Design of the Lower Error Fixed-Width Multiplier and Its Application

Lan-Da Van, Shuenn-Shyang Wang, and Wu-Shiung Feng

***Abstract*—This brief develops a general methodology for designing a lower-error two's-complement fixed-width multiplier that receives two n-bit numbers and produces an n-bit product. By properly choosing the generalized index, we derive the better error-compensation bias to reduce the truncation error and then construct a lower error fixed-width multiplier, which is area efficient for VLSI implementation. Finally, we successfully apply the proposed fixed-width multiplier to realizing a digital FIR filter, which has shown that the performance is better than that using other fixed-width multipliers.**

***Index Terms*—Area-efficient, fixed-width multiplier, speech processing, truncation error.**

I. Introduction

Low-error, small-area, and high-speed multipliers are the most important processing element for digital signal processing (DSP) applications [1] such as digital filters [2], [3], Moving Picture Experts Group (MPEG) coding, and so on. The multipliers based on the Baugh–Wooley algorithm [4], [5] produce $2n$-bit output with n-bit multiplier and n-bit multiplicand input. However, for some practical applications, we only require n-bit multiplication output, which may be obtained by directly truncating the n least-significant bits and preserving the n most significant bits. However, by this way, significant errors introduced in the fixed-width operation are undesirable for many DSP applications. To reduce the introduced truncation error, Kidambi et al. [6] proposed the bias compensation structure derived from the statistics of carry propagation, but this structure did not adaptively adjust the proper bias by taking account of a variety of input signals. Next, Jou *et al.* [7] provided the carry-generating circuit to improve the truncation error corresponding to J-Ks' index. However, there exist two problems that have never been discussed before. One is how to choose proper indices, and the other is whether other lower error multipliers exist or not. The work proposes the general methodology for designing the lower error two's-complement fixed-width multiplier. In addition, this new multiplier has the same area-ratio as J-Ks' multiplier under reasonable assumption. This brief is organized as follows. In Section II, we propose a better error-compensation bias to reduce the truncation error by properly choosing the generalized index, as well as binary thresholding and then construct a simply lower error fixed-width multiplier. In Section III, we guarantee that this compensation bias still be held for large width n. The performance comparison results in terms of maximum error, average error, variance of the error, and area ratio are discussed in Section IV. In Section V, we apply the proposed lower error fixed-width multiplier to a low-pass FIR digital filter [8], and it can be shown that the performance is better than that using other fixed-width multipliers. At last, short statements conclude the representation of this brief.

II. Design of Fixed-Width Multipliers

Considering two two's-complement integer operands, a n-bit multiplicand X and a n-bit multiplier Y can be, respectively, represented by

$$X = -x_{n-1}2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i \tag{1}$$

$$Y = -y_{n-1}2^{n-1} + \sum_{j=0}^{n-2} y_j 2^j \tag{2}$$

where $x_i, y_i \in \{0, 1\}$. Their product P_{Standard} can be written as

$$\begin{aligned} P_{\mathrm{Standard}} &= X \times Y \\ &= x_{n-1}y_{n-1}2^{2n-2} + \sum_{i=0}^{n-2}\sum_{j=0}^{n-2} x_i y_j 2^{i+j} \\ &\quad + 2^{n-1}\left(-2^{n-1} + \sum_{j=0}^{n-2} \overline{x_{n-1}y_j} 2^j + 1\right) \\ &\quad + 2^{n-1}\left(-2^{n-1} + \sum_{i=0}^{n-2} \overline{y_{n-1}x_i} 2^i + 1\right). \end{aligned} \tag{3}$$

Equation (3) is the famous Baugh–Wooley array multiplier [4], [5], in which this algorithm combines partial products with the same weighting factor and places them in the same column. Fig. 1 shows the subproduct array for 8×8 multiplication. According to (3), an 8×8 standard multiplier structure can be obtained as shown in Fig. 2(a) in which its main symbolic cells are depicted in Fig. 2(b) and other cells A, ND, HA, and FA denote an AND gate, a NAND gate, a half adder, and a full adder, respectively. By partitioning the subproducts into two sections, (3) can be rewritten as follows:

$$\begin{aligned} P_{\mathrm{Standard}} &= MP + LP \\ &= \sum_{i=n}^{2n-1} P_i 2^i + \sum_{i=0}^{n-1} P_i 2^i \end{aligned} \tag{4}$$

where $P_i \in \{0, 1\}$, $MP = \sum_{i=n}^{2n-1} P_i 2^i$ is the most-significant section, and $LP = \sum_{i=0}^{n-1} P_i 2^i$ is the least-significant section as shown in the upper right triangular area of Fig. 2(a). It is well known that the simplest fixed-width multiplier is to directly truncate LP section, but this approach leads to the largest truncation error. So, Kidambi *et al.* [6] provided a constant bias method, which was derived from the carry propagation probability of LP. The truncated multiplier presented in [6] yields the approximate n-bit fixed-width product $P_{\mathrm{K\text{-}G\text{-}A}}$; that is

$$P_{\mathrm{Standard}} \cong P_{\mathrm{K\text{-}G\text{-}A}} = MP + \sigma_{\mathrm{K\text{-}G\text{-}A}} \times 2^n \tag{5}$$

where $\sigma_{\mathrm{K\text{-}G\text{-}A}}$ represents the error-compensation bias depending on the width n. While the width n is given, the error-compensation bias $\sigma_{\mathrm{K\text{-}G\text{-}A}}$ is a constant under the uniform probability distribution of input bits. Although this approach compensates more information than the simplest truncated multiplier, the bias cannot be adaptively adjusted for different input signals. Thus, the truncation error is still large. Jou *et al.* [7] presented another way to analyze the error compensation, and suggested a truncated multiplier which results in the fixed-width product $P_{\mathrm{J\text{-}K}}$ as in (6) and (7), shown at the bottom of the next page, where the index $\theta_{\mathrm{J\text{-}K}} = x_{n-1}y_0 + x_{n-2}y_1 + \cdots + x_0 y_{n-1}$. Note that the index $\theta_{\mathrm{J\text{-}K}}$, which is a function of input signals X and Y, determines the error compensation bias $\sigma_{\mathrm{J\text{-}K}}$. Though the bias $\sigma_{\mathrm{J\text{-}K}}$

Manuscript received February 1999; revised June 2000. This work is supported by National Science Council under Contact Number NSC: 88-2216-E-002-018. This paper was recommended by Associate Editor P. Cheung.

L.-D. Van is with the Department of Electrical Engineering, National Taiwan University, Taipei, 106 Taiwan, R.O.C. (e-mail: d86029@cad.ee.ntu.edu.tw).

S.-S. Wang is with the Department of Electrical Engineering, Tatung University, Taipei, Taiwan, R.O.C. (e-mail: sswang@ttu.edu.tw)

W.-S. Feng is with the Department of Electronics Engineering, Chang Gung University, Taoyuan, Taiwan, R.O.C.

Publisher Item Identifier S 1057-7130(00)09334-4.

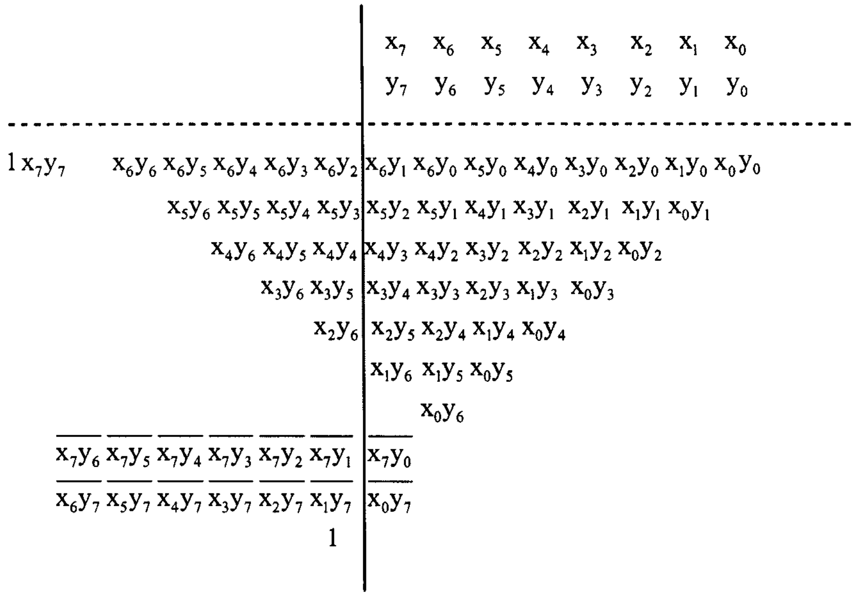
x7 x6 x5 x4 x3 x2 x1 x0
y7 y6 y5 y4 y3 y2 y1 y0
1 x7y7 x6y6 x6y5 x6y4 x6y3 x6y2 x6y1 x6y0 x5y0 x4y0 x3y0 x2y0 x1y0 x0y0
x5y6 x5y5 x5y4 x5y3 x5y2 x5y1 x4y1 x3y1 x2y1 x1y1 x0y1
x4y6 x4y5 x4y4 x4y3 x4y2 x3y2 x2y2 x1y2 x0y2
x3y6 x3y5 x3y4 x3y3 x2y3 x1y3 x0y3
x2y6 x2y5 x2y4 x1y4 x0y4
x1y6 x1y5 x0y5
x0y6
x7y6 x7y5 x7y4 x7y3 x7y2 x7y1 x7y0
x6y7 x5y7 x4y7 x3y7 x2y7 x1y7 x0y7
1

Fig. 1. Subproduct array of 8×8 multiplication.

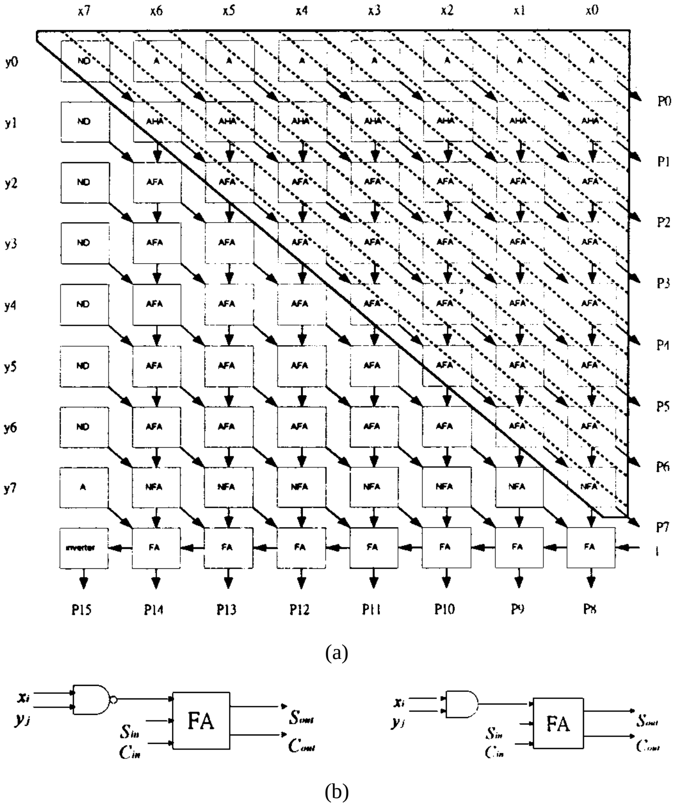

Fig. 2. (a) Block diagram of the standard multiplier. (b) NFA gate on left and AFA gate on right.

performs better than the bias $\sigma_{\text{K-G-A}}$, it is deeply expected to develop a generalized methodology to further improve truncation error. It is because that there exist two never discussed problems in [7]: how to choose proper indices and whether other lower-error multipliers exist or not.

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

$$P_{\text{Standard}} \cong MP + \sigma_{\text{Temp}} \times 2^n, \tag{8}$$

$$\begin{aligned}\sigma_{\text{Temp}} &= [LP]_r \\ &= \Bigg[\frac{1}{2}(\overline{x_{n-1}y_0} + x_{n-2}y_1 + \cdots + x_1y_{n-2} \\ &\quad + \overline{x_0y_{n-1}}) + \frac{1}{2^2}(x_{n-2}y_0 + \cdots + x_0y_{n-2}) \\ &\quad + \cdots + \frac{1}{2^{n-1}}(x_1y_0 + x_0y_1) + \frac{1}{2^n}x_0y_0\Bigg]_r \end{aligned} \tag{9}$$

where $[t]_r$ represents the rounding integer for t. It should be emphasized that σ_{Temp} is an ideal error-compensation term and it is infeasible to implement the truncated fixed-width multiplier without using any acceptable approximation. From (9), it is observed that σ_{Temp} is mainly affected by $\overline{x_{n-1}y_0} + x_{n-2}y_1 + \cdots + x_1y_{n-2} + \overline{x_0y_{n-1}}$ due to the largest weight. Now, let us define the main-error compensation term E_{main} and the remain-error compensation term E_{remain}, respectively, as

$$E_{\text{main}} \triangleq \overline{x_{n-1}y_0} + x_{n-2}y_1 + \cdots + x_1y_{n-2} + \overline{x_0y_{n-1}}, \tag{10}$$

$$\begin{aligned}E_{\text{remain}} &\triangleq \frac{1}{2}(x_{n-2}y_0 + x_{n-3}y_1 + \cdots + x_0y_{n-2}) \\ &\quad + \cdots + \frac{1}{2^{n-1}}x_0y_0. \end{aligned} \tag{11}$$

$$\begin{aligned}P_{\text{Standard}} &\cong P_{\text{J-K}} \\ &= MP + \sigma_{\text{J-K}} \times 2^n \end{aligned} \tag{6}$$

$$\sigma_{\text{J-K}} = \begin{cases}(x_{n-2}y_1 + x_{n-3}y_2 + \cdots + x_2y_{n-3} + x_1y_{n-2}) + 1, & \text{if } \theta_{\text{J-K}} = 0 \\ x_{n-2}y_1 + x_{n-3}y_2 + \cdots + x_2y_{n-3} + x_1y_{n-2}, & \text{if } \theta_{\text{J-K}} > 0\end{cases} \tag{7}$$

Thus, we can rewrite (9) as

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

Note that σ_{Temp} varies while the input bits x_i's or y_i's alternates. For general analyzing, we first define a generalized index θ_{index} as

$$\begin{aligned}&\theta_{\text{index}}(q_{n-1}, q_{n-2}, \ldots, q_0)\\ &\quad \triangleq \langle x_{n-1}y_0\rangle^{q_{n-1}} + \langle x_{n-2}y_1\rangle^{q_{n-2}} + \cdots + \langle x_0 y_{n-1}\rangle^{q_0}\end{aligned} \tag{13}$$

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

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

in which $\bar{T}$ is the complement of binary T. By utilizing (13), one can rewrite (12) as

$$\begin{aligned}\sigma_{\text{Temp}} &= \theta_{\text{index}} + \left[\frac{1}{2}E_{\text{main}} - \theta_{\text{index}} + \frac{1}{2}E_{\text{remain}}\right]_r\\ &= (\langle x_{n-2}y_1\rangle^{q_{n-2}} + \cdots + \langle x_1 y_{n-2}\rangle^{q_1}) + [K]_r \end{aligned} \tag{15}$$

where

$$\begin{aligned}K = {}& \langle x_{n-1}y_0\rangle^{q_{n-1}} + \langle x_0 y_{n-1}\rangle^{q_0} + \frac{1}{2}E_{\text{main}}\\ & - \theta_{\text{index}} + \frac{1}{2}E_{\text{remain}}.\end{aligned} \tag{16}$$

In (15), the first term in the bracket is referred to as coarse-adjustment term and the second term $[K]_r$ is referred to as fine-adjustment term. The coarse adjustment term can be easily realized as simple circuit while the index is decided. On the other hand, the fine-adjustment term can be approached by the expected value in rounding operation after analyzing the statistics.

With the spirit of designing simple and realizable error-compensation circuit, we propose two types of binary thresholding for bias estimation. Both types of binary thresholding of θ_{index} are described as follows.

Type 1: See (17), at the bottom of the page.

Type 2: See (18), at the bottom of the page.

where $K_1, K_2, K_3,$ and K_4 are, respectively, the average of K for those satisfying $\theta_{\text{index}} = 0$, $\theta_{\text{index}} > 0$, $\theta_{\text{index}} < n$, and $\theta_{\text{index}} = n$.

Next, in order to achieve high accuracy compensation, an investigation on the choice of the generalized index θ_{index} is required. By exhaustive search, we can find some good generalized indices for small width n $(n \leq 12)$. For large width n, because of high computation load, we have to utilize statistic method to verify error-compensation equations performed by these better indices. It is noted that $\theta_{\text{J-K}}$ in Type 1 threshoding is a special index of the generalized index θ_{index} by choosing $q_0 = q_1 = \cdots = q_{n-1} = 0$.

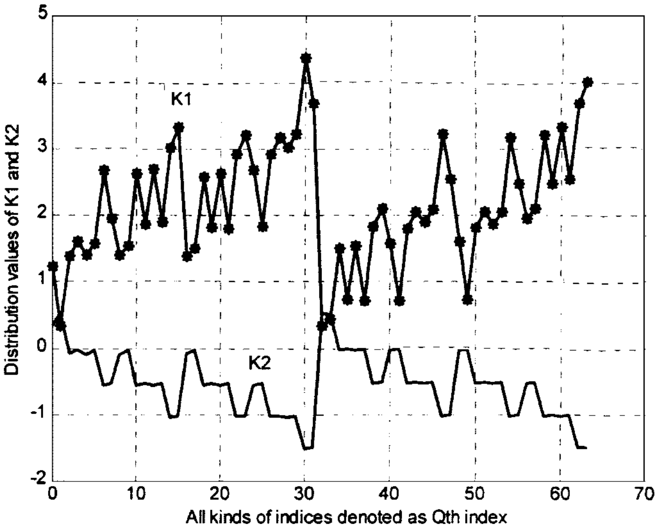

Fig. 3. Values of K_1 and K_2 versus different θ_{index} in Type 1 thresholding for $n = 6$.

For evaluating the resulting performance, given inputs X and Y, let ε, $\bar{\varepsilon}$, and υ be the absolute error between the standard multiplier and various truncated multiplier, the average error, and the variance of error, respectively. That is

$$\varepsilon \triangleq |P_{\text{Standard}} - P_{\text{Truncated}}| \tag{19}$$

$$\bar{\varepsilon} \triangleq E\{\varepsilon\} \tag{20}$$

$$\upsilon \triangleq E\{(\varepsilon - \bar{\varepsilon})^2\} \tag{21}$$

where P_{Standard} and $P_{\text{Truncated}}$ represent the output value for the standard multiplier and output value for various truncated multipliers, respectively, and $E\{\cdot\}$ is the expectation operator. Given $\theta_{\text{index}}(q_{n-1}, q_{n-2}, \ldots, q_0)$ in (13), in the following development, we call the index $\theta_{\text{index}}(q_{n-1}, q_{n-2}, \ldots, q_0)$ as the Qth index where

$$Q \triangleq q_{n-1} \times 2^{n-1} + q_{n-2} \times 2^{n-2} + \cdots + q_0 \times 2^0. \tag{22}$$

Note that Q has a range varying from 0 to $2^n - 1$; for example, $\theta_{\text{index}}(100\,001)$ denotes the 33th index for $n = 6$.

By full search simulation for $n = 6$, we obtain values of K_1 and K_2 as shown in Fig. 3 for all possible indices. In order to design a simply realizable error-compensation circuit, we choose the indices which satisfy $[K_1]_r \in \{0, 1\}$ and $[K_2]_r \in \{0, 1\}$ for the 6×6 multiplier. Interesting speaking, there exist three indices performing better error compensation than J-Ks' result in our full search experiment. However, these three indices (i.e., $\theta_{Q=1}$, $\theta_{Q=2^{n-1}}$, and $\theta_{Q=2^{n-1}+1}$) in Type 1 thresholding result in nonconstant K_1 and K_2 for different wordlength n. This phenomenon can be easily verified and explained by statistic techniques similar to what addressed in Section III. Since our goal is to find the fixed K_1 as well as K_2 independent of wordlength n and

$$\sigma_{\text{Type 1}} = \begin{cases} ((\langle x_{n-2}y_1\rangle^{q_{n-2}} + \langle x_{n-3}y_2\rangle^{q_{n-3}} + \cdots + \langle x_1 y_{n-2}\rangle^{q_1}) + [K_1]_r, & \text{if } \theta_{\text{index}} = 0\\ ((\langle x_{n-2}y_1\rangle^{q_{n-2}} + \langle x_{n-3}y_2\rangle^{q_{n-3}} + \cdots + \langle x_1 y_{n-2}\rangle^{q_1}) + [K_2]_r, & \text{if } \theta_{\text{index}} > 0\end{cases} \tag{17}$$

$$\sigma_{\text{Type 2}} = \begin{cases} \langle x_{n-2}y_1\rangle^{q_{n-2}} + \langle x_{n-3}y_2\rangle^{q_{n-3}} + \cdots + \langle x_1 y_{n-2}\rangle^{q_1} + [K_3]_r, & \text{if } \theta_{\text{index}} < n\\ \langle x_{n-2}y_1\rangle^{q_{n-2}} + \langle x_{n-3}y_2\rangle^{q_{n-3}} + \cdots + \langle x_1 y_{n-2}\rangle^{q_1} + [K_4]_r, & \text{if } \theta_{\text{index}} = n\end{cases} \tag{18}$$

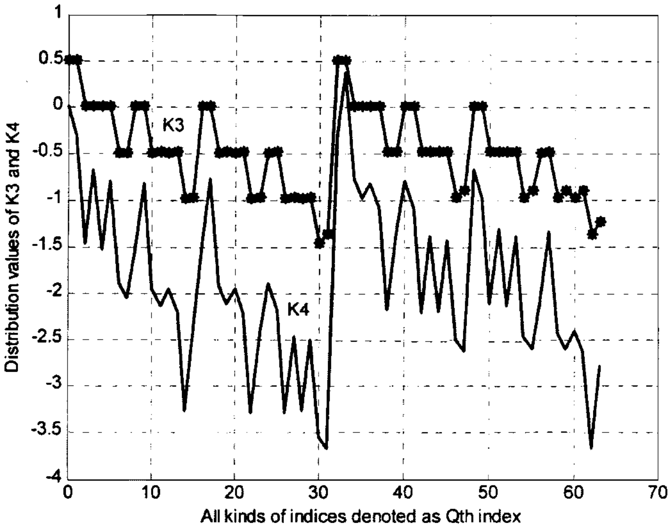

Fig. 4. Values of K_3 and K_4 versus different θ_{index} in Type 2 thresholding for $n = 6$.

lower-error in terms of average error and variance of error, we direct our attention to Type 2 thresholding.

Type 2 is another new proposed thresholding and, by exhaustive-search simulation, it is found that Type 2 structure is an excellent structure with feasible implementation and better performance. We obtain the values of K_3 and K_4, as shown in Fig. 4, for $n = 6$. To have a simple and feasible compensation circuit, it is found that the 33th index is one of the choices, where K_3 and K_4 are close to integers 1 and 0 as possible, respectively. Followed above procedure, we can simulate for the wordlength n from 4 to 12 by the full-search simulation in Type 2 thresholding. After possible simulation of different width n, we observe that the specific index $\theta_{Q=2^{n-1}+1}$ achieves better performance as described in Section IV in detail. Of course, the chosen index is of satisfying $[K_3]_r = 1$ and $[K_4]_r = 0$ for different width n. Hence, the simply realizable error-compensation structure with the lower truncation error for Type 2 thresholding is described as in (23), shown at the bottom of the page, where $\theta_{Q=2^{n-1}+1} = \overline{x_{n-1}y_0} + x_{n-2}y_1 + \cdots + x_1y_{n-2} + \overline{x_0y_{n-1}}$. Equation (23) has been completely simulated for $n \leq 12$ and can be mapped to a new structure. Thus, the proposed 8×8 lower-error fixed-width multiplier with the 129th index can be depicted in Fig. 5.

III. Fixed-Width Multiplier With Large Width

It is known that $(2^{n-1} + 1)$th index in Type 2 thresholding can be expressed as

$$\theta_{Q=2^{n-1}+1} = \overline{x_{n-1}y_0} + x_{n-2}y_1 + \cdots + x_1y_{n-2} + \overline{x_0y_{n-1}}. \tag{24}$$

By computer simulations, we find that this index $\theta_{Q=2^{n-1}+1}$ achieves better performance for small width n. It is difficult to simulate that the index is of the better performance for large width n since the exhaustive simulation takes significant computation time. In this section, we show that the index $\theta_{Q=2^{n-1}+1}$ is also suitable to being adopted to design the fixed-width multiplier for large width n; that is, we show that $[K_3]_r = 1$ and $[K_4]_r = 0$ for large width n. While analyzing fine-adjustment in (15), we encounter the problem in which $[K]_r$ depends on input signals of E_{main}, E_{remain}, $\theta_{Q=2^{n-1}+1}$ and other two terms. Herein, the probability of the input bits is assumed to be uniform distribution, so we approximate $(1/2)E_{\text{main}}$, $(1/2)E_{\text{remain}}$ and other terms using the analysis of output expected value of logic functions. Two cases can be taken into consideration: $\theta_{Q=2^{n-1}+1} < n$ and $\theta_{Q=2^{n-1}+1} = n$.

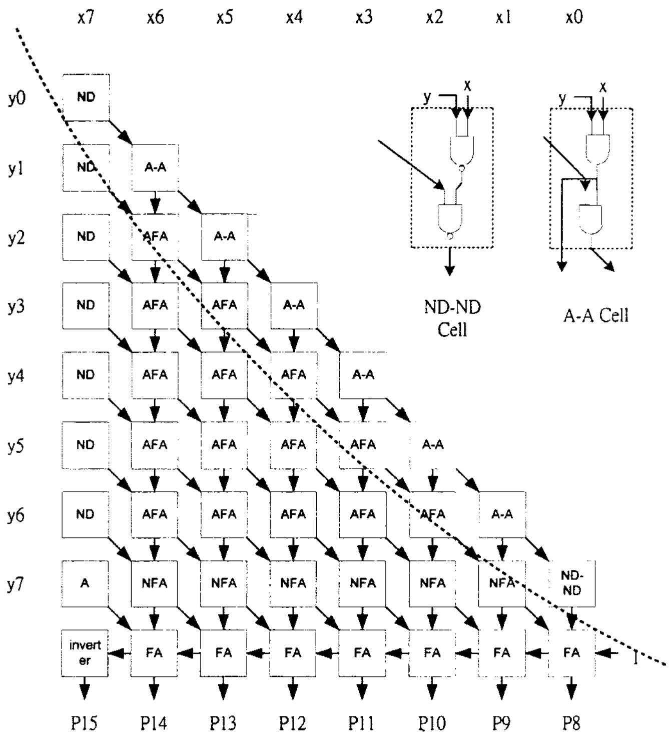

Fig. 5. Proposed lower-error fixed-width 8×8 multiplier.

Case 1: $\theta_{Q=2^{n-1}+1} < n$ It follows from (10) that

$$E\left\{\frac{1}{2}E_{\text{main}}\right\} = \frac{1}{2} \times \left(\frac{3}{4} + \frac{3}{4} + \frac{1}{4} \times (n-2)\right) = \frac{n}{8} + \frac{1}{2}. \tag{25}$$

Note that $E\{x_iy_j\} = 1/4$ and $E\{\overline{x_iy_j}\} = 3/4$, since the probability of input bits is assumed to be uniform distribution. Similarly, we can obtain (26) from (11) as

$$\begin{aligned} E&\left\{\frac{1}{2}E_{\text{remain}}\right\} \\ &= \frac{1}{2^2} \times \frac{1}{4} \times (n-1) + \frac{1}{2^3} \times \frac{1}{4} \cdot (n-2) + \cdots + \frac{1}{2^n} \times \frac{1}{4} \times 1 \\ &= \frac{1}{4}\left(\frac{1}{2^2} \times (n-1) + \frac{1}{2^3} \times (n-2) + \cdots + \frac{1}{2^n} + 1\right) \\ &\cong \frac{n}{8} - \frac{1}{4}, \quad \text{if } n \geq 4. \end{aligned} \tag{26}$$

$$\sigma_{\text{Type 2},Q=2^{n-1}+1} = \begin{cases} x_{n-2}y_1 + x_{n-3}y_2 + \cdots + x_1y_{n-2} + 1, & \text{if } \theta_{Q=2^{n-1}+1} < n \\ x_{n-2}y_1 + x_{n-3}y_2 + \cdots + x_1y_{n-2}, & \text{if } \theta_{Q=2^{n-1}+1} = n \end{cases} \tag{23}$$

Apparently, while using $\theta_{Q=2^{n-1}+1}$, we find that

$$\theta_{Q=2^{n-1}+1} = \overline{x_{n-1}y_0} + \overline{x_0y_{n-1}} + \sum_{\substack{(i+j)=n-1\\ i\neq n-1, j\neq n-1}} x_iy_j = E_{\text{main}} \tag{27}$$

where $q_{n-1} = q_0 = 1$ and $q_{n-2} = q_{n-3} = \cdots = q_1 = 0$. By applying (25)–(27), it follows from (16) that the expected value K_3 is given by

$$\begin{aligned}[K_3]_r &= [E\{K\}]_r \\ &= \left[E\left\{\overline{x_{n-1}y_0} + \overline{x_0y_{n-1}} - \frac{1}{2}E_{\text{main}} + \frac{1}{2}E_{\text{remain}}\right\}\right]_r \\ &= \left[\frac{3}{4} + \frac{3}{4} - \frac{n}{8} - \frac{1}{2} + \frac{n}{8} - \frac{1}{4}\right]_r = 1.\end{aligned} \tag{28}$$

Consequently, we obtain following equation:

$$\sigma_{\text{Type 2},Q=2^{n-1}+1} = (x_{n-2}y_1 + \cdots + x_1y_{n-2}) + 1, \quad \text{if } \theta_{Q=2^{n-1}+1} < n \tag{29}$$

Case 2: $\theta_{Q=2^{n-1}+1} = n$ This case $\theta_{Q=2^{n-1}+1} = n$ is met only when $\overline{x_0y_{n-1}} = \overline{x_{n-1}y_0} = 1$ and $x_1y_{n-2} = x_2y_{n-3} = \cdots = x_{n-2}y_1 = 1$. That is, Case 2 is a conditional probability case and, thus, we deduce (30) and (31) as follows:

$$E\left\{\frac{1}{2}E_{\text{main}}\right\} = \frac{1}{2} \times 1 \times n = \frac{1}{2}n \tag{30}$$

$$\begin{aligned}E\left\{\frac{1}{2}E_{\text{remain}}\right\} &= \frac{1}{2^2}\left(\frac{1}{3} \times 1 \times 2 + 1 \times (n-3)\right) \\ &\quad + \frac{1}{2^3}\left(\frac{1}{3} \times 1 \times 2 + 1 \times (n-4)\right) \\ &\quad + \cdots + \frac{1}{2^{n-2}}\left(\frac{1}{3} \times 1 \times 2 + 1 \times 1\right) \\ &\quad + \frac{1}{2^{n-1}}\left(\frac{1}{3} \times 1 \times 2\right) \\ &\quad + \frac{1}{2^n}\left(\frac{1}{9} \times 1 \times 1\right) \\ &\cong \frac{1}{2}n - \frac{5}{3}, \quad \text{if } n \geq 4.\end{aligned} \tag{31}$$

In view of (30), (31), and (16), we have

$$\begin{aligned}[K_4]_r &= [E\{K\}]_r \\ &= \left[E\left\{\overline{x_{n-1}y_0} + \overline{x_0y_{n-1}} - \frac{1}{2}E_{\text{main}} + \frac{1}{2}E_{\text{remain}}\right\}\right]_r = 0.\end{aligned} \tag{32}$$

Thus, $\sigma_{\text{Type 2},Q=2^{n-1}+1}$ for $\theta_{Q=2^{n-1}+1} = n$ can be written as

$$\sigma_{\text{Type 2},Q=2^{n-1}+1} = x_{n-2}y_1 + x_{n-3}y_2 + \cdots + x_1y_{n-2}, \quad \text{if } \theta_{Q=2^{n-1}+1} = n \tag{33}$$

By combining (29) and (33), we get the conclusioin in (34), shown at the bottom of the page. Equation (34) shows that Type 2 thresholding with the chosen index $\theta_{Q=2^{n-1}+1}$ is suitable to implementing the fixed-width multiplier with large width n.

TABLE I
COMPARISON RESULTS OF MAXIMUM ERROR $\varepsilon_{\max}$

Multiplier	n=4	n=6	n=8	n=10	n=12
K-G-As' Structure	33	193	1281	6145	32769
J-Ks' Structure	21	107	515	2403	10979
Proposed Structure	17	89	441	2105	9785

TABLE II
COMPARISON RESULTS OF AVERAGE ERROR $\bar{\varepsilon}$

Multiplier	n=4	n=6	n=8	n=10	n=12
K-G-As' Structure	6.96	41.01	188.29	906.40	3842.06
J-Ks' Structure	7.20	37.27	170.46	736.62	3065.25
Proposed Structure	5.17	24.07	105.96	456.14	1907.36

TABLE III
COMPARISON RESULTS OF VARIANCE OF ERRORS υ

Multiplier	n=4	n=6	n=8	n=10	n=12
K-G-As' Structure	39.80	788.45	22959.01	416043	9204493
J-Ks' Structure	28.24	537.70	10158.54	190805	3417020
Proposed Structure	17.63	320.65	6031.32	112079	1973508

TABLE IV
COMPARISON RESULTS OF AREA RATIO R

Multiplier	n=4	n=6	n=8	n=10	n=12
K-G-As' Structure	0.555	0.536	0.527	0.522	0.518
J-Ks' Structure	0.608	0.569	0.550	0.540	0.533
Proposed Structure	0.608	0.569	0.550	0.540	0.533

IV. PERFORMANCE COMPARISONS AND AREA COMPARISON

In this section, it has been shown that the proposed fixed-width multiplier achieves better performance than the other fixed-width multipliers by computer simulations. The performance is evaluated in terms of average error $\bar{\varepsilon}$ and the variance of errors υ defined in (20) and (21), respectively, and the maximum error defined by

$$\varepsilon_{\max} \triangleq \max(|P_{\text{Standard}} - P_{\text{Truncated}}|). \tag{35}$$

It is obvious that a fixed-width multiplier is more accurate if $\bar{\varepsilon}$, υ and $\varepsilon_{\max}$ are smaller. Tables I–III show the simulated results for the various fixed-width multipliers of different width n. The K-G-As' structure [6] is the truncated multiplier with constant compensation bias only depending on the width of the multiplier, the J-Ks' structure is the fixed-width multiplier devised by Jou *et al.* [7], and the proposed structure is our fixed-width multiplier of Type 2 thresholding with the index $\theta_{Q=2^{n-1}+1}$. The comparison results show that our proposed fixed-width multiplier is more accurate than the others. The excellent performance is achieved due to the fact that we derive a better error-compensation bias to reduce the effect of truncation error.

Let A_{AND}, A_{NAND}, A_{OR}, A_{NOR}, A_{HA}, and A_{FA} be the areas of an AND gate, NAND gate, OR gate, NOR gate, a half adder, and a full adder, respectively. Furthermore, we adopt the same notation as in [6] such that let $A_{\text{AND}} = \phi_1 A_{\text{FA}}$, $A_{\text{AND}} = \phi_2 A_{\text{FA}}$, $A_{\text{OR}} = \phi_3 A_{\text{OR}}$, $A_{\text{NOR}} = \phi_4 A_{\text{FA}}$, and $A_{\text{HA}} = \xi A_{\text{FA}}$, where $0 < \phi_1, \phi_2, \phi_3, \phi_4 < 0.1$ and $0 < \xi < 0.5$. For convenience of comparison, we reasonably

$$\sigma_{\text{Type 2},Q=2^{n-1}+1} = \begin{cases} x_{n-2}y_1 + x_{n-3}y_2 + \cdots + x_1y_{n-2} + 1, & \text{if } \theta_{Q=2^{n-1}+1} < n \\ x_{n-2}y_1 + x_{n-3}y_2 + \cdots + x_1y_{n-2}, & \text{if } \theta_{Q=2^{n-1}+1} = n \end{cases} \tag{34}$$

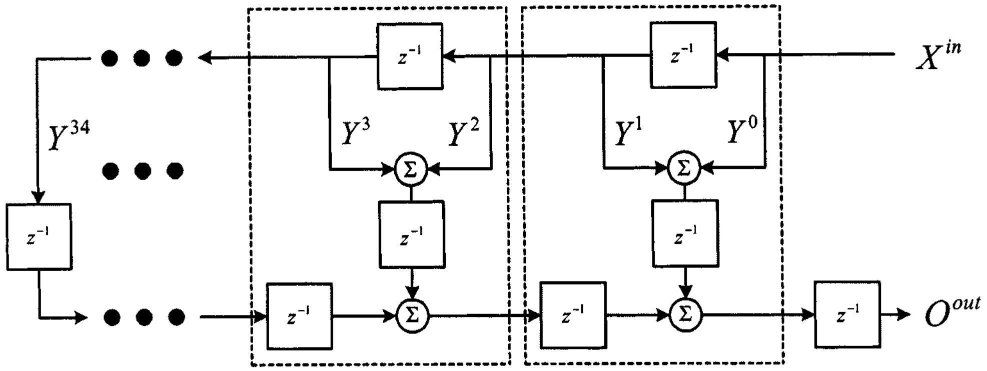

Fig. 6. Block diagram of a 35-tap FIR digital filter.

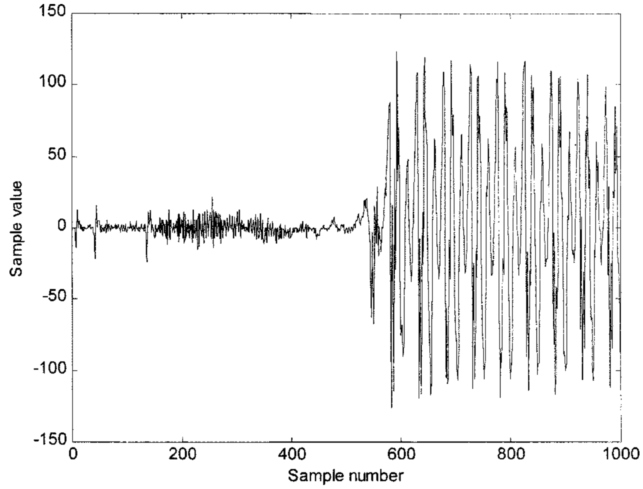

Fig. 7. Original input voice signal with 1000 samples.

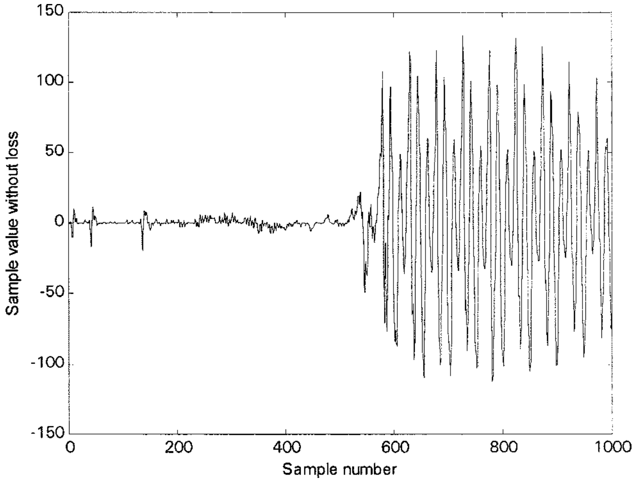

Fig. 8. Standard voice output signal without loss.

assume $\phi = \phi_1 = \phi_2 = \phi_3 = \phi_4$. The areas of the standard $n \times n$ multiplier is given as

$$A_{\text{Standard}} = n^2\phi A_{\text{FA}} + (n-1)\xi A_{\text{FA}} + (n-1)^2 A_{\text{FA}}. \tag{36}$$

On the other hand, the maximum area of $n \times n$, K-G-As' multiplier, J-Ks' multiplier, and the proposed multiplier are briefly formulated as

$$A_{\text{K-G-A}} = \frac{1}{2}n(n-1)(\phi+1)A_{\text{FA}} \tag{37}$$

$$A_{\text{J-K}} = \frac{1}{2}n(n-1)(\phi+1)A_{\text{FA}} + (2n-1)\phi A_{\text{FA}} \tag{38}$$

$$A_{Q=2^{n-1}+1} = \frac{1}{2}n(n-1)(\phi+1)A_{\text{FA}} + (2n-1)\phi A_{\text{FA}} \tag{39}$$

where subscripts denote the corresponding fixed-width multipliers. The area ratio is defined as follows:

$$R \triangleq \frac{A_{\text{Truncated}}}{A_{\text{Standard}}}. \tag{40}$$

Substituting (36)–(39) into (40) to evaluate area-ratio with $\phi = 0.09$ and $\xi = 0.45$, we tabulated as Table IV. The area ratio in Table IV shows that our proposed multiplier is area efficient since closely to half the area of the standard multiplier.

V. DSP Application of Fixed-Width Multipliers

In this section, we apply the proposed fixed-width multiplier to the 35-tap FIR filter as shown in Fig. 6 for speech processing. The behavior of a digital FIR filter can be represented as follows:

$$O^m = \sum_{i=0}^{L-1} X^i Y^{(m-i)} \tag{41}$$

where

X^i	input sequence;
Y^i	filter coefficient;
O^i	output sequence at ith discrete time.

The superscript i, is the time index. First, for practical consideration [8], the maximum input voice data and filter coefficient in two's complement are normalized to the same value 127 with 8-bit quantization. In the experimental simulation, the temporary output is an accumulated value using 32 bits. Finally, the outputs, O^i, are then obtained by scaling the accumulated values. 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. 7. 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. 8 shows the standard filtering output signals and Figs. 9–11 show the filtering output signals processed by the 35-tap low-pass FIR filter applying a variety

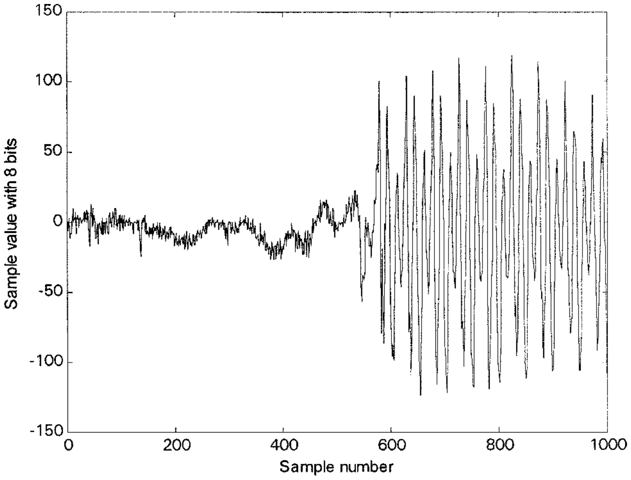

Fig. 9. Output signals using K-G-As' structure [6].

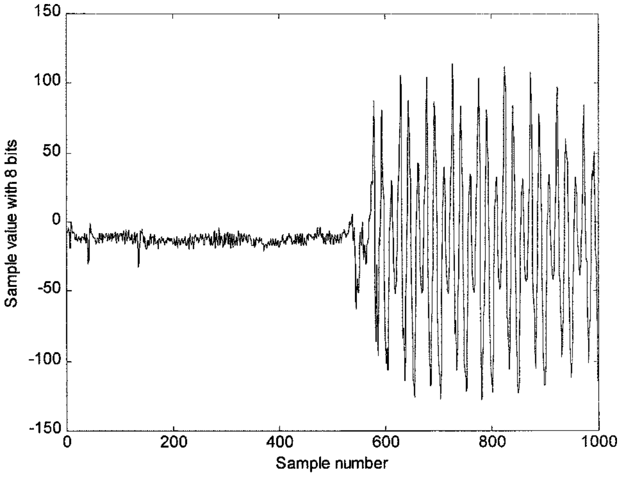

Fig. 10. Output signals using J-Ks' structure [7].

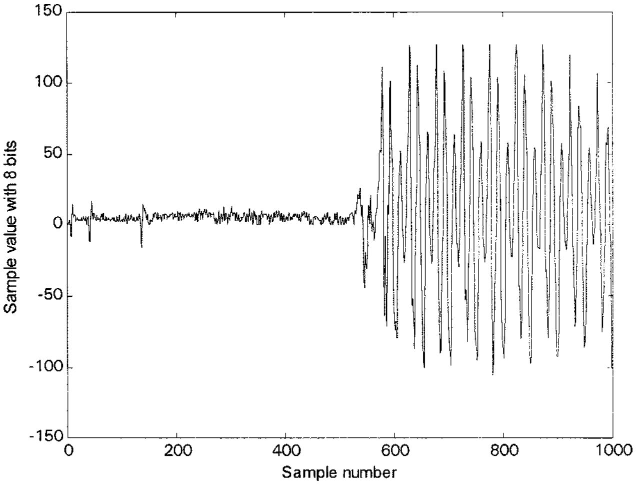

Fig. 11. Output signals using the proposed structure.

of fixed-width multipliers. Using constant bias K-G-As' multiplier, it is seen from Fig. 9 that there are larger average error and variance of errors in consonant part. Fig. 10 is obtained by applying J-Ks' multiplier and it shows better performance than that of Fig. 9. However, compared to standard output, we find that output signals in Fig. 10 still have large average error as well as variance of the errors. The smaller average error and variance of the errors especially for consonant part is obtained by using our proposed fixed-width multiplier as shown in Fig. 11.

VI. Conclusion

This brief develops the general methodology for designing a lower-error two's-complement fixed-width multiplier. By properly choosing the generalized index, we derive a better error-compensation bias to reduce the truncation error and then construct a lower error fixed-width multiplier, which is area-efficient for VLSI realization. Finally, we successfully apply the proposed fixed-width multiplier to a digital FIR filter for speech processing application. It has shown that the performance for consonant part is better than that using other fixed-width multipliers. On the other hand, interested readers can study other binary thresholding with generalized indices and use different operators, such as ceiling [9] or flooring operators, to devise another useful and realizable fixed-width multiplier.

References

[1] S. Y. Kung, *VLSI Array Processors*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[2] S. L. Freeny, "Special-purpose hardware for digital filtering," *Proc. IEEE*, vol. 63, pp. 633–647, Apr. 1975.

[3] L. D. Van, C. C. Tang, S. Tengchen, and W. S. Feng, "A new VLSI architecture without global broadcast for 2-D digital filters," in *Proc. IEEE Int. Symp. Circuits Systems*, Geneva, Switzerland, May 2000, pp. 547–550.

[4] K. Hwang, *Computer Arithmetic: Principles, Architecture, and Design*. New York: Wiley, 1979.

[5] F. Cavanagh, *Digital Computer Arithmetic: Design and Implementation*. New York: McGraw-Hill, 1984.

[6] S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-efficient multipliers for digital signal processing applications," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 90–94, Feb. 1996.

[7] J. M. Jou and S. R. Kuang, "Design of a low-error fixed-width multiplier for DSP applications," *Electron. Lett.*, vol. 33, no. 19, pp. 1597–1598, 1997.

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

[9] L. D. Van, S. S. Wang, S. Tengchen, W. S. Feng, and B. S. Jeng, "Design of a lower error fixed-width multiplier for speech processing application," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, pp. 130–133, May 1999.