$ and $\theta_{Q,w=0} = n/2$, respectively. In this subsection, the values of K are restricted so that the chosen indices satisfy $[K_i]_r \in \{-1,\ 0,\ 1\}$ for $i = 1, 2, 3$ and 4. Next, in order to achieve high accuracy in compensation, we investigate on the values of K and the error performance of the generalized index $\theta_{Q,w=0}$. Note in Type 1 binary *thresholding* is a special index of the generalized index $\theta_{Q,w=0}$ when $q_0 = q_1 = \ldots = q_{n-1} = 0$. After performing exhaustive search simulation on Type 1 binary *thresholding* for $n = 8$, we obtained the optimal values of K_1 and K_2 for all possible indices. However, these two indices (i.e., $\theta_{Q=0,w=0}$ and $\theta_{Q=2^{(n/2)-1},w=0}$) in Type 1 binary *thresholding* lead to nonconstant K_1 for $n \leq 16$. This phenomenon can be easily verified by computer simulation and verified by statistical techniques [3] similar to that mentioned in Sect. 3.3. Besides, Type 2 was proven to outperform Type 1 for array multipliers with $w = 0$, and thus we do not take any other approximations for Type 1. Figure 3 illustrates the values of K_3 and K_4 that we have obtained through exhaustive search simulation for $n = 8$. Corresponding to the rounding values of K_3 and K_4, we also simulated the average values, shown in Fig. 4. Considering the goal of smaller error and the restriction on K, we find that the $\theta_{Q=0,w=0}$ and $\theta_{Q=2^{(n/2)-1},w=0}$ indices are of better performance (see Fig. 4. with $[K_3]_r = -1$ and $[K_4]_r = 0$ for $n = 8$.)

Following the above procedures, we can simulate the values of K_3 as well as K_4 and the error performance for n from 4 to 16 in Type 2 binary *thresholding*. After the full-search simulation, we observe that two specific indices $\theta_{Q=0,w=0}$ and $\theta_{Q=2^{(n/2)-1},w=0}$ still achieve better performance where the chosen indices satisfy $[K_3]_r = -1$ and $[K_4]_r = 0$ for $n \leq 16$. Hence, the simple error-compensation biases with smaller truncation error in Type 2 binary *thresholding* are described as

$$\begin{aligned}&\sigma_{Type2,Q=0,w=0} \\ &= \begin{cases} S_{3,1} + S_{2,3} + S_{1,5} + S_{0,7} - 1, & \text{if } \theta_{Q=0,w=0} < n/2 \\ S_{3,1} + S_{2,3} + S_{1,5} + S_{0,7}, & \text{if } \theta_{Q=0,w=0} = n/2 \end{cases}\end{aligned} \tag{20}$$

where $\theta_{Q=0,w=0} = S_{3,1} + S_{2,3} + S_{1,5} + S_{0,7}$ and

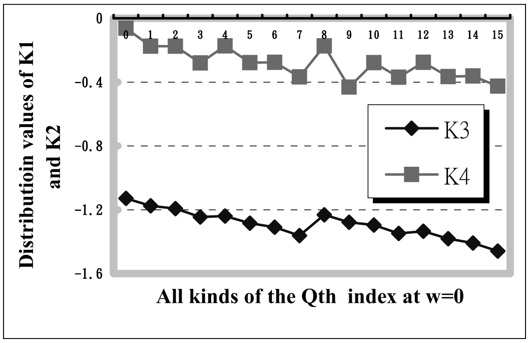

Fig. 3 Values of K_3 and K_4 versus different Q of the binary thresholding for $n = 8$.

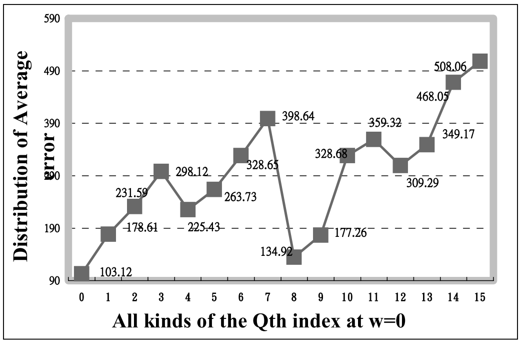

Fig. 4 Average errors by exhaustive search simulation versus different Q.

$$\sigma_{Type2,Q=2^{(n/2)-1},w=0} = \begin{cases} \overline{S_{3,1}} + S_{2,3} + S_{1,5} + S_{0,7} - 1, \\ \text{if } \theta_{Q=2^{(n/2)-1},W=0} < n/2 \\ \overline{S_{3,1}} + S_{2,3} + S_{1,5} + S_{0,7}, \\ \text{if } \theta_{Q=2^{(n/2)-1},W=0} = n/2 \end{cases} \tag{21}$$

where $\theta_{Q=0,w=0} = \overline{S_{3,1}} + S_{2,3} + S_{1,5} + S_{0,7}$.

Equation (20) has been completely simulated for $n \leq 16$ and can be mapped to a new structure.

3.2 Realizable Error-Compensation Bias by Keeping More than n Columns ($w \geq 1$)

By lower truncation error can be obtained if larger $n+w$ most significant columns are kept in hardware. However, the area cost could be increased. Since the reduction and rounding errors are not of the same weight, Eq. (12) can be rewritten as

$$\sigma_{Temp} \cong \left[\frac{1}{2}E_{main} + \frac{1}{2}E_{remain} + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} + E_{reduct,w} - \frac{1}{2^w}\theta_{Q,w}\right]_r \cong \left\lfloor \frac{1}{2}E_{main} + \frac{1}{2}E_{remain} + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} + E_{reduct,w} - \frac{1}{2^w}\theta_{Q,w} + E_{round,w} \right\rfloor$$

$$\cong \left\lfloor \left(\frac{1}{2}E_{main} + \frac{1}{2}E_{remain} + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} + [K]_r/2^w \right) \right\rfloor \tag{22}$$

Where

$$K = 2^w \left(E_{reduct,w} - \frac{1}{2^w}\theta_{Q,w} + E_{round,w} \right) \tag{23}$$

$$E_{reduct,w} = \frac{1}{2^{w+1}}\left(S_{3,1-w} + S_{2,3-w} + S_{1,5-w} + S_{0,7-w}\right) + \ldots \ldots + \frac{1}{2^n}S_{0,0} + \left(\frac{1}{2^2}Ctrl_{3,1-w}[2] + \frac{1}{2^4}Ctrl_{2,3-w}[2] + \frac{1}{2^6}Ctrl_{1,5-w}[2] + \frac{1}{2^8}Ctrl_{0,7-w}[2] \right) \tag{24}$$

$$E_{round,w} = 2^{-1}\left(1 - 2^{-w}\right) \tag{25}$$

$\lfloor \bullet \rfloor$ denotes the maximum integer equal to or less than $\bullet$. Treating reduction and rounding errors in Eq. (23). In the same way, to design a realizable error-compensation bias, two types of binary *thresholding* for the error-compensation bias can be changed to

- Type 1:

$$\sigma_{Type1,Q=0,w=1} = \begin{cases} \left\lfloor \frac{1}{2}\left(E_{main} + E_{remain}\right) + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} + [K_1]_r/2^w \right\rfloor, \\ \text{if } \theta_{Q,w} = 0 \\ \left\lfloor \frac{1}{2}\left(E_{main} + E_{remain}\right) + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} + [K_2]_r/2^w \right\rfloor, \\ \text{if } \theta_{Q,w} > 0 \end{cases} \tag{26}$$

- Type 2:

$$\sigma_{Type2,Q=0,w=1} = \begin{cases} \left\lfloor \frac{1}{2}\left(E_{main} + E_{remain}\right) + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} + [K_3]_r/2^w \right\rfloor, \\ \text{if } \theta_{Q,w} < n/2 \\ \left\lfloor \frac{1}{2}\left(E_{main} + E_{remain}\right) + \frac{1}{2^w}\theta_{Q,w} - E_{reduct,w} + [K_4]_r/2^w \right\rfloor, \\ \text{if } \theta_{Q,w} = n/2 \end{cases} \tag{27}$$

where K_1, K_2, K_3 and K_4 are defined as those of Eqs. (18) and (19) except for $w = 1$. The restriction of K can be modified as $[K_i]_r \in \left\{0, 1, 2^{w-1} - 1, 2^{w-1}\right\}$ for $i = 1, 2, 3$ and 4. For w=1, with identical simulation procedures as mentioned in Sect. 3.1, we introduce only the analysis and design for w=1. In Type 1 binary *thresholding*, by exhaustive search we can find one good index, as shown in Fig. 5. We observe that the specific index, $\theta_{Q=0,w=1}$, achieves best error performance given that $[K_1]_r = 1$ and $[K_2]_r = 0$, as shown in Fig. 6. On the other hand, for Type 2 binary *thresholding*, all the average errors are larger than those resulted from the best index in Type 1 *thresholding*, $\theta_{Q=0,w=1}$. Therefore, we do not need to discuss Type 2. So far, the second step is processed. As a consequence, a new smaller error fixed-width Booth multiplier under w=1 can be described and simplified as:

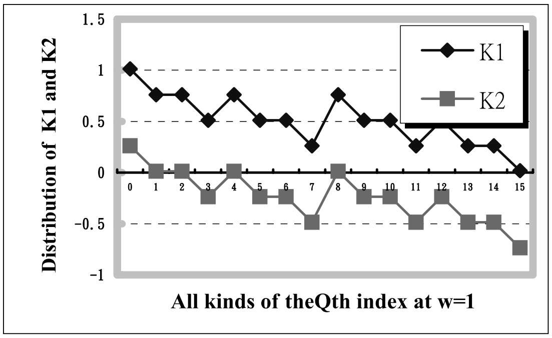

Fig. 5 K1 and K2 versus binary thresholding Q.

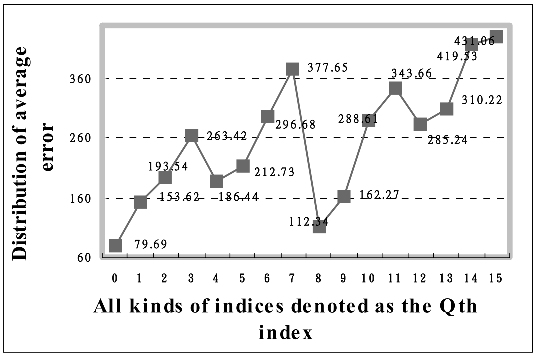

Fig. 6 Average errors by exhaustive search simulation versus the binary threshoding Q.

$$
\begin{aligned}
&\sigma_{Type1,Q=0,w=1}\\
&=\begin{cases}\left\lfloor \frac{1}{2}\left(E_{main}+E_{remain}+\theta_{Q=0,w=1}\right)-E_{reduct,w=1}+1/2\right\rfloor,\\ \text{if } \theta_{Q=0,w=1}=0\\ \left\lfloor \frac{1}{2}\left(E_{main}+E_{remain}+\theta_{Q=0,w=1}\right)-E_{reduct,w=1}+0\right\rfloor,\\ \text{if } \theta_{Q=0,w=1}>0\end{cases}
\end{aligned}
\tag{28}
$$

Because $\frac{1}{2}E_{remain}=E_{reduct,w=1}$, Eq. (29) can be simplified as

$$
\begin{aligned}
&\sigma_{Type1,Q=0,w=1}\\
&=\begin{cases}\left\lfloor \frac{1}{2}\left(E_{main}+\theta_{Q=0,w=1}\right)+\frac{1}{2}\right\rfloor, \text{ if } \theta_{Q=0,w=1}=0\\ \\ \left\lfloor \frac{1}{2}\left(E_{main}+\theta_{Q=0,w=1}\right)+0\right\rfloor, \text{ if } \theta_{Q=0,w=1}>0\end{cases}
\end{aligned}
\tag{29}
$$

where $\theta_{Q=0,w=1}=S_{3,0}+S_{2,2}+S_{1,4}+S_{0,6}$. In the third step, Eq. (29) can be mapped to a new structure as shown in Fig. 7. Note that the error-compensation circuit only needs three basic gates. For other w, we can evaluate K with the same procedures.

3.3 Low-Error Fixed-Width Booth Multipliers with Large Width n

By full search simulation, we find that $\theta_{Q=0,w=0}$ and $\theta_{Q=2^{(n/2)-1},w=0}$ in Type 2 binary *thresholding* achieve better performance for small width n. Similarly, we observe that $\theta_{Q=0,w=1}$ in Type 1 binary *thresholding* leads to better results. It is difficult to simulate the performance for large width n since the exhaustive simulation takes lots of computation time. In this section, we show that these indices can be adopted in designing fixed-width Booth multiplier for larger width n. That is, we verify $[K_3]_r=-1$ and $[K_4]_r=0$ for $w=0$ in Type 2 binary *thresholding* and $[K_1]_r=1$ and $[K_2]_r=0$ for $w=1$ in Type 1 binary *thresholding* for large width n by statistical techniques. The input bits are assumed to be uniformly distributed, so we can approximate $\frac{1}{2}E_{main}$, $\frac{1}{2}E_{remain}$ and other terms with expected output values from logic functions. We verify Eq. (29) with $\theta_{Q=0,w=1}$ in Type1 binary *thresholding* for large width n. The statistical techniques can be used to verify values of K_1 and K_2. Two cases are taken into consideration: $\theta_{Q=0,w=1}=0$ and $\theta_{Q=0,w=1}>0$

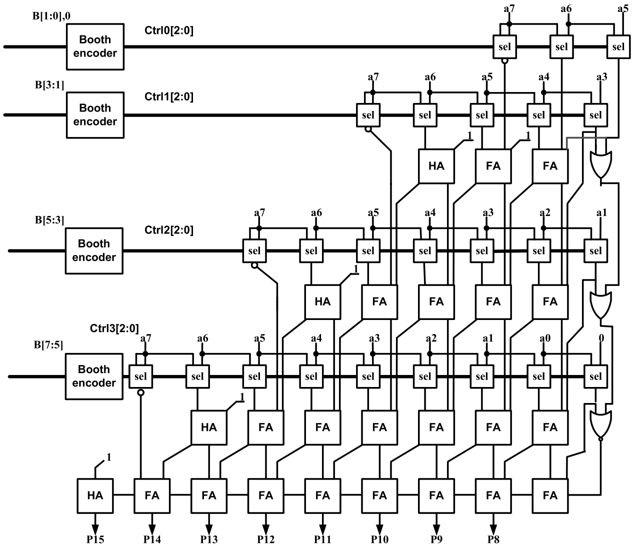

Fig. 7 The proposed fixed-width 8 × 8 Booth multiplier with $\theta_{Q=0,w=1}$.

- Case 1: $\theta_{Q=0,w=1}=0$

Note that $E\left\{S_{i,j}\right\}=3/8$ and $E\left\{\overline{S_{i,j}}\right\}=5/8$ since the probability distribution of input bits is assumed to be uniform. In this case, $\theta_{Q=0,w=1}=0$ is met only when $S_{3,1}=S_{2,3}=S_{1,5}=S_{0,7}=0$. That is, Case 1 is a conditional probability case, and thus we deduce Eq. (30) from Eq. (23) as

$$
\begin{aligned}
K_1 &= E\left\{2\left(E_{reduct,w=1}-\frac{1}{2^w}\theta_{Q=0,w=1}+E_{round,w=1}\right)\right\}\\
&= 2\times\left(\frac{3}{8}\left(\left(\frac{1}{2^3}+\frac{1}{2^4}\right)(n-1)+\left(\frac{1}{2^5}+\frac{1}{2^6}\right)(n-2)\right.\right.\\
&\qquad\left.\left.+\left(\frac{1}{2^7}+\frac{1}{2^8}\right)(n-3)\right)+\frac{1}{4}\right)\\
&= \frac{3}{4}\left(\frac{63n-81}{256}\right)+\frac{1}{2} \quad \text{if } n\geq 4
\end{aligned}
\tag{30}
$$

Note that K_1 is proportional to $63n/256$, so this error-compensation circuit is difficult to design for $n\geq 4$. We observe the fact that Case 1 is a minor case for all input combinations and the rounding value of K_1 is always equal to or greater than one. In order to design a simple error-compensation circuit based on the above two facts, we adopt a constant to approximate the value of K_1. Equation (30) can be approximately set to

$$K_1 \cong 1 \tag{31}$$

Substituting Eq. (31) into Eq. (26), we obtain

$$\sigma_{Type1,Q=0,w=1} = \begin{cases} \left\lfloor \frac{1}{2}\left(E_{main}+E_{remain}\right)+\frac{1}{2^w}\theta_{Q=0,w=1}-E_{reduct,w=1}+1/2\right\rfloor, \\ \text{if } \theta_{Q=0,w=1}=0 \end{cases} \tag{32}$$

- Case 2: $\theta_{Q=0,w=1} > 0$

From Eq. (23), we obtain Eq. (33) by statistical techniques as

$$\begin{aligned} K_2 &= E\left\{2\left(E_{reduct,w=1}-\frac{1}{2^w}\theta_{Q=0,w=1}+E_{round,w=1}\right)\right\} \\ &= 2\times\left(\frac{3}{8}\left(\frac{1}{2}n+\left(\frac{1}{2^3}+\frac{1}{2^4}\right)(n-1)+\left(\frac{1}{2^5}+\frac{1}{2^6}\right)(n-2)\right.\right. \\ &\qquad \left.\left.+\left(\frac{1}{2^7}+\frac{1}{2^8}\right)(n-3)-\frac{1}{2}n\right)+\frac{1}{4}\right) \\ &= \frac{3}{4}\left(\frac{-n-81}{256}\right)+\frac{1}{2} \quad \text{if } n \geq 4 \end{aligned} \tag{33}$$

Substituting Eq. (33) into Eq. (26), we obtain

$$\sigma_{Type1,Q=0,w=1} = \begin{cases} \left\lfloor \frac{1}{2}\left(E_{main}+E_{remain}\right)+\frac{1}{2^w}\theta_{Q=0,w=1}-E_{reduct,w=1}+0/2\right\rfloor, \\ \text{if } \theta_{Q=0,w=1}>0 \end{cases} \tag{34}$$

By combining Eqs. (32) and (34), we obtain Eq. (28) and simplify it as Eq. (29). The chosen index $\theta_{Q=0,w=1}$ is suitable for error-compensation bias for large width n. Similarly, for $w \geq 2$, the statistical verification techniques and constant approximation can be applied. We can conclude that the error-compensation bias is

$$\sigma_{Type1,Q=0,w\geq1} = \begin{cases} \left\lfloor \frac{1}{2}\left(E_{main}+E_{remain}+\theta_{Q=0,w\geq1}\right)-E_{reduct,w\geq1}+1/2\right\rfloor, \\ \text{if } \theta_{Q=0,w\geq1}=0 \\ \left\lfloor \frac{1}{2}\left(E_{main}+E_{remain}+\theta_{Q=0,w\geq1}\right)-E_{reduct,w\geq1}+\frac{1}{2}-\frac{1}{2^w}\right\rfloor, \\ \text{if } \theta_{Q=0,w\geq1}>0 \end{cases} \tag{35}$$

4. Performance and Area Comparisons

The new structure can achieve better error performance than other multipliers. Compared with J-T-T's multiplier in [1], our multiplier (Type 2, w=0) outstands in average error. When compared with C-L-P's multiplier in [2], our multiplier (Type 1, w=2) has less gates and shorter critical path and delay time as listed in Table 1. The simulation results of the fixed-width Booth multipliers that varies in width n as listed in Table 2. Our proposed fixed-width Booth multiplier is of Type 2 *thresholding* with the index $\theta_{Q=0,w=0}$ and of Type 1 *thresholding* with the index $\theta_{Q=0,w=2}$. Clearly the proposed fixed-width Booth multiplier is more accurate than others. The improved performance is achieved by applying a better error-compensation bias to reduce the effect of truncation error.

Table 1 Comparison results of area and critical delay among various Booth multipliers and [3], [4] for n = 8.

Multiplier	Area (# of gates)		Critical Delay Time
	FA	HA	
Full-Precision Multiplier	28	12	$13T_{FA}+3T_{HA}$
Direct-Truncation Multiplier	12	7	$8T_{FA}+2T_{HA}$
J-T-T's Multiplier	16	4	$10T_{FA}+T_{HA}$
C-L-P's Multiplier	23	6	$11T_{FA}+2T_{HA}$
Type 1 with Q=0, w=2 in [3,4]	43	28	$18T_{FA}+2T_{HA}$
Type 2 with Q=0, w=0	16	4	$10T_{FA}+T_{HA}$
Type 1 with Q=0, w=1	20	4	$11T_{FA}+1T_{HA}$
Type 1 with Q=0, w=2	22	6	$11T_{FA}+1T_{HA}$

Table 2 Comparison results of four types of errors among various Booth multipliers and [3],[4].

Multiplier	Wid.	Maximum E.	Average E.	Variance of E.
J-T-T's Multiplier	6	85	23.24	746.19
	8	443	107.10	6,806
	10	2181	477.89	73,166.19
	12	10,363	2,083.53	1,042,92
C-L-P's Multiplier	6	64	19.27	184.5
	8	256	69.99	2,761
	10	1,280	316.10	42,186
	12	5,120	1,305.18	843,218.25
Type 1 with Q=0, w=2 in [3,4]	6	49	16.29	110.32
	8	237	69.15	2597.87
	10	1,117	292.27	39,345
	12	5,149	1,205.49	723,507
Type 2 with Q=0, w=0	6	83	21.56	339.67
	8	433	103.12	4,478.43
	10	2,153	473	56,674.87
	12	10,122	1,989.88	912,872.65
Type 1 with Q=0, w=2	6	48	17.52	144.5
	8	213	69.57	2,667.23
	10	1,211	313.99	40,384.21
	12	5,253	1,215.37	790,916.71

Besides, the new Booth multiplier also saves much chip area and performs better with much smaller average error. Most importantly, the gate count and the critical delay of the proposed structure are better those of J-T-T's and C-L-P's multiplier. Based on the circuitry architecture of the Booth multiplier we have designed, by using our compensation circuit of (Type 1, w=2) and that of (ACGP 2) as provided by C-L-P respectively. Thus, the proposed fixed-width 8 × 8 Booth multiplier shown in Fig. 8 has the combined advantage of area-time efficiency and excellent error performance.

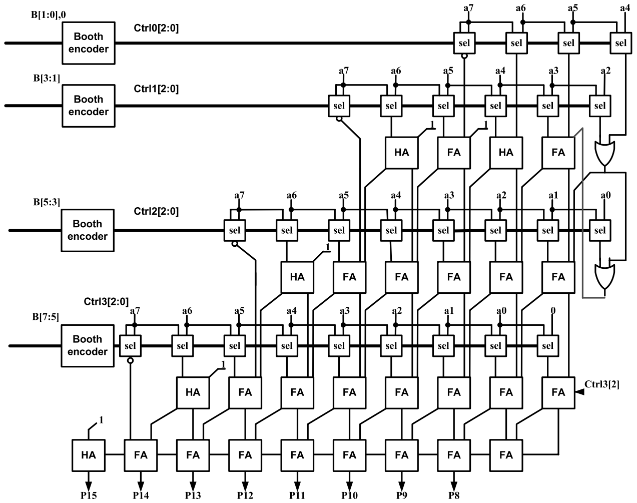

Fig. 8 The proposed fixed-width 8 × 8 Booth multiplier with $\theta_{Q=0,w=2}$.

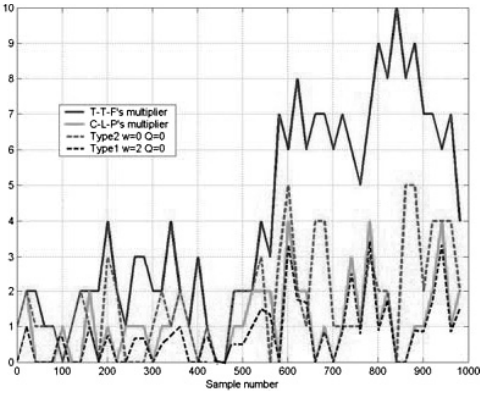

Fig. 9 Comparison results of error signals obtained with four kinds of Booth multipliers.

5. DSP Application of Fixed-Width Booth Multipliers

In this section, we apply the proposed fixed-width multiplier to the 35-tap FIR filter for speech processing. First, for practical consideration [7], the maximum input voice data and filter coefficient in two's complement are normalized with 8-bit quantization. In the simulation, the temporary output is an accumulated value using 32 bits. Finally, the outputs are then obtained by scaling the accumulated values. For convenience of comparison among various fixed-width Booth multipliers, we take 1000 samples for the consonant part and the vowel part of "Chicken." What we are concerned here is whether the filtered waveform is accurate via the proposed fixed-width Booth multiplier in which the precise standard output is the controlled group. The filtered output signals are processed by the 35-tap low-pass FIR filter with different fixed-width Booth multipliers. From the comparison results in Fig. 9 on four fixed-width Booth multipliers in speech processing application, we observed that the Type1 Booth multiplier with $\theta_{Q=0.w=2}$ satisfies our requirement both in the consonant and the vowel parts.

6. Conclusions

This paper proposed a new methodology for designing low-error two's-complement fixed-width Booth multipliers. By properly choosing the generalized index, we derive a better error-compensation bias than previous works to reduce the maximum error, the average error and the variance of error, and improve the truncation error. Furthermore, these error-compensation biases can be easily implemented. It is very suitable for VLSI digital signal processing applications where the accuracy, area, and speed issues are crucial. Besides, these Booth multipliers can be easily applied to computing engines such as digital filters and wavelet transform. Finally, we successfully apply the proposed fixed-width Booth multiplier to a digital FIR filter for speech processing application. It has been shown that the performance of our multiplier for the consonant part is far better than that of existing fixed-width Booth multipliers. The future works include the study of other binary *thresholding* with our generalized index and the restriction on K to design more useful and realizable fixed-width Booth multipliers.

Acknowledgments

This research was supported by the Ministry of Economics, Taiwan under contract number 92-EC-17-A-08-S1-0006 and by the National Science Council, Taiwan under Grant NSC 95-2221-E-002-068.

References

[1] S.J. Jou, M.H. Tsai, and Y.L. Tsao, "Low-error reduced-width Booth multiplier for DSP application," IEEE Trans. Circuits Syst. I, vol.50, pp.1470–1474, Nov. 2003.

[2] K.J. Cho, K. C Lee, J.G. Chung, and K.K. Parhi, "Design low-error fixed-width mondified Booth multiplier," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.12, no.5, pp.522–531, May 2004.

[3] 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.

[4] L.D. Van and S.H. Lee, "A generalized methodology for lower-error area-efficient fixed-width multipliers," Proc. IEEE Int. Symp. Circuits Syst., vol.1, pp.65–68, Phoenix, Arizona, May 2002.

[5] M.A. Song, L.D. Van, T.C. Huang, and S.Y. Kuo, "Framework for the design of error-aware power-efficient fixed-width Booth multipliers?," IEEE Int. Symp. Circuits and Syst., Kobe, Japan, May 2005.

[6] S.M. Kim, J.G. Chung, and K.K. Parhi, "Low-error fixed-width CSD multiplier with efficient sign extension," IEEE Trans. Circuits Syst. II, vol.50, pp.984–993, Dec. 2003.

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

[8] E.E. Swartzlander, Jr., "Truncated multiplication with approximate rounding?," Proc. 33rd Asilomar Conference on Signals, Systems, and Computers, vol.2, pp.1480–1483, 1999.

[9] M.A. Song, L.D. Van, T.C. Huang, and S.Y. Kuo, "A generalized methodology for low-error and area-time efficient fixed-width booth multipliers," IEEE Int. Midwest Symp. Circuits Syst., vol.47, pp.I9–I12, July 2004.

[10] UMC 0.18 μm Process 1.8 V SAGE-X$^{\text{TM}}$Standard Cell Library Databook, Release 1.1, Artisan Components, Feb. 2002.

Min-An Song was born in 1966 in Taipei, Taiwan. He received the B.S. degree in Electrical Engineering form St. John's University in 1983 and 1988 and M.S. degree in Electrical Engineering from National Taiwan University in 1998 and 2000. He is currently working toward the Ph.D. degree in Electrical Engineering from National Taiwan University. His research interests include high-speed and low-power VLSI architectures and algorithms for digital signal processing and communication applications with the emphasis on digital filters and computer arithmetic, parallel and distributed computing, embedded systems, and intelligent systems.

Lan-Da Van was born in Miaoli, Taiwan, R.O.C., in 1972. He received the B.S. (Honors) and the M.S. degrees from Tatung Institute of Technology, Taipei, Taiwan, in 1995 and 1997, respectively, and the Ph.D. degree from National Taiwan University (NTU), Taipei, Taiwan, in 2001, all in electrical engineering. From 2001 to 2006, he was he was an Associate Researcher at National Chip Implementation Center (CIC), Hsinchu, Taiwan. Since Feb. 2006, he joined the faculty of Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, where he is currently an Assistant Professor. His research interests are in VLSI algorithms, architectures, and chips for digital signal processing (DSP), baseband communication systems, and computer applications. This includes the design of high-performance /low-power /area-aware DSP processors, adaptive filters, transform, computer arithmetic, and platform-based system-on-a-chip (SOC) designs.

Sy-Yen Kuo is Dean of the College of Electrical and Computer Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. He is also a Professor at the Department of Electrical Engineering, National Taiwan University where he is currently on leave and was the Chairman at the same department from 2001 to 2004. He received the BS (1979) in Electrical Engineering from National Taiwan University, the MS (1982) in Electrical & Computer Engineering from the University of California at Santa Barbara, and the Ph.D. (1987) in Computer Science from the University of Illinois at Urbana-Champaign. He spent his sabbatical years as a Visiting Professor at the Computer Science and Engineering Department, the Chinese University of Hong Kong from 2004–2005 and as a visiting researcher at AT&T Labs-Research, New Jersey from 1999 to 2000, respectively. He was the Chairman of the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan from 1995 to 1998, a faculty member in the Department of Electrical and Computer Engineering at the University of Arizona from 1988 to 1991, and an engineer at Fairchild Semiconductor and Silvar-Lisco, both in California, from 1982 to 1984. In 1989. Professor Kuo is an IEEE Fellow. He has published more than 260 papers in journals and conferences. He received the distinguished research award between 1997 and 2005 consecutively from the National Science Council in Taiwan and is now a Research Fellow there. He was also a recipient of the Best Paper Award in the 1996 International Symposium on Software Reliability Engineering, the Best Paper Award in the simulation and test category at the 1986 IEEE/ACM Design Automation Conference (DAC), the National Science Foundation's Research Initiation Award in 1989, and the IEEE/ACM Design Automation Scholarship in 1990 and 1991.