PAPER

Fixed-Width Group CSD Multiplier Design

Yong-Eun KIM[†], **Kyung-Ju CHO**[††a)], ***Nonmembers***, **Jin-Gyun CHUNG**[†††], ***Member***, ***and*** **Xinming HUANG**[††††], ***Nonmember***

SUMMARY This paper presents an error compensation method for fixed-width group canonic signed digit (GCSD) multipliers that receive a W-bit input and generate a W-bit product. To efficiently compensate for the truncation error, the encoded signals from the GCSD multiplier are used for the generation of the error compensation bias. By Synopsys simulations, it is shown that the proposed method leads to up to 84% reduction in power consumption and up to 78% reduction in area compared with the fixed-width modified Booth multipliers.

key words: *fixed-width, GCSD multiplier, quantization error, digital arithmetic*

1. Introduction

In some DSP applications such as FFT and pulse-shaping filters, multiplications are performed only with a few predetermined coefficients which are time-varying in periodical order. In these applications, multipliers should have programmability. When a few coefficients share a multiplier, modified Booth encoding, which halves the number of partial products, is generally used. To further reduce the number of partial products, the group canonic signed digit (GCSD) multiplier in [1] was recently proposed based on the variation of canonic signed digit (CSD) encoding [2] and partial product sharing algorithm. This multiplier provides an efficient design when the multiplications are performed only with a few predetermined coefficients (e.g., FFT).

In many multimedia and digital signal processing (DSP) applications, it is desirable to maintain fixed-with property through multiplication operations to avoid quick growth in word size. For example, the $(2W-1)$-bit product obtained from two W-bit operands is quantized to W-bits by eliminating the $(W-1)$-least-significant bits (LSBs). In practice, fixed-width multipliers can be designed based on Baugh-Wooley, modified Booth and CSD algorithms. In typical fixed-width multipliers, the adder cells required for the computation of the $(W-1)$-LSBs are eliminated and error compensation biases are introduced to the retained adder cells.

In order to reduce the truncation error, various error compensation methods for fixed-width multipliers have been proposed [3]–[9]. Error compensation biases of these methods can be classified into constant biases [3] and adaptive biases [4]–[9]. While a constant bias is generated independent of the truncated partial product bits and is fixed for a given input word size, an adaptive bias is determined depending on each input. Thus, adaptive bias methods achieve better computation accuracy than the constant bias method.

In this paper, we propose an error compensation method for low-error fixed-width GCSD multiplier. To efficiently compensate for the truncation error with reduced hardware complexity, the encoded signals from the GCSD multiplier are used for the generation of error compensation bias.

This paper is organized as follows. After a brief review of the GCSD multiplier in Sect. 2, we propose an error compensation bias design method for GCSD multipliers in Sect 3. In Sect. 4, two application examples of the proposed fixed-width multiplier design method are presented. Finally, short statements conclude this paper.

2. GCSD Multiplier

Figure 1 shows the N-point radix-2^4 single-path delay feedback (SDF) FFT architecture [10]. In the first and the third multiplication blocks, three coefficients $\{\cos(\pi/8),\ \cos(\pi/4),\ \text{and}\ \sin(\pi/8)\}$ are multiplied by an complex input signal in periodical order as

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ \cos(\pi/8) - j\sin(\pi/8) \\ \cos(\pi/4) - j\sin(\pi/4) \\ \sin(\pi/8) - j\cos(\pi/8) \end{bmatrix} (x_r + jx_i). \quad (1)$$

In general, the multiplications in (1) can be implemented using a programmable multiplier such as the modified Booth multiplier. If the coefficient word-length is W, the number of the partial products obtained by the modified Booth algorithm is $W/2$.

To further reduce the number of partial products, the following coefficient grouping algorithm can be used [1]:

Manuscript received June 2, 2009.
Manuscript revised December 2, 2009.

[†]The author is with Vehicle-IT Fusion Research Center, Korea Automotive Technology Institute, 74 Yongjeong-Ri, Pungse-Myun, Dongnum-Gu, Chonan-si, 330–912, Korea.
[††]Corresponding author. The author is with Korea Association of Aids to Navigation (KAAN), IT Castle2 12F, Kasan Geumcheon-ku, Seoul, 153–768, Korea.
[†††]The author is with Division of Electronic Eng., and Information and Communication Research Center, Chonbuk National University, Deokjindong 1-ga Deokjin-gu, Jeonju, 561–756, Korea.
[††††]The author is with the Dept. of Electrical & Computer Engr., Worcester Polytechnic Institute, Worcester, MA, USA.
a) E-mail: kjcho@chonbuk.ac.kr
DOI: 10.1587/transinf.E93.D.1497

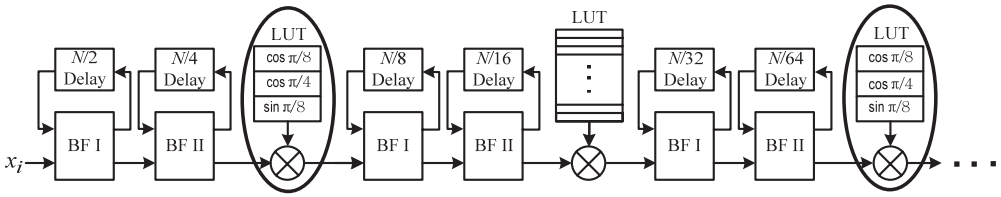

Fig. 1 Radix-2^4 SDF FFT architecture.

Table 1 CSD representation and grouping of coefficients.

Column	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cos(π/8)	1	0	0	0	-1	0	-1	0	0	1	0	0	0	0
cos(π/4)	1	0	-1	0	-1	0	1	0	1	0	0	0	0	0
sin(π/8)	0	1	0	-1	0	0	0	1	0	0	0	0	0	-1
Group	G_4		G_3		G_2		G_1		G_0					

Table 2 New representation of CSD coefficients using control signals.

Group	G_4			G_3			G_2			G_1			G_0		
Control signals	S_4	N_4	Z_4	S_3	N_3	Z_3	S_2	N_2	Z_2	S_1	N_1	Z_1	S_0	N_0	Z_0
cos(π/8)	1	0	1	1	1	0	1	1	1	1	1	1	01	0	1
cos(π/4)	1	0	1	1	1	1	1	1	1	1	0	1	10	0	1
sin(π/8)	0	0	1	0	1	1	0	1	0	0	0	1	00	1	1

1. If the number of given 2's complement coefficients is N_c with the word length of N_w bits, the coefficients are arranged as an $N_c \times N_w$ table.
2. The coefficients in the table are converted to CSD encoded representations.
3. Starting from the first column, a group is defined such that each row in a group contains at most one nonzero digit. A group should contain as many columns as possible so that the number of groups is minimized.

By applying the grouping algorithm to the three coefficients in (1) with $N_w = 14$, the CSD coefficient table with 5 groups is obtained as shown in Table 1. Each group G_i generates a corresponding partial product P_i. Thus, the multiplication result (Y) is obtained as

$$Y = P_4 + 2^{-2}P_3 + 2^{-4}P_2 + 2^{-6}P_1 + 2^{-8}P_0. \quad (2)$$

In Table 1, the number of partial products required by the grouping algorithm is only 5, while the modified Booth encoding requires 7 ($= N_w/2$) partial products. Thus the grouping algorithm reduces the number of partial products by 2, which can lead to lower power consumption and higher speed. Each group includes at least two columns by the grouping algorithm since CSD coding does not allow any consecutive nonzero digits. Thus, the number of partial products generated by the grouping algorithm is always less than or equal to that of the modified Booth encoding.

If the nonzero digit locations of two groups are the same as in G_4 and G_1 in Table 1, the two groups can share PP generation circuits. The sign difference in the first rows of G_4 and G_1 can be taken care of later by additional complementing circuits. For any row in a group that contains only 0's, the corresponding PP is 0. In this case, the zero digits in the row can be changed to nonzero digits to share the partial product generation circuits, since the output value can be easily changed back to 0 using a control signal. By the partial product sharing algorithm in [1], a new representation of each group in Table 1 can be obtained using control signals as shown in Table 2, where S_i, N_i and Z_i are shift, negation, and zero control signals, respectively.

In conventional approach, the coefficient look-up table (LUT) has 14 columns if the coefficient word-length is 14. In GCSD multipliers, encoded values are stored instead of binary coefficients. In [11], a look-up table (LUT) reduction method for modified Booth encoded coefficients was proposed. By similar approach, the number of columns of Table 2 can be reduced to 7. Thus, in this case, LUT size is reduced by 50% compared with conventional approaches. By Synopsys simulations, it is shown that the GCSD method reduces the area, power consumption and propagation de-

lay by 41%, 45% and 12%, respectively, compared with the conventional modified Booth multiplier [1].

3. Fixed-Width GCSD Multiplier

For the coefficient Table 1 with $N_w = 14$, the corresponding partial product array for the GCSD multiplier can be obtained as shown in Fig. 2. The partial product array can be divided into MP and LP as shown in Fig. 2, where MP and LP mean more significant and less significant parts, respectively. Then, if S_MP and S_LP represent the sums of the elements in MP and LP, respectively, we can express $(2W-1)$-bit ideal product P_I as

$$P_I = S\_MP + S\_LP. \tag{3}$$

In typical fixed-width multipliers, the adder cells required for S_LP are omitted and appropriate biases are introduced to the retained adder cells based on a probabilistic estimation. Thus, the W-bit quantized product P_Q can be expressed as

$$P_Q = S\_MP + \sigma \times 2^{-(W-1)}, \tag{4}$$

where σ means the error compensation bias. Note that σ approximates the carry signals propagated from LP to MP.

To generate error compensation bias more efficiently, the truncated bits in LP can be further divided into LP_{major} and LP_{minor} depending on their effects on the truncation error. Then, S_LP can be expressed as

$$S\_LP = S\_LP_{major} + S\_LP_{minor}. \tag{5}$$

As an example, in Fig. 2, S_LP_{major} and S_LP_{minor} can be expressed as

$$\begin{aligned} S\_LP_{major} &= p_{0,12} + p_{1,6} + p_{2,4} + p_{3,2} + p_{4,0}, \\ S\_LP_{minor} &= 2^{-1}(p_{0,11} + p_{1,5} + p_{2,3} + p_{3,1}) \\ &\quad + 2^{-2}(p_{0,10} + p_{1,4} + p_{2,2} + p_{3,0} + N_3 Z_3) + \cdots \\ &\quad + 2^{-12}(p_{0,0} + N_0), \end{aligned} \tag{6}$$

where $p_{i,j}$ means the jth partial product bit of partial product P_i.

Theoretically, the most accurate error compensation bias can be obtained by true rounding as

$$\sigma_{true} = \left[\frac{S\_LP}{2}\right]_r, \tag{7}$$

where $[t]_r$ denotes the rounding operation of t.

Let 2^{-L_i} be the weight of the LSB of the partial product P_i. Also, let M_i be the required number of shift-left bit positions for the partial product P_i. As an example, for G_0 in Table 1, coefficient $\sin(\pi/8)$ has a nonzero digit at the last column of G_0. Thus M_0 is defined to be 0 for $\sin(\pi/8)$ since no shift operation is required in this case. On the other hand, $\cos(\pi/8)$ has a nonzero digit at the second column of G_0. Thus for $\cos(\pi/8)$, M_0 is defined to be 4 since shift-left by 4 bit positions are required in this case. Using Table 1, Table 2 and Fig. 2, the possible values of $LP_{minor}(P_i)$ can be obtained depending on the control signals as shown in Fig. 3. If $N_i = 1$, the input signals are negated as can be seen in Fig. 3. When $Z_i = 0$, partial product bits are changed to 0.

Assume that each bit x_i of input X has the uniform probability distribution. Then, the expected value of x_i is

$$E[x_i] = 1/2. \tag{8}$$

Then, it can be shown that the expected value of $S\_LP_{minor}(P_i)$ can be computed as

$$\begin{aligned} &E[S\_LP_{minor}(P_i)] \\ &= \begin{cases} 0, & Z_i = 0 \\ 2^{-1}(1 - 2^{-f(L_i - M_i)}), & Z_i \neq 0,\ N_i = 0 \\ 2^{-1}(1 + 2^{-f(L_i - M_i)}), & Z_i \neq 0,\ N_i \neq 0, \end{cases} \end{aligned} \tag{9}$$

where

$$f(L_i - M_i) = \begin{cases} L_i - M_i, & \text{for } L_i > M_i \\ 0, & \text{for } L_i \leq M_i. \end{cases} \tag{10}$$

The error compensation bias of fixed-width modified Booth multiplier in [8] is defined as follows:

$$\sigma_{[8]} = C_E[S\_LP_{major} + C_A\{S\_LP_{minor}\}], \tag{11}$$

where $C_E[t]$ and $C_A[t]$ represent the exact carry value and the approximate carry value of t, respectively. In (11), $C_A\{S\_LP_{minor}\}$ is the approximate carry value (a_carry) propagated from LP_{minor} to LP_{major}. In [8], the expect value of $S\_LP_{minor}(P_i)$ is identified as

$$E[S\_LP_{minor}(P_i)]_{[8]} = \begin{cases} 0, & y_i'' = 0, \\ 2^{-1}, & y_i'' = 1. \end{cases} \tag{12}$$

where y_i'' corresponds to Z_i signal in this paper. The approximate carry signal in [8] is defined as the rounded value of

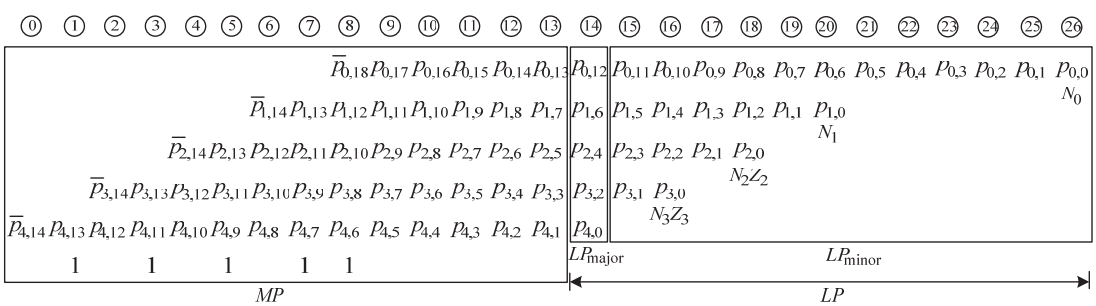

Fig. 2 MP and LP in a partial product array of GCSD multiplier.

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	
$P_{0,11}$	$P_{0,10}$	$P_{0,9}$	$P_{0,8}$	$P_{0,7}$	$P_{0,6}$	$P_{0,5}$	$P_{0,4}$	$P_{0,3}$	$P_{0,2}$	$P_{0,1}$	$P_{0,0}$ N_0	
x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	0	0	0	0 0	$\leftarrow M_0=4\ N_0=0\ Z_0=1$
x_6	x_5	x_4	x_3	x_2	x_1	x_0	0	0	0	0	0 0	$\leftarrow M_0=5\ N_0=0\ Z_0=1$
$\overline{x}_{11}$	$\overline{x}_{10}$	$\overline{x}_9$	$\overline{x}_8$	$\overline{x}_7$	$\overline{x}_6$	$\overline{x}_5$	$\overline{x}_4$	$\overline{x}_3$	$\overline{x}_2$	$\overline{x}_1$	$\overline{x}_0$ 1	$\leftarrow M_0=0\ N_0=1\ Z_0=1$

(a)

$p_{1,5}$	$p_{1,4}$	$p_{1,3}$	$p_{1,2}$	$p_{1,1}$	$p_{1,0}$ N_1	
$\overline{x}_4$	$\overline{x}_3$	$\overline{x}_2$	$\overline{x}_1$	$\overline{x}_0$	$\overline{0}$ 1	$\leftarrow M_1=1\ N_1=1\ Z_1=1$
x_4	x_3	x_2	x_1	x_0	0 0	$\leftarrow M_1=1\ N_1=0\ Z_1=1$
x_5	x_4	x_3	x_2	x_1	x_0 0	$\leftarrow M_1=0\ N_1=0\ Z_1=1$

(b)

$p_{2,3}$	$p_{2,2}$	$p_{2,1}$	$p_{2,0}$ N_2Z_2	
$\overline{x}_2$	$\overline{x}_1$	$\overline{x}_0$	$\overline{0}$ 1	$\leftarrow M_2=1\ N_2=1\ Z_2=1$
0	0	0	0 0	$\leftarrow M_2=0\ N_2=1\ Z_2=0$

(c)

$p_{3,1}$	$p_{3,0}$ N_3Z_3	
0	0 0	$\leftarrow M_3=1\ N_3=1\ Z_3=0$
$\overline{x}_0$	$\overline{0}$ 1	$\leftarrow M_3=1\ N_3=1\ Z_3=1$
$\overline{x}_1$	$\overline{x}_0$ 1	$\leftarrow M_3=0\ N_3=1\ Z_3=1$

(d)

Fig. 3 Possible values of (a) $LP_{minor}(P_0)$, (b) $LP_{minor}(P_1)$, (c) $LP_{minor}(P_2)$, and (d) $LP_{minor}(P_3)$ depending on the control signals.

$E[S\_LP_{minor}]$. For given N_w, the approximate carry value of the fixed-width modified Booth multiplier is computed as

$$a\_carry_{[8]} = \left\lfloor \sum_{i=0}^{N_w/2-2} \frac{y_i''}{2} \right\rfloor. \tag{13}$$

Thus, (11) can be rewritten as

$$\sigma_{[8]} = \left\lfloor \frac{S\_LP_{major} + a\_carry}{2} \right\rfloor, \tag{14}$$

where $\lfloor t \rfloor$ defines the largest integer less than or equal to t.

Note that in [8], only control signal Z_i (or, y_i'') is used for the calculation of $E[S\_LP_{minor}]$. However, in this paper, we can compute more accurately the expected values of the truncated bits in LP_{minor} of the GCSD multiplier since L_i, N_i, M_i and Z_i signals are used for the computation of $E[S\_LP_{minor}]$. As an example, for P_0 in Fig. 3, the control signals $\{L_0, M_0, N_0, Z_0\}$ are varied as $\{12, 4, 0, 1\}$, $\{12, 5, 0, 1\}$ and $\{12, 0, 1, 1\}$, depending on the selected coefficient from $\{\cos(\pi/8), \cos(\pi/4), \sin(\pi/8)\}$. By (9), $E[S\_LP_{minor}(P_0)]$ can be computed as $2^{-1}(1-2^{-8})$, $2^{-1}(1-2^{-7})$, and $2^{-1}(1+2^{-12})$, respectively.

Let the partial products included in LP_{minor} be P_K, P_{K-1}, …, P_0. Since the LSB of P_i always has a smaller weight than that of P_{i-1} under any coefficient conditions, the following relation holds:

$$0 \le f(L_K - M_K) < f(L_{K-1} - M_{K-1}) < \cdots < f(L_0 - M_0). \tag{15}$$

Thus, using (15) and CSD property, the maximum value of $E[S\_LP_{minor}]$ can be obtained as

$$\begin{aligned} &\max\{E[S\_LP_{minor}]\} \\ &\quad = 2^{-1} \times \max\left\{\left(1 + 2^{-f(L_K - M_K)}\right) + \left(1 + 2^{-f(L_{K-1} - M_{K-1})}\right) \right. \\ &\qquad \left. + \cdots + \left(1 + 2^{-f(L_0 - M_0)}\right)\right\} \\ &\quad = 2^{-1} \times (K + 1 + 2^0 + 2^{-2} + 2^{-4} + \cdots + 2^{-2K}) \\ &\quad = 2^{-1} \times (K + 2 + 2^{-2} + \cdots + 2^{-2K}). \end{aligned} \tag{16}$$

The carry signals generated from LP_{minor} are determined by the integer part of the terms inside the parenthesis in (16). Thus, $2^{-f(L_K - M_K)}$ can have an effect on the carry signals generated from LP_{minor} to LP_{major} but $\sum_{i=0}^{K-1} 2^{-f(L_i - M_i)}$ has no effect on the carry signals.

In addition, for the partial product P_K, when M_K is larger than or equal to L_K, the partial product bits inside the LP_{minor} are filled with 0's. Thus, when the partial product P_K is negated (i.e., $N_K = 1$), a carry signal is propagated to LP_{major}.

Based on these observations, $E[S\_LP_{minor}]$ can be easily computed as

$$E[S\_LP_{minor}] = 2^{-1} \times \{g(L_K - M_K) \cdot 2N_K + \overline{g(L_K - M_K)} \cdot Z_K \cdot N_K + Z_{K-1} + \cdots + Z_0\}, \quad (17)$$

where

$$g(L_K - M_K) = \begin{cases} 1, & \text{for } L_K \le M_K \\ 0, & \text{for } L_K > M_K, \end{cases} \quad (18)$$

In this paper, the rounded value of $E[S\_LP_{minor}]$ is defined as the approximate carry value propagated from LP_{minor} to LP_{major}. As an example, from Fig. 3 (d), it can be seen that L_3 (= 2) is larger than M_3 (= 0 or 1). Thus, by (17), the approximate carry value is decided as

$$a\_carry = \left\lceil \frac{1}{2}(Z_3 + Z_2 + Z_1 + Z_0) \right\rceil, \quad (19)$$

where $\lceil t \rceil$ defines the smallest integer greater than or equal to t. In general, (19) can be implemented as shown in Fig. 4 (a). However, since the number of nonzero Z_i signals in Table 2 is either 3 or 4, the values of a_carry signals for the three coefficients are always 2. Thus, in this case, no additional hardware is required for the generation of the approximate carry signals as shown in Fig. 4 (b).

The proposed error compensation bias is computed using S_LP_{major} and a_carry. When the number of nonzero Z_i signals is odd, the effect of rounding error can be large in the computation of a_carry signal. To alleviate this problem, we propose an error compensation bias for fixed-width GCSD multipliers as follows:

If N_{NZPP} = odd, then

$$\sigma_{prop} = \left\lfloor \frac{S\_LP_{major} + a\_carry}{2} \right\rfloor,$$

else

$$\sigma_{prop} = \left\lceil \frac{S\_LP_{major} + a\_carry}{2} \right\rceil, \quad (20)$$

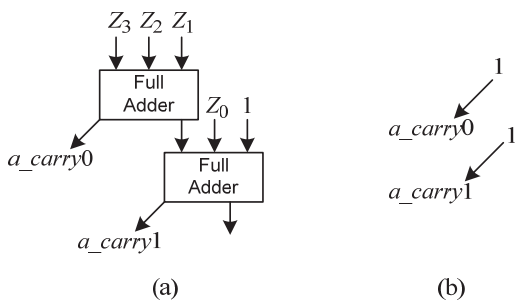

Fig. 4 Approximate carry generation circuits.

where N_{NZPP} is the number of nonzero Z_i signals.

In GCSD multipliers, the number of the different coefficients in a group is assumed to be small. Thus, the coefficient selection signals (or, address) need only a few bits. Using this property, the approximate carry signals can be designed using the address bits. For example, if the word length N_w of the coefficients in Table 1 is 12, the approximate carry value can be obtained as follows:

$$a\_carry = \begin{cases} 2, & \text{for } \cos(\pi/8),\ a_1a_0 = 00, \\ 2, & \text{for } \cos(\pi/4),\ a_1a_0 = 01, \\ 1, & \text{for } \sin(\pi/8),\ a_1a_0 = 10, \end{cases} \quad (21)$$

where a_1a_0 means the address for the coefficients. Since the maximum value of a_carry is 2 in (21), a_carry can be represented as

$$a\_carry = 2a\_carry_1 + a\_carry_0. \quad (22)$$

From (21), the following expression can be obtained using Karnaugh map:

$$a\_carry_1 = \overline{a_1}, \quad a\_carry_0 = a_1\overline{a_0}. \quad (23)$$

In general, the implementation using address bits requires smaller area compared with the implementation based on coefficient control signals when the number of coefficients is small.

4. Performance Comparisons

To evaluate the performance of the proposed fixed-width GCSD multiplier, we compute the maximum absolute error ε_{max}, the average of absolute error ε_{avg} and the variance of error ε_{var} for all the possible 2^W input values of X as follows

$$\varepsilon_{abs} = |P_I - P_Q|, \quad \varepsilon_{max} = \max(|\varepsilon_{abs}|), \quad \varepsilon_{avg} = 2^{-W} \sum \varepsilon_{abs}, \quad \varepsilon_{var} = 2^{-W} \sum (\varepsilon - \varepsilon_{avg})^2. \quad (24)$$

Table 3 shows the simulation results of the fixed-width GCSD multiplier for the input word size $W = 14$. Let M_{true} and M_{post} denote the fixed-width multiplier by the true rounding and post-truncation, respectively. For the computation of M_{true} and M_{post}, all the possible bits are required during the multiplication and the final product is obtained by rounding or truncating the least significant $(W - 1)$-bits from the exact $(2W - 1)$-bit result. Also, Table 4 compares the Synopsys simulation results using MagnaChip 0.18-μm CMOS technology. Notice that, compared with the fixed-width modified Booth multiplier, the proposed GCSD fixed-width multiplier can reduce about 10% average error. In addition, the proposed multiplier leads to 29%, 36% and 9% reduction in area, power consumption and propagation delay, respectively.

As another example, the proposed algorithm is applied to the following coefficients used in the pulse-shaping filter design for CDMA [11]:

$a_1a_0(00)$: 1111111010,
$a_1a_0(01)$: 1111111000,
$a_1a_0(10)$: 1111110111,
$a_1a_0(11)$: 1111111100. (25)

Figure 5 shows the partial product array corresponding to the GCSD multiplier. Table 5 compares the error performances of several methods and Table 6 compares the Synopsys simulation results. In this case, although the error performances of the proposed method and the method in [8]

Table 3 Comparison of the error performances for FFT applications.

	ε_{max} ($\times 2^{-(W-1)}$)	ε_{avg} ($\times 2^{-(W-1)}$)	ε_{var} ($\times 2^{-2(W-1)}$)
M_{true}	0.5 (1)	0.25 (1)	0.0208 (1)
M_{post}	0.9999 (2)	0.4990 (2)	0.0833 (4)
$M_{[6]}$	2.6328 (5.27)	0.8712 (3.48)	0.4370 (20.98)
$M_{[8]}$	1.2070 (2.41)	0.3369 (1.35)	0.0581(2.79)
M_{prop}	1.0508 (2.10)	0.3025 (1.21)	0.045 (2.13)

Table 4 Synopsys simulation results for FFT applications.

	Area (cell)	Power (mW)	Delay (ns)
$[8]_{fixed}$	8159 (1)	15.31 (1)	10.25 (1)
GCSD	7577 (0.93)	12.93 (0.84)	11.89 (1.16)
$GCSD_{fixed}$	5779 (0.71)	9.84 (0.64)	9.32 (0.91)

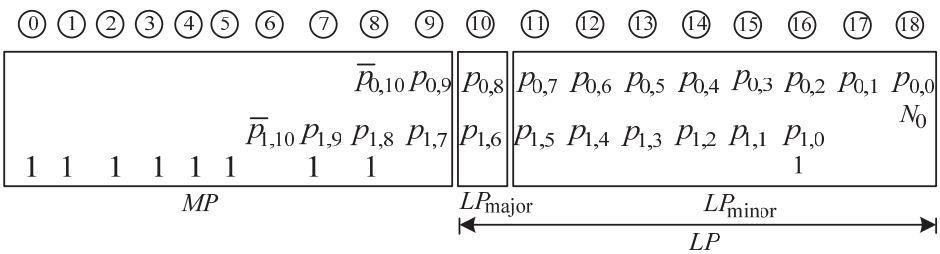

Fig. 5 Partial product array of the GCSD multiplier for pulse-shaping filters.

Table 5 Comparison of the error performances for pulse-shaping filters.

	ε_{max} ($\times 2^{-(W-1)}$)	ε_{avg} ($\times 2^{-(W-1)}$)	ε_{var} ($\times 2^{-2(W-1)}$)
M_{true}	0.5 (1)	0.25 (1)	0.0208 (1)
M_{post}	0.9980 (2)	0.4963 (1.98)	0.0833 (1.85)
$M_{[6]}$	2.2188 (4.44)	1.0080 (4.03)	0.4061 (19.47)
$M_{[8]}$	1 (2)	0.2813 (1.13)	0.0355 (1.71)
M_{prop}	0.9824 (1.96)	0.2886 (1.15)	0.0386 (1.85)

Table 6 Synopsys simulation results for pulse-shaping filters.

	Area (cell)	Power (mW)	Delay (ns)
$[8]_{fixed}$	6116 (1)	10.92 (1)	9.56 (1)
GCSD	2343 (0.38)	3.72 (0.34)	8.14 (0.85)
$GCSD_{fixed}$	1338 (0.22)	1.77 (0.16)	4.46 (0.47)

are almost the same, the proposed multiplier leads to 78%, 84% and 53% reduction in area, power consumption and propagation delay, respectively.

5. Conclusions

In this paper, an efficient error compensation method for fixed-width GCSD multipliers is proposed. To compute the error compensation bias more accurately, the encoded signals from the GCSD multiplier are used for the bias generation.

The simulation results show that the proposed method leads to significant reduction in area, power consumption, and delay time compared with the fixed-width modified Booth multipliers.

References

[1] Y.E. Kim, S.H. Cho, and J.G. Chung, "Modified CSD group multiplier design for predetermined coefficient groups," Proc. IEEE ISCAS 2008, pp.3362–3365, May 2008.
[2] S.W. Reitwiesner, "Binary arithmetic," Advances in Computers, pp.231–308, 1966.
[3] S.S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-efficient multipliers for digital signal processing applications," IEEE Trans. Circuits Syst. II, vol.43, no.2, pp.90–94, Feb. 1996.
[4] J.M. Jou, S.R. Kuang, and R.D. Chen, "Design of a low-error fixed-width multipliers for DSP applications," IEEE Trans. Circuits Syst. II, vol.46, no.6, pp.836–842, June 1999.
[5] L.D. Van and C.C. Yang, "Generalized low-error area-efficient fixed-width multipliers," IEEE Trans. Circuits Syst. I, vol.52, no.8, pp.1608–1619, Aug. 2005.
[6] M.A. Song, L.D. Van, and S.Y. Kuo, "Adaptive low-error fixed-width Booth multipliers," IEICE Trans. Fundamentals, vol.E90-A, no.6, pp.1180–1187, June 2007.
[7] H.A. Huang, Y.C. Liao, and H.C. Chang, "A self-compensation fixed-width Booth multiplier and its 128-point FFT applications," Proc. IEEE ISCAS 2006, pp.3538–3541, May 2006.
[8] K.J. Cho, K.C. Lee, J.G. Chung, and K.K. Parhi, "Design of low-error fixed-width modified Booth multiplier," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.12, no.5, pp.522–531, May 2004.
[9] 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, no.12, pp.984–993, Dec. 2003.
[10] J.Y. Oh and M.S. Lim, "New radix-2 to the 4th power pipeline FFT processor," IEICE Trans. Electron., vol.E88-C, no.8, pp.1740–1746, Aug. 2005.
[11] Y.E. Kim, K.J. Cho, and J.G. Chung, "Low power small area modified Booth multiplier design for predetermined coefficients," IEICE Trans. Fundamentals, vol.E90-A, no.3, pp.694–697, March 2007.

Yong-Eun Kim received his MS and PhD in Information and Communication Engineering from Chonbuk National University, Jeonju, Korea in 2007 and 2010, respectively. He is a member of researchers with Korea Automotive Technology Institute (KATCH). His research interests are in the area of VLSI implementation for digital signal processing and communication systems.

Kyung-Ju Cho received his MS and PhD in Information and Communication Engineering from Chonbuk National University, Jeonju, Korea in 2002 and 2006, respectively. He was a Post-doctor with Advanced Graduate Education Center of Jeonbuk for Electronics and Information Technology-BK21 from 2006 to 2008. He is a member of research engineers with Korea Association of Aids to Navigation (KAAN). His research interests are in the area of VLSI architectures and algorithms for digital signal processing and communication systems with emphasis on radio aids to navigation.

Jin-Gyun Chung received his MS and PhD in Electrical Engineering from the University of Minnesota, Minneapolis, Minnesota in 1991 and 1994, respectively. Since 1995, he has been with the Division of Electronics and Information Engineering at Chonbuk National University. His research interests are in the area of VLSI architectures and algorithms for signal processing and communication systems, which include the design of high-speed and low-power algorithms for digital filters, DSL systems, OFDM systems and ultrasonic NDE systems.

Xinming Huang is an Assistant Professor in the Department of Electrical and Computer Engineering at Worcester Polytechnic Institute (WPI), Worcester, MA. He received the Ph.D. degree in electrical engineering from Virginia Polytechnic Institute and State University, Blacksburg, VA in 2001. He was a member of technical staff with the wireless advanced technology laboratory, Bell Labs of Lucent Technologies, from 2001 to 2003. He was also an Assistant Professor with the Department of Electrical Engineering at the University of New Orleans from 2003 to 2006. He was among the recipients of the DARPA/MTO young faculty award in 2007, the IBM faculty fellowship award in 2004, and the central Bell Labs annual excellence and teamwork award in 2002. His research interests are in the areas of circuit design and system architecture, with emphasis on reconfigurable computing, wireless communications, and embedded systems.