A High-Accuracy Adaptive Conditional-Probability Estimator for Fixed-Width Booth Multipliers

Yuan-Ho Chen, *Student Member, IEEE*, and Tsin-Yuan Chang, *Member, IEEE*

***Abstract*—In this paper, a single compensation formula of adaptive conditional-probability estimator (ACPE) applied to fixed-width Booth multiplier is proposed. Based on the conditional-probability theory, the ACPE can be easily applied to large length Booth multipliers (such as 32-bit or larger) for achieving a higher accuracy performance. To consider the trade-off between accuracy and area cost, the ACPE provides varying column information w to adjust the accuracy with respect to system requirements. The 16-bit ACPE Booth multiplier with $w = 3$ reduces 28.9% silicon area with only 0.39 dB signal-to-noise ratio (SNR) loss when compared with post-truncated (P-T) Booth multiplier. Furthermore, the ACPE Booth multipliers are applied to two-dimensional (2-D) discrete cosine transform (DCT) to evaluate the system performance. Implemented in a TSMC 0.18 μm CMOS process, the DCT core with ACPE ($w = 3$) can save 14.3% area cost with only 0.48 dB peak-signal-to-noise-ratio (PSNR) penalty compared to P-T method.**

***Index Terms*—Adaptive conditional-probability estimator (ACPE), Booth multiplier, Discrete cosine transform (DCT), fixed-width.**

I. Introduction

FIXED-WIDTH multipliers are the important components in digital signal processing (DSP) systems [1]–[5]. In general, they truncate the least significant half part directly to generate the output with the same word length as input, and the computation errors occur from the operators that perform the truncation. Therefore, many compensation methods are presented to reduce the truncation errors in Baugh-Wooley array multipliers [6]–[14] and Booth multipliers [15]–[22].

Booth multipliers are popular due to the reduced elements of the partial products [23]–[27]. The fixed-width Booth multiplier with the best accuracy is the post-truncated (P-T) multiplier, which uses rounding off operator after calculating all products. However, the P-T multiplier consumes large silicon area in very large scale integration (VLSI) designs. In order to reduce the area cost, the direct-truncated (D-T) multiplier chops the least significant half partial products directly. Thus, a large number of truncation errors occur in the D-T Booth multiplier. For this reason, the adaptive compensation methods for fixed-width Booth multipliers are presented to reduce the truncation errors while still reserving the merits of low area cost [15]–[22]. A low cost compensation bias in [15] is approximated from a linear regression. The errors can be reduced hugely in [15] compared with the D-T Booth multiplier. Song *et al.* present two types (Type 1 and Type 2 in [16]) of binary threshold to compensate the truncation errors adaptively, and a little improvement in accuracy is observed. To improve the accuracy, Huang *et al.* [17] run some time-consuming exhaustive simulations and find out the compensation functions to fit them. The compensation functions are actually not fit well in [17]. Thus, the combination circuits from the truth table are addressed in [18] by using Karnaugh maps, and the compensation circuits can match the statistical compensation value exactly. According to the applications of larger bit width fixed-width multipliers, like pseudo random number generator (PRNG) [5], huge number of time will be consumed to establish the compensation function based on exhaustive simulation [15]–[18]. Therefore, a compensation function is presented in binary-sign-digit (BSD) Booth multipliers by using condition-probability method [19]. The conditional bits in [19] are not chosen well. Thus, the compensation circuits can not improve the performance in accuracy when compared with existing works. The probabilistic estimation bias is presented [20] to replace the time-consuming exhaustive simulation methods and it also exhibits good accuracy performance.

Recently, many works use more information from Booth encoder [21], [22] and partial products [16] to achieve higher accuracy performance. In [21], taking more information provided by Booth encoder, the compensation circuits can reduce the truncation errors. Wang *et al.* [22] further improve the work of [21], and the truncation errors can be alleviated evidently. Nevertheless, the area cost is increased from the extra information of compensation circuits. There is a trade-off between accuracy and area cost. Song *et al.* introduce the column information w to provide more choices between accuracy and area cost [16]. The compensation function is also established by the exhaustive simulation to adjust the accuracy with respect to system requirements based on varying w.

In this paper, a high-accuracy adaptive conditional-probability estimator (ACPE) is proposed to be applied in fixed-width Booth multipliers. Instead of the time-consuming exhaustive simulation of the previous works [15]–[18], [21], [22], the ACPE is derived from the conditional-probability theory. Thus, the ACPE can be easily applied to large length Booth multipliers (such as 32-bit or larger). Considering system

Manuscript received January 21, 2011; revised April 29, 2011, June 28, 2011, and July 21, 2011; accepted July 22, 2011. Date of publication October 13, 2011; date of current version February 24, 2012. This work was supported in part by the National Science Council under project number NSC 100-2221-E-007-092. This paper was recommended by Associate Editor C. H. Chang.

The authors are with the Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan. (e-mail: yhchen@larc.ee.nthu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2011.2167275

TABLE I
MODIFIED BOOTH ENCODER

y_{2i+1}	y_{2i}	y_{2i-1}	y'_i	nz_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	2	1
1	0	0	-2	1
1	0	1	-1	1
1	1	0	-1	1
1	1	1	0	0

TABLE II
PARTIAL PRODUCTS FOR 8-BIT BOOTH ENCODER

y'_i	p_{8i}	p_{7i}	p_{6i}	p_{5i}	p_{4i}	p_{3i}	p_{2i}	p_{1i}	p_{0i}	n_i
0	0	0	0	0	0	0	0	0	0	0
1	x_7	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	0
-1	$\bar{x_7}$	$\bar{x_7}$	$\bar{x_6}$	$\bar{x_5}$	$\bar{x_4}$	$\bar{x_3}$	$\bar{x_2}$	$\bar{x_1}$	$\bar{x_0}$	1
2	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	0	0
-2	$\bar{x_7}$	$\bar{x_6}$	$\bar{x_5}$	$\bar{x_4}$	$\bar{x_3}$	$\bar{x_2}$	$\bar{x_1}$	$\bar{x_0}$	1	1

requirements, the ACPE also provides varying column information $(w \geq 1)$ to adjust the system accuracy. Furthermore, the ACPE Booth multipliers are also applied to a two-dimensional (2-D) discrete cosine transform (DCT) [2] to demonstrate the performance of real applications. As a result, the ACPE can reduce 14.3% area cost compared to P-T method with only 0.48 dB peak-signal-to-noise-ratio (PSNR) lost. Consequently, the proposed ACPE saves the establishment time of compensation circuit, provides varying column information w, and achieves the high accuracy performance in a single compensation function.

The rest of this paper is organized as follows. In Section II, the background of the fixed-width modified Booth multiplier is given. The proposed ACPE formula and architecture are discussed in Section III. Section IV depicts the comparisons of accuracy and area and further demonstrates the performance of DCT with the proposed compensation circuit. Finally, conclusions of this paper are drawn in Section V.

II. FIXED-WIDTH MODIFIED BOOTH MULTIPLIER

Modified Booth encoding is popular for reducing the number of partial products [28]. The $2L$-bit product P can be expressed in two's complement representation as follows:

$$X = -x_{L-1}2^{L-1} + \sum_{i=0}^{L-2} x_i \cdot 2^i$$
$$Y = -y_{L-1}2^{L-1} + \sum_{i=0}^{L-2} y_i \cdot 2^i$$
$$P = X \times Y. \quad (1)$$

Three concatenated inputs y_{2i+1}, y_{2i}, and y_{2i-1} can be mapped into y'_i by using Booth encoder, as tabulated in Table I, where nonzero code nz_i is a 1-bit digit whose value is determined by whether y'_i equals to zero. There are $Q = L/2$ rows in partial product array producing by Booth encoding, where L is an even number. Taking 8×8 Booth multiplier as

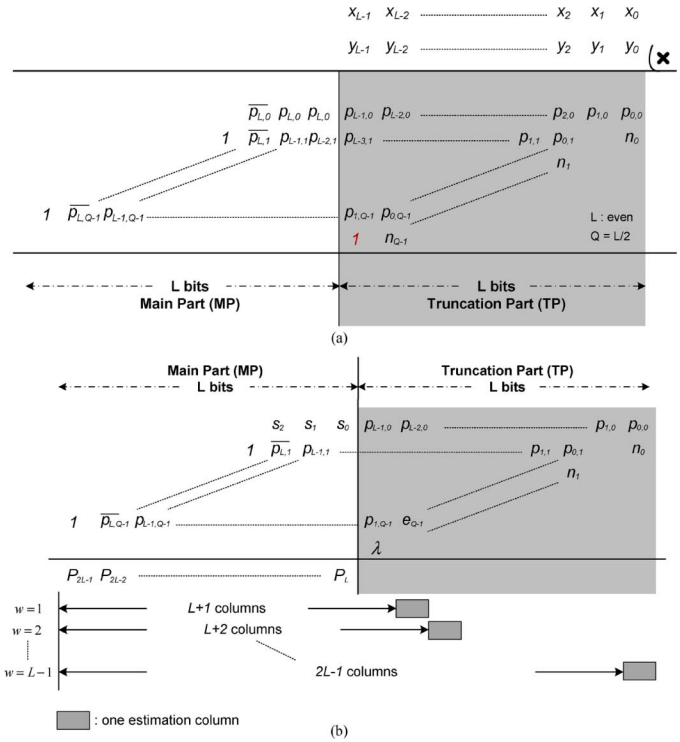


Fig. 1. Algorithms of $L \times L$ fixed-width Booth multipliers. (a) Conventional P-T Booth algorithm of $L \times L$ Booth multiplier. (b) Modified algorithm of $L \times L$ Booth multiplier.

TABLE III
MAPPING TABLE OF MODIFIED BOOTH ALGORITHM

n_{Q-1}	$p_{0,Q-1}$	$p_{L,0}$	s_2	s_1	s_0	λ	e_{Q-1}
0	0	0	1	0	0	1	0
0	0	1	0	1	1	1	0
0	1	0	1	0	0	1	1
0	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1
1	0	1	0	1	1	1	1
1	1	0	1	0	1	0	0
1	1	1	1	0	0	0	0

an example, the corresponding partial products represented in input y'_i are tabulated in Table II. Due to the two's complement computation, n_i is equal to 1 when y'_i is negative; otherwise, n_i is equal to 0. According to the encoder table (Table I) and partial product table (Table II), the partial products of $L \times L$ fixed-width post-truncated (P-T) Booth multiplier are shown in Fig. 1(a). For the fixed-width Booth multiplier design, some of products are truncated by using a rounding operator to hold the data length fixed in L-bit. Therefore, an extra one binary bit 1 added into the most significant column of truncation part (TP) in Fig. 1(a), which indicates the rounding off operation of the P-T Booth multipliers, and the conventional P-T Booth algorithm in Fig. 1(a) can be further modified as Fig. 1(b) according to previous work [29]. Table III maps the partial products $\{n_{Q-1}, p_{0,Q-1}, p_{L,0}\}$ in Fig. 1(a) to $\{s_2, s_1, s_0, \lambda, e_{Q-1}\}$ in Fig. 1(b) by the following operations:

$$\{s_2, s_1, s_0, \lambda, e_{Q-1}\} = \{\overline{p_{L,0}}, p_{L,0}, p_{L,0}, 1, p_{0,Q-1}\} + \{0, 0, 0, 0, n_{Q-1}\}. \quad (2)$$

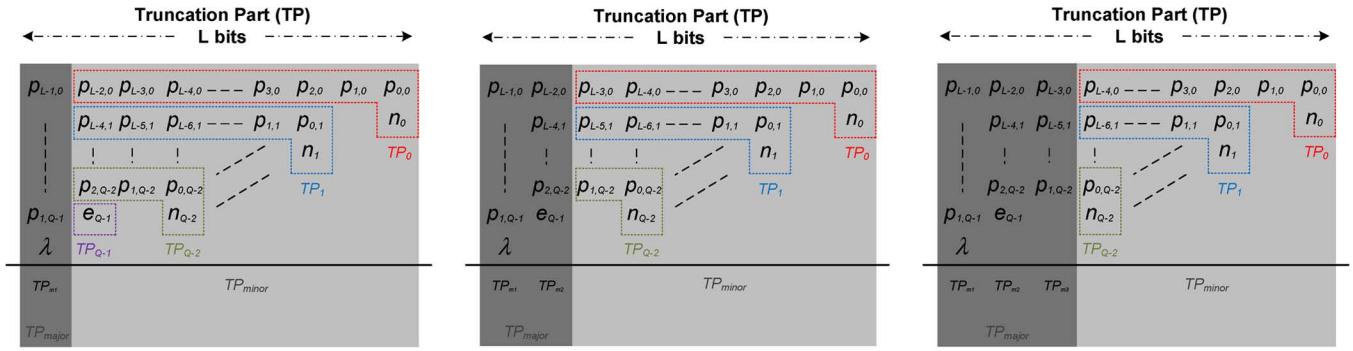


Fig. 2. The partition of truncation part with $w = 1, 2, 3$. (a) Partition with $w = 1$. (b) Partition with $w = 2$. (c) Partition with $w = 3$.

The partial product array in Fig. 1(b) also can be divided into two parts: the main part (MP) including the most significant L columns, and the truncation part (TP) including the least significant L columns. Besides, the column information w is included to adjust accuracy with respect to system requirements, and w means that $L + w$ most significant columns (MSCs) are calculated and the $(L + w + 1)^{th}$ MSC is chosen to estimate the compensation values. Therefore, $L + w + 1$ MSCs are used to produce the results. Taking $w = 2$ as an example, the results are calculated by the partial products of most significant $L + 3 (= L + w + 1)$ columns.

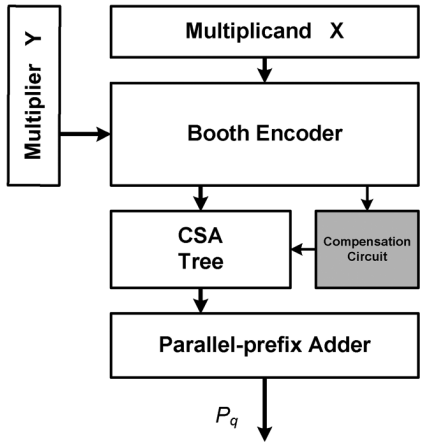


Fig. 3. The whole architecture of the proposed ACPE Booth multiplier.

III. Proposed Adaptive Conditional Probability Estimator (ACPE)

When fixed-width multiplication is concern, the quantized product P_q can be expressed as follows:

$$\begin{aligned} P \approx P_q &= MP + TP \\ &= MP + \sigma \cdot 2^L \end{aligned} \tag{3}$$

where σ represents the bias of adaptive conditional-probability estimator (ACPE) which can be decomposed further into TP_{major} and TP_{minor} parts as following equations.

$$\sigma = \lfloor TP_{\text{major}} + TP_{\text{minor}} \rfloor \tag{4}$$

where $\lfloor \bullet \rfloor$ rounds down the elements $\bullet$. The TP_{major} and TP_{minor} are the major and the minor compensation part in TP, respectively. The TP_{major} provides truly information to estimate ACPE, and the TP_{minor} contributes compensation bias to MP based on conditional probability estimation. Therefore, the compensation bias σ can be calculated by obtaining TP_{major} and estimating TP_{minor}.

A. Derived Adaptive Conditional Probability Formula

According to the column information w, the TP_{major} and TP_{minor} can be further derived as the following equations:

$$\begin{aligned} TP_{\text{major}} &= TP_{m1} + TP_{m2} + \cdots + TP_{mw} \\ TP_{\text{minor}} &= TP_0 + TP_1 + \cdots + TP_{Q-1-\lfloor \frac{w}{2} \rfloor} \end{aligned} \tag{5}$$

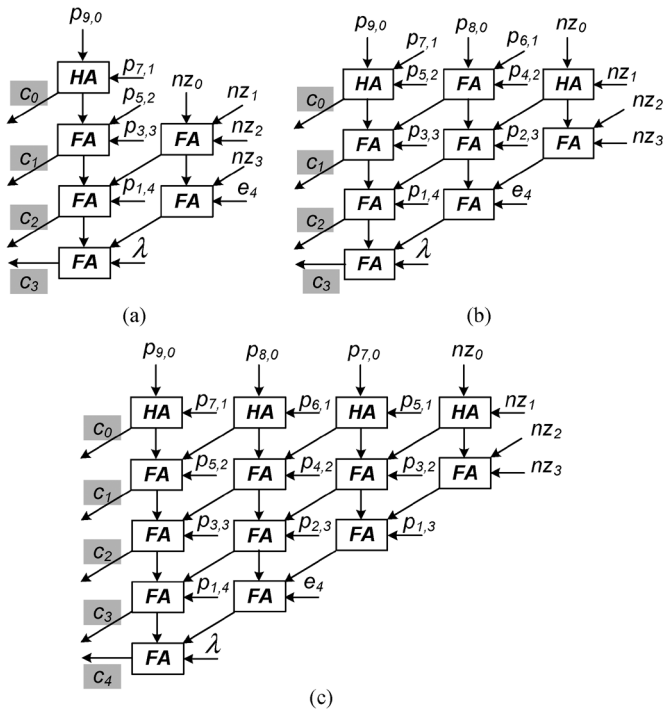


Fig. 4. Compensation circuits of ACPE Booth multipliers for $w = 1, 2, 3$. (a) $w = 1$. (b) $w = 2$. (c) $w = 3$.

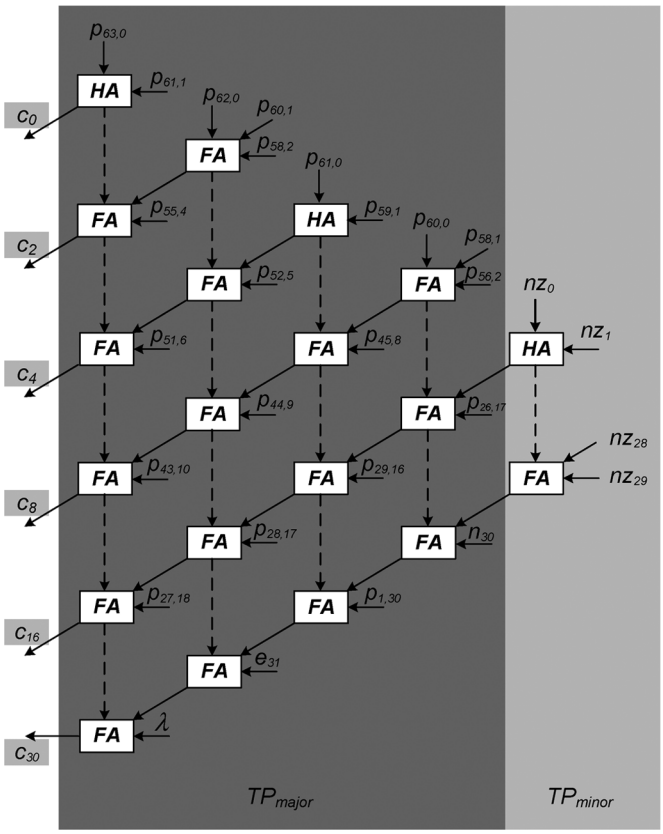


Fig. 5. The proposed 64-bit ACPE compensation circuit with $w = 4$.

where TP_{major} is partitioned into the parts of TP_{mi}, $(w \geq i \geq 1)$:

$$\begin{aligned} TP_{m1} &= 2^{-1}(p_{L-1,0} + p_{L-3,1} + \cdots + p_{1,Q-1} + \lambda) \\ TP_{m2} &= 2^{-2}(p_{L-2,0} + p_{L-4,1} + \cdots + e_{Q-1}) \\ &\vdots \\ TP_{m(L-1)} &= 2^{-(L-1)}p_{1,0}. \end{aligned} \tag{6}$$

TP_{minor} is also partitioned into the parts of TP_j, $(Q-1-\lfloor w/2 \rfloor \geq j \geq 0)$:

$$\begin{aligned} TP_0 &= 2^{-L}(p_{0,0} + n_0) + \cdots + 2^{-1-w}p_{L-1-w,0} \\ TP_1 &= 2^{-(L-2)}(p_{0,1} + n_1) + \cdots + 2^{-1-w}p_{L-3-w,1} \\ &\vdots \\ TP_{Q-2} &= 2^{-4}(p_{0,Q-2} + n_{Q-2}) + \cdots + 2^{-1-w}p_{3-w,Q-2} \\ TP_{Q-1} &= 2^{-2}e_{Q-1}. \end{aligned} \tag{7}$$

and the range of each TP_j with column information w is

$$\begin{aligned} TP_0 &: w \leq L-1 \\ TP_1 &: w \leq L-3 \\ &\vdots \\ TP_{Q-2} &: w \leq 3 \\ TP_{Q-1} &: w = 1. \end{aligned} \tag{8}$$

The partition of the TP_{major} and TP_{minor} with $w = 1, 2, 3$ is illustrated as Fig. 2. Each TP_j partition is closely related to e_{Q-1} and the nonzero code nz_i denoted in the Booth encoder in Tables I and III. Therefore, the conditional expected value with corresponding weight of each TP_j partition can be derived as follows (see Appendix):

$$\begin{aligned} E[TP_0|z_0 = 1] &\simeq \frac{z_0}{2} \cdot \left(\frac{1}{2}\right)^w \\ E[TP_1|z_1 = 1] &= \frac{z_1}{2} \cdot \left(\frac{1}{2}\right)^w \\ &\vdots \\ E[TP_{Q-1}|z_{Q-1} = 1] &= \frac{z_{Q-1}}{2} \cdot \left(\frac{1}{2}\right)^w \end{aligned} \tag{9}$$

where $E[\bullet]$ is the expected operator, the nonzero conditional code z_j is defined as follows:

$$\begin{cases} z_j = nz_j, & Q-2 \geq j \geq 0 \\ z_j = e_{Q-1}, & j = Q-1. \end{cases}$$

Based on the conditional expected value of each TP_j, the conditional expected value of TP_{minor} can be obtained:

$$\begin{aligned} E[TP_{\text{minor}}|\text{nonzero}] &= \sum_{j=0}^{Q-1-\lfloor \frac{w}{2} \rfloor} E[TP_j|z_j = 1] \\ &= \left(\frac{1}{2}\right)^w \sum_{j=0}^{Q-1-\lfloor \frac{w}{2} \rfloor} \frac{z_j}{2} \\ &= 2^{-w}\sigma_w \end{aligned} \tag{10}$$

where

$$\sigma_w = \sum_{j=0}^{Q-1-\lfloor \frac{w}{2} \rfloor} \frac{z_j}{2}. \tag{11}$$

By substituting the TP_{minor} in (4) with the conditional expected value in (10), the adaptive compensation bias σ can be rewritten as the following equations:

$$\begin{aligned} \sigma &= \lfloor TP_{\text{major}} + TP_{\text{minor}} \rfloor \\ &= \lfloor TP_{\text{major}} + 2^{-w}\sigma_w \rfloor \\ &= \left\lfloor TP_{\text{major}} + 2^{-w} \sum_{j=0}^{Q-1-\lfloor \frac{w}{2} \rfloor} \frac{z_j}{2} \right\rfloor. \end{aligned} \tag{12}$$

B. Architecture Design of ACPE Booth Multiplier

Based on the proposed ACPE formula in (12), the adaptive compensation bias σ can be calculated by obtaining word-length L and column information w. The architecture of the proposed ACPE Booth multiplier is designed by using tree-based carry-save reduction followed by parallel-prefix carry-propagate addition architecture. The whole architecture of the proposed ACPE Booth multiplier is shown in Fig. 3. According to the mapping Tables I–III, Booth encoder will generate the partial products

TABLE IV
COMPARISONS OF AVERAGE ERROR $\bar{\varepsilon}$ WITH VARIOUS METHODS

w	Methods	$L=8$	$L=10$	$L=12$	$L=14$	$L=16$
$\times$	D-T	1.5010	1.8752	2.2501	2.6250	3.0000
0	Jou *et al.* [15]	0.0010	0.0002	0.0001	1.5×10^{-5}	3.8×10^{-6}
	Song *et al.* Type2 [16]	0.0208	0.0077	0.0028	0.0011	0.0004
	Juang *et al.* [19]	−0.0825	−0.0799	−0.0770	−0.0753	−0.0739
	PEB [20]	0.0000	0.1875	−0.1249	0.0625	−0.2500
1	Song *et al.* Type1 [16]	0.1758	0.2028	0.2203	0.2314	0.2384
	Cho *et al.* [21]	0.1328	0.1211	0.1270	0.1240	0.1255
	Wang *et al.* [22]	0.0078	−0.0039	0.0020	−0.0010	0.0005
	ACPE	**-0.1152**	**-0.1284**	**-0.1229**	**-0.1259**	**-0.1245**
2	Song *et al.* Type1 [16]	0.0649	0.0871	0.1012	0.1101	0.1157
	Cho *et al.* [21]	−0.1943	−0.1833	−0.1894	−0.1865	−0.1880
	ACPE	**-0.0537**	**-0.0662**	**-0.0605**	**-0.0635**	**-0.0620**
3	Song *et al.* Type1 [16]	−0.0085	0.0004	0.0061	0.0097	0.0119
	Cho *et al.* [21]	−0.0977	−0.0918	−0.0947	−0.0933	−0.0940
	ACPE	**-0.0273**	**-0.0332**	**-0.0303**	**-0.0317**	**-0.0310**
L	P-T	−0.0078	−0.0024	−0.0007	−0.0002	−0.0001

TABLE V
COMPARISONS OF MAXIMUM ABSOLUTE ERROR $\varepsilon_{\max}$ WITH VARIOUS METHODS

w	Methods	$L=8$	$L=10$	$L=12$	$L=14$	$L=16$
$\times$	D-T	4.0000	5.0000	6.0000	7.0000	8.0000
0	Jou *et al.* [15]	1.7305	2.1299	2.5300	2.9300	3.3300
	Song *et al.* Type2 [16]	1.7305	2.1299	2.5300	2.9300	3.3300
	Juang *et al.* [19]	1.9180	2.2607	3.0867	3.4201	3.9175
	PEB [20]	1.5000	2.0000	2.0000	2.5000	2.5000
1	Song *et al.* Type1 [16]	1.1641	1.3652	1.5649	1.7650	1.9650
	Cho *et al.* [21]	1.5000	1.5000	2.0000	2.0000	2.5000
	Wang *et al.* [22]	1.1680	1.5000	1.6667	2.0000	2.1667
	ACPE	**1.1680**	**1.5000**	**1.6667**	**2.0000**	**2.1667**
2	Song *et al.* Type1 [16]	0.7344	0.8369	0.9363	1.0364	1.1364
	Cho *et al.* [21]	1.0000	1.0000	1.2500	1.2500	1.5000
	ACPE	**0.7500**	**1.0000**	**1.0000**	**1.2500**	**1.2500**
3	Song *et al.* Type1 [16]	0.6172	0.6660	0.7163	0.7662	0.8163
	Cho *et al.* [21]	0.7500	0.7500	0.8750	0.8750	1.0000
	ACPE	**0.6250**	**0.7500**	**0.7500**	**0.8750**	**0.8750**
L	P-T	0.5000	0.5000	0.5000	0.5000	0.5000

as the formate in Fig. 1(b). The compensation circuit is implemented to compute compensation bias σ based on (12), and then the carry-save-adder (CSA) tree and Brent-Kung parallel-prefix adder sum up the partial products of MP and σ based on (3). The CAS tree is implemented by using the 4-2 or 3-2 compressors [24]–[27].

The compensation circuits are the main difference modules in varying w. Taking 10×10 $(L = 10)$ as an example, when $w = 1$ and $w = 2$, the ACPE compensation circuit has four compensation biases (c_0, c_1, c_2, c_3) as shown in Fig. 4(a) and 4(b). Due to the fact that more partial product information is applied, the ACPE compensation circuit for $w = 3$ has more full-adders (FAs) and half-adders (HAs) than those in $w = 1$ and $w = 2$. In this way, the ACPE compensation circuit for $w = 3$ has five compensation biases $(c_0, c_1, c_2, c_3, c_4)$ as shown in Fig. 4(c).

Instead of heuristic design of time-consuming exhaustive simulation methods in previous works [15]–[18] and [21], [22], the proposed conditional-probability method can save huge time consumption as designing large length $(L > 16)$ Booth multiplier. Taking the large length as an example ($L = 64$ and $w = 4$), the ACPE compensation circuit can be easily designed according to (12) as the CSA array expression and shown in Fig. 5.

IV. COMPARISONS AND DISCUSSIONS

In this section, some important issues, such as accuracy, area, delay, power, established time of compensation bias,

TABLE VI
COMPARISONS OF ERROR VARIANCE ε_{var} WITH VARIOUS METHODS

w	Methods	$L=8$	$L=10$	$L=12$	$L=14$	$L=16$
$\times$	D-T	0.4350	0.5397	0.6441	0.7483	0.8524
0	Jou *et al.* [15]	0.2717	0.3368	0.4019	0.4670	0.5321
	Song *et al.* Type2 [16]	0.2499	0.3249	0.3960	0.4642	0.5309
	Juang *et al.* [19]	0.2767	0.3148	0.3589	0.4017	0.4477
	PEB [20]	0.1830	0.2083	0.2338	0.2599	0.2860
1	Song *et al.* Type1 [16]	0.1150	0.1302	0.1474	0.1656	0.1841
	Cho *et al.* [21]	0.1487	0.1641	0.1788	0.1934	0.2078
	Wang *et al.* [22]	0.1367	0.1542	0.1671	0.1821	0.1961
	ACPE	**0.1237**	**0.1379**	**0.1521**	**0.1663**	**0.1806**
2	Song *et al.* Type1 [16]	0.0882	0.0914	0.0954	0.0998	0.1045
	Cho *et al.* [21]	0.0933	0.0973	0.1010	0.1047	0.1083
	ACPE	**0.0940**	**0.0977**	**0.1012**	**0.1048**	**0.1083**
3	Song *et al.* Type1 [16]	0.0850	0.0864	0.0877	0.0890	0.0903
	Cho *et al.* [21]	0.0861	0.0874	0.0884	0.0893	0.0902
	ACPE	**0.0866**	**0.0876**	**0.0885**	**0.0893**	**0.0902**
L	P-T	0.0833	0.0833	0.0833	0.0833	0.0833

TABLE VII
COMPARISONS OF MULTIPLIER SIGNAL-TO-NOISE RATIO SNR WITH VARIOUS METHODS

w	Methods	$L=8$	$L=10$	$L=12$	$L=14$	$L=16$
$\times$	D-T	22.29	32.54	43.10	53.87	64.81
0	Jou *et al.* [15]	32.24	43.35	54.62	66.01	77.49
	Song *et al.* Type2 [16]	32.60	43.50	54.69	66.04	77.50
	Juang *et al.* [19]	32.06	43.55	55.04	66.60	78.18
	PEB [20]	33.96	44.76	56.70	68.49	79.32
1	Song *et al.* Type1 [16]	34.94	46.29	57.74	69.30	80.93
	Cho *et al.* [21]	34.37	46.10	57.76	69.51	81.25
	Wang *et al.* [22]	35.22	46.74	58.43	70.10	81.82
	ACPE	**35.21**	**46.74**	**58.43**	**70.10**	**81.82**
2	Song *et al.* Type1 [16]	36.93	48.67	60.43	72.21	84.03
	Cho *et al.* [21]	35.41	47.45	59.30	71.26	83.17
	ACPE	**36.72**	**48.53**	**60.46**	**72.34**	**84.25**
3	Song *et al.* Type1 [16]	37.28	49.26	61.23	73.20	85.18
	Cho *et al.* [21]	36.77	48.81	60.78	72.79	84.79
	ACPE	**37.17**	**49.14**	**61.15**	**73.15**	**85.15**
L	P-T	37.37	49.41	61.46	73.50	85.54

and applications, for the fixed-width Booth multipliers are discussed.

A. Accuracy and Circuit Characteristics Comparisons

For the comparison of accuracy, the average error $\bar{\varepsilon}$, the maximum absolute error $\varepsilon_{\max}$, the variance ε_{var}, and the system signal-to-noise ratio (SNR) for the proposed ACPE with D-T, P-T, and previous works, with various column information w are shown in Tables IV–VII. The $\bar{\varepsilon}$, $\varepsilon_{\max}$, ε_{var}, and SNR are defined as follows:

$$\begin{aligned}\bar{\varepsilon} &= E[P-P_q]/2^L \\ \varepsilon_{\max} &= \max\{|P-P_q|\}/2^L \\ \varepsilon_{var} &= \left(E\left[(P-P_q)^2\right] - E[P-P_q]^2\right)/2^{2L} \\ \mathrm{SNR(dB)} &= 10\log\frac{E[P^2]}{E\left[(P-P_q)^2\right]}\end{aligned} \tag{13}$$

where $|\,|$ and $\max\{\}$ are the absolute and maximum operators, respectively.

The SNR value is the most important error metric for the fixed-width Booth multiplier applying to a system. For $w=1$ case, the ACPE almost achieves the accuracy performance of [22], which is the highest accuracy in previous works as $w=1$. However, as for $w=2$ and $w=3$ cases, the ACPE Booth multipliers achieve the highest accuracy performance. Especially in $w=3$, the SNR values of the proposed ACPE Booth multiplier almost achieve the P-T one (less than 0.39 dB as $L=16$)

TABLE VIII
COMPARISONS OF AREA, DELAY, AND POWER WITH VARIOUS METHODS

ω	Methods	$L = 8$			$L = 10$			$L = 12$			$L = 14$			$L = 16$		
		Area (μm^2)	Delay (ns)	Power (mW)	Area (μm^2)	Delay (ns)	Power (mW)	Area (μm^2)	Delay (ns)	Power (mW)	Area (μm^2)	Delay (ns)	Power (mW)	Area (μm^2)	Delay (ns)	Power (mW)
L	P-T	9898	3.83	1.49	14750	4.67	2.84	19599	5.35	4.04	25599	6.21	5.88	32041	7.26	8.26
0	[15]	6318	3.36	0.77	8742	3.87	1.32	11664	4.25	1.95	14986	4.75	2.92	18626	5.17	3.96
	[16]	6318	3.36	0.77	8835	3.87	1.31	11772	4.25	1.97	15104	4.75	2.95	18760	5.17	3.94
	[19]	7470	3.95	1.18	11021	4.93	2.32	14986	5.40	2.92	18760	5.76	3.95	23244	6.31	5.24
	[20]	5986	3.03	0.69	8649	3.65	1.22	11227	4.07	1.79	14514	4.71	2.74	18224	5.12	3.75
1	[16]	7138	3.47	0.93	9800	4.05	1.52	12995	4.55	2.26	16641	5.24	3.33	20433	5.67	4.51
	[21]	6552	3.26	0.80	9300	3.75	1.37	12096	4.11	1.91	15624	4.72	2.87	19182	5.14	3.88
	[22]	6552	3.14	0.79	9300	3.59	1.30	11988	4.55	1.86	15624	5.01	2.72	19296	5.21	3.73
	ACPE	**6552**	**3.09**	**0.92**	**9300**	**3.75**	**1.35**	**12204**	**4.51**	**1.99**	**15748**	**4.90**	**2.98**	**19832**	**5.23**	**3.90**
2	[16]	8008	3.66	1.08	10815	4.19	1.78	14160	4.78	2.55	18146	5.37	4.40	22061	5.79	5.73
	[21]	7621	3.69	1.08	10764	3.78	2.09	13981	4.39	2.83	17716	4.89	4.23	21601	5.40	5.59
	ACPE	**7055**	**3.34**	**1.01**	**10094**	**3.98**	**1.51**	**12995**	**4.44**	**2.18**	**16770**	**5.02**	**3.24**	**20572**	**5.44**	**4.28**
3	[16]	8607	3.68	1.22	11718	4.23	1.98	15225	4.83	2.85	19471	5.44	4.01	23242	6.02	5.29
	[21]	8432	3.80	1.19	11446	4.26	1.88	15075	4.62	2.68	18763	5.48	3.92	22555	5.74	5.09
	ACPE	**8184**	**3.68**	**1.14**	**11227**	**4.38**	**1.91**	**14750**	**4.78**	**2.63**	**18760**	**5.29**	**3.92**	**22797**	**5.80**	**5.15**
×	D-T	5032	2.83	0.56	7553	3.61	1.01	9996	3.84	1.46	12995	4.50	2.28	16492	4.91	3.11

with 28.9% smaller area in $L = 16$. As the result indicates, the ACPE Booth multipliers provide a superior performance in accuracy compared with the existing works. Another important issue in fixed-width Booth multiplier design is circuit performance that includes area, delay, and power. Table VIII illustrates a comparison of area, delay, and power between the proposed ACPE and previous works. All the multipliers are implemented by using the architecture in Fig. 3 with their own compensation circuit expect for the BSD Booth multipliers in [19]. The values of area and delay in Table VIII are synthesized by Synopsys Design Compiler using a TSMC 0.18 μm standard cell library and run placement and routing by using the Cadence SoC Encounter. The power consumption is estimated by Synopsys PrimeTime PX using 100, 000 patterns for each case to verify those fixed-width Booth multipliers listed in Table VIII. Due to the highest accuracy performance, the P-T Booth multipliers have the largest area, delay, and power compared with other fixed-width Booth multipliers in Table VIII. On the other hand, D-T Booth multipliers achieve the lowest area, delay, and power performance. However, the largest error will be induced in D-T Booth multipliers. In [16], [21], [22], and ACPE, the Booth multipliers use more area than in other works ([15], [20], and [16] with $w = 0$) to achieve higher accuracy as shown in Tables IV–VII. However, the ACPE with $w = 1$ costs the same area as [21], [22] and lower area than [16]. As a result, the ACPE Booth multipliers employ a little area overhead to achieve higher accuracy.

B. Compensation Method Comparisons

Table IX shows a comparison of compensation methods with the existing works. All the works in Table IX use exhaustive simulations to produce the compensation fit functions except [19], [20], and the proposed ACPE. The fit functions are established

TABLE IX
COMPARISON OF ESTABLISHED COMPENSATION CIRCUITS

Methods	Fit Function	Supported varying w
Jou *et al.* [15]	Exhaustive Simulation	×
Song *et al.* [16]	Exhaustive Simulation	◯
Cho *et al.* [21]	Exhaustive Simulation	◯
Wang *et al.* [22]	Exhaustive Simulation	×
PEB [20]	Probability	×
Juang *et al.* [19]	Conditional-probability	×
ACPE	Conditional-probability	◯

◯ : Supported.
× : No supported.

from exhaustive simulations that consume long time especially for larger L, because the simulation time increases exponentially. Instead of exhaustive simulation methods, the probability methods can fast and mathematically generate the compensation functions. In this way, the ACPE establishes compensation circuit by applying the conditional-probability method and can achieve higher accuracy performance than the work in [20] that only uses the probability method. The conditional-probability method introduced in [19], however, does not achieve higher accuracy than existing works because the conditional bits are not chosen well.

In order to adjust the accuracy, the column information w should be induced. Song *et al.* use two fit functions to accommodate the exhaustive simulation results in $w > 0$ case [16], while other works, [15], [19], [20], [22], do not pose any fit function for varying w cases. For this reason, these works ([15], [19], [20], and [22]) can not support the varying w to adjust the accuracy. Besides, Cho *et al.* present the fit function to support varying w in the special format fixed-width multipliers

TABLE X
COMPARISON OF PSNR, AREA, AND POWER WITH EXISTING FIXED-WIDTH BOOTH MULTIPLIERS APPLIED IN DCT CORE

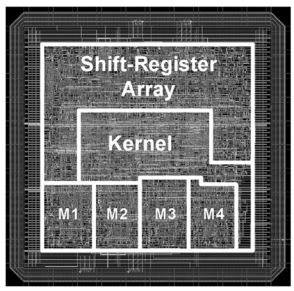
		P-T	D-T	ACPE			[15]	[16]				[19]	[20]	[21]			[22]
				$\omega = 1$	$\omega = 2$	$\omega = 3$		$\omega = 0$	$\omega = 1$	$\omega = 2$	$\omega = 3$			$\omega = 1$	$\omega = 2$	$\omega = 3$	
PSNR (dB)	Lena	56.10	34.62	52.55	54.84	55.58	51.36	51.36	52.15	54.69	55.98	47.15	52.04	50.47	51.41	53.74	52.55
	Baboon	55.95	34.61	52.50	54.60	55.56	50.94	50.94	51.53	54.33	55.86	46.96	52.09	50.73	50.98	53.62	52.53
	Peppers	56.08	34.62	52.49	54.76	55.58	51.18	51.18	51.99	54.64	55.99	47.09	52.01	50.51	51.32	53.77	52.50
	Elain	56.07	34.61	52.52	54.75	55.64	51.20	51.20	51.73	54.56	55.96	46.99	52.08	50.58	51.20	53.73	52.53
	Barb	56.11	34.62	52.62	54.83	55.56	51.31	51.31	52.13	54.65	56.01	47.15	52.12	50.51	51.34	53.70	52.59
	Goldhill	56.06	34.58	52.49	54.73	55.61	51.19	51.19	51.83	54.54	55.97	47.05	52.03	50.54	51.23	53.75	52.50
	Zelda	56.14	34.62	52.62	54.88	55.55	51.42	51.42	52.26	54.74	56.04	47.15	52.07	50.43	51.44	53.72	52.60
	Aerial	56.03	34.61	52.53	54.73	55.58	51.10	51.10	51.83	54.48	55.92	47.17	52.11	50.62	51.21	53.76	52.53
	Frog	55.95	34.60	52.59	54.61	55.59	51.04	51.04	51.52	54.35	55.86	47.10	52.19	50.80	50.99	53.59	52.59
	Boat	56.12	34.61	52.58	54.86	55.58	51.49	51.49	52.16	54.66	56.00	47.10	52.08	50.48	51.40	53.78	52.56
	Averaged	**56.06**	**34.61**	**52.55**	**54.76**	**55.58**	**51.22**	**51.22**	**51.91**	**54.56**	**55.96**	**47.09**	**52.08**	**50.57**	**51.25**	**53.72**	**52.55**
		100%	**61.7%**	**93.7%**	**97.7%**	**99.1%**	**91.4%**	**91.4%**	**92.6%**	**157.7%**	**99.8%**	**84.0%**	**92.9%**	**90.2%**	**91.4%**	**95.8%**	**93.7%**
Area (μm^2)		210525	158194	167518	171443	180464	164810	165741	172221	178498	183035	192665	162102	166416	175115	180451	166340
		100%	75.1%	79.6%	81.4%	85.7%	78%	78.7%	81.8%	84.8%	86.9%	91.5%	77.0%	79.0%	83.2%	85.7%	79.0%
Power (mW)		20.26	13.20	15.72	15.86	17.57	14.54	14.94	15.74	16.38	17.20	15.88	14.11	15.12	17.20	17.30	14.91

[21]. Fortunately, a single function (12) with varying w is derived based on a conditional-probability theory to support the adjusting accuracy. Therefore, the proposed ACPE provides a high accuracy, mathematical, and flexible compensation function for fixed-width Booth multiplier design.

C. Application to 2-D DCT Computations

The proposed ACPE Booth multipliers are then applied to a two-dimensional (2-D) DCT [2] for demonstrating the performance. The 2-D DCT core is implemented by a Shift-Register-Array and a one-dimensional (1-D) DCT kernel, which contains four 14-bit Booth multipliers. Therefore, the 10 test images, which are all 512×512 pixels with 8-bit 256 gray level data, are fed into the 2-D DCT to verify the accuracy performance. The system accuracy, defined as peak-signal-to-noise-ratio (PSNR), and area cost are the important data for evaluating the performance of 2-D DCT. By changing the four 14-bit Booth multipliers to P-T, D-T, ACPE ($w = 1, 2, 3$), and other works, the comparison table is established in Table X.

In Table X, the Synopsys Design Compiler and PrimeTime PX are applied with Artisan TSMC 0.18 μm standard cell library to implement all listed 2-D DCT circuits and estimate their power at 55 MHz clock frequency [2], respectively. Due to no pipeline stage and multiplexer structure for the DCT core [2], the DCT core only has a throughput rate of 27.5 mega pixels/sec when operated at 55 MHz. However, the 27.5 M-pel/s throughput rate of the DCT core can support the standard-definition television (SDTV) 480p 4:2:2 video specification, which needs $720 \times 480 \times 30 \times 2 = 20.7\ \mathrm{M} - \mathrm{pels/s}$ throughput rate, in a small area design. In DCT application, the D-T multiplier leads to the block effect and other fixed-width multipliers have almost the same image quality due to more than 40 dB. Compared with D-T Booth multiplier applications, the accuracy of ACPE with $w = 1, 2, 3$ improves more than 32% of the averaged PSNR performance. For the $w = 3$ case, the ACPE almost achieves the accuracy of P-T applications, the best PSNR in fixed-width multipliers, and further saves 14.3% area cost, compared with P-T applications. Therefore, the ACPE Booth multipliers have a superior performance in accuracy and area cost when applying to a 2-D DCT. Furthermore, when implementing the DCT core with the proposed ACPE Booth multiplier $(w = 3)$ on a chip, the Synopsys Design Compiler is used to synthesize the RTL design and the Cadence SoC Encounter is adopted for placement and routing (P&R). The core layout and simulated characteristics of the proposed 2-D DCT core are shown in Fig. 6. With the 1.8-V TSMC 0.18 μm 1P6M CMOS process, the proposed 2-D DCT core consumes 18 K gate counts and 17.6 mW of power, while operated at 55 MHz.



Characteristic	
Technology	0.18 μm 1P6M
Supply power	1.8V
Core size	509 μm x 518 μm
Gate Count	18 K
Max Freq.	55 MHz
Power	17.6 mW @55MHz

Fig. 6. Core layout and characteristics of the proposed ACPE with $w = 3$ applied in DCT core.

V. CONCLUSION

This paper proposes an adaptive conditional-probability estimator (ACPE) in fixed-width Booth multipliers. Without heuristic method, the ACPE is derived from conditional-probability theory, which can be easily applied to large length Booth multipliers for achieving higher accuracy performance. The column information w is also induced in ACPE to adjust the accuracy with respect to system requirements. For the 2-D DCT implementation with the ACPE Booth multipliers, an excellent performance is exhibited in terms of PSNR and area cost. As a result, the proposed ACPE provides a flexible and high accuracy compensation circuit applied to fixed-width Booth multipliers.

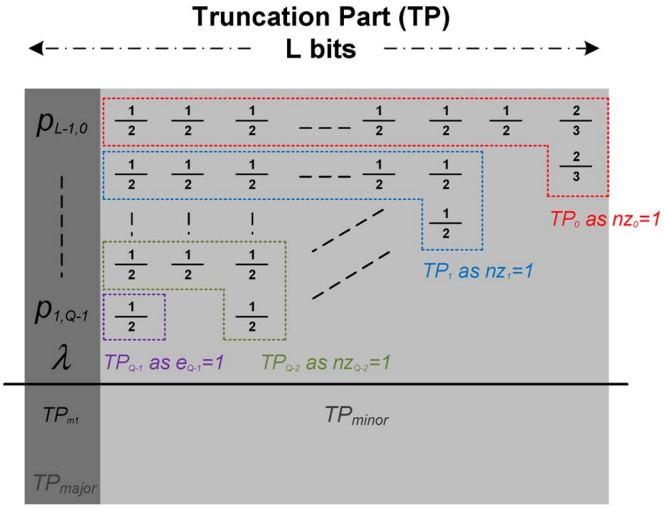


Fig. 7. The conditional expected values of TP.

APPENDIX

Based on the probability theory [30], [31], the conditional expected values of the partial products can be derived. Taking $E[p_{0,0}|nz_0 = 1]$ as an example, the conditional expected value can be obtained as follows:

$$\begin{aligned} E[p_{0,0}|nz_0=1] &= \sum_{k=-2,-1,1} P\left[y_0'=k|nz_0=1\right] P[p_{0,0}=1] \\ &= \frac{1}{3}\cdot 1 + \frac{1}{3}\cdot 0.5 + \frac{1}{3}\cdot 0.5 \\ &= \frac{2}{3}. \end{aligned} \tag{14}$$

Also, taking $E[p_{1,0}|nz_1 = 1]$ as another example.

$$\begin{aligned} E[p_{1,0}|nz_1=1] &= \sum_{k=-2,-1,1,2} P\left[y_1'=k|nz_1=1\right] P[p_{1,0}=1] \\ &= \frac{1}{6}\cdot 1 + \frac{2}{6}\cdot 0.5 + \frac{2}{6}\cdot 0.5 + \frac{1}{6}\cdot 0 \\ &= \frac{1}{2}. \end{aligned} \tag{15}$$

Using the same derivation, the conditional expected values based on nonzero code nz_i and e_{Q-1} are established as shown in Fig. 7. Therefore, the conditional expected value of TP_0 with $nz_0 = 1$ condition for $w = 1$ example is derived as follows:

$$\begin{aligned} &E[TP_0|nz_0 = 1] \\ &\quad = E\left[(p_{0,0}+n_0)2^{-L} + \cdots + p_{L-2}\cdot 2^{-2}|nz_0 = 1\right] \\ &\quad = \left(E[p_{0,0}|nz_0 = 1] + E[n_0|nz_0 = 1]\right)2^{-L} \\ &\qquad + E[p_{1,0}|nz_0 = 1]2^{-L+1} + \cdots \\ &\qquad + E[p_{L-2,0}|nz_0 = 1]2^{-2} \\ &\quad = \left(\frac{2}{3}+\frac{2}{3}\right)2^{-L} + \frac{1}{2}2^{-L+1} + \cdots + \frac{1}{2}2^{-2} \\ &\quad = \frac{1}{4} + \frac{1}{3}\left(\frac{1}{2}\right)^{L} \simeq \frac{1}{4} = \frac{nz_0}{2}\times\frac{1}{2}. \end{aligned} \tag{16}$$

Considering the varying w, the conditional expected value of TP_0 with $nz_0 = 1$ condition can be derived as follows:

$$\begin{aligned} &E[TP_0|nz_0 = 1] \\ &\quad = E\left[(p_{0,0}+n_0)2^{-L} + \cdots + p_{L-1-w}\cdot 2^{-1-w}|nz_0 = 1\right] \\ &\quad = \left(E[p_{0,0}|nz_0 = 1] + E[n_0|nz_0 = 1]\right)2^{-L} + \cdots \\ &\qquad + E[p_{L-1-w,0}|nz_0 = 1]2^{-1-w} \\ &\quad = \left(\frac{2}{3}+\frac{2}{3}\right)2^{-L} + \frac{1}{2}2^{-L+1} + \cdots + \frac{1}{2}2^{-1-w} \\ &\quad = \frac{1}{2}\left(\frac{1}{2}\right)^{w} + \frac{1}{3}\left(\frac{1}{2}\right)^{L} \simeq \frac{1}{2}\left(\frac{1}{2}\right)^{w} = \frac{nz_0}{2}\left(\frac{1}{2}\right)^{w}. \end{aligned} \tag{17}$$

Therefore, the TP_j with corresponding nonzero conditions $nz_j = 1$ are summarized as the following equations, where $Q-1-\lfloor w/2\rfloor \geq j \geq 0$:

$$\begin{aligned} E[TP_0|nz_0 = 1] &\simeq \frac{1}{2}\times\left(\frac{1}{2}\right)^{w} = \frac{nz_0}{2}\cdot\left(\frac{1}{2}\right)^{w} \\ E[TP_1|nz_1 = 1] &= \frac{1}{2}\times\left(\frac{1}{2}\right)^{w} = \frac{nz_1}{2}\cdot\left(\frac{1}{2}\right)^{w} \\ &\vdots \\ E[TP_{Q-1}|e_{Q-1} = 1] &= \frac{1}{2}\times\left(\frac{1}{2}\right)^{w} = \frac{e_{Q-1}}{2}\cdot\left(\frac{1}{2}\right)^{w}. \end{aligned} \tag{18}$$

Assume the nonzero conditional code z_j equals to nz_j as $Q-2 \geq j \geq 0$ and z_{Q-1} equals to e_{Q-1}, (18) can be rewritten as following equation:

$$\begin{aligned} E[TP_0|z_0 = 1] &\simeq \frac{1}{2}\times\left(\frac{1}{2}\right)^{w} = \frac{z_0}{2}\cdot\left(\frac{1}{2}\right)^{w} \\ E[TP_1|z_1 = 1] &= \frac{1}{2}\times\left(\frac{1}{2}\right)^{w} = \frac{z_1}{2}\cdot\left(\frac{1}{2}\right)^{w} \\ &\vdots \\ E[TP_{Q-1}|z_{Q-1} = 1] &= \frac{1}{2}\times\left(\frac{1}{2}\right)^{w} = \frac{z_{Q-1}}{2}\cdot\left(\frac{1}{2}\right)^{w}. \end{aligned} \tag{19}$$

ACKNOWLEDGMENT

The authors would like to thank the National Chip Implementation Center (CIC), Taiwan, for providing the circuit design automation tools.

REFERENCES

[1] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. New York: Wiley, 1999.
[2] S. C. Hsia and S. H. Wang, "Shift-register-based data transposition for cost-effective discrete cosine transform," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 6, pp. 725–728, Jun. 2007.
[3] H. Y. Lee and I. C. Park, "Balanced binary-tree decomposition for area-efficient pipelined FFT processing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 4, pp. 889–900, Apr. 2007.
[4] P. Bougas, P. Kalivas, A. Tsirikos, and K. Z. Pekmestzi, "Pipelined array-based FIR filter folding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 1, pp. 108–118, Jan. 2005.

[5] C. Y. Li, Y. H. Chen, T. Y. Chang, L. Y. Deng, and K. W. To, "Period extension and randomness enhancement using high-throughput reseeding-mixing PRNG," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2011, to be published.
[6] J. M. Jou, S. R. Kuang, and R. D. Chen, "Design of low-error fixed-width multipliers for DSP applications," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 6, pp. 836–842, Jun. 1999.
[7] 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, Analog Digit. Signal Process.*, vol. 47, no. 10, pp. 1112–1118, Oct. 2000.
[8] A. G. M. Strollo, N. Petra, and D. DeCaro, "Dual-tree error compensation for high performance fixed-width multipliers," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 8, pp. 501–507, Aug. 2005.
[9] L. D. Van and C. C. Yang, "Generalized low-error area-efficient fixed-width multipliers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 8, pp. 1608–1619, Aug. 2005.
[10] S. R. Kuang and J. P. Wang, "Low-error configurable truncated multipliers for multiply-accumulate applications," *Electron. Lett.*, vol. 42, no. 16, pp. 904–905, Mar. 2006.
[11] J. H. Tu and L. D. Van, "Power-efficient pipelined reconfigurable fixed-width Baugh-Wooley multipliers," *IEEE Trans. Comput.*, vol. 58, no. 10, pp. 1346–1355, Oct. 2009.
[12] N. Petra, D. D. Caro, V. Garofalo, E. Napoli, and A. G. M. Strollo, "Truncated binary multipliers with variable correction and minimum mean square error," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 6, pp. 1312–1325, Jun. 2010.
[13] C. H. Chang and R. K. Satzoda, "A low error and high performance multiplexer-based truncated multiplier," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 12, pp. 1767–1771, Dec. 2010.
[14] N. Petra, D. D. Caro, V. Garofalo, E. Napoli, and A. G. M. Strollo, "Design of fixed-width multipliers with linear compensation function," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 5, pp. 947–960, May 2011.
[15] S. J. Jou, M. H. Tsai, and Y. L. Tsao, "Low-error reduced-width Booth multipliers for DSP applications," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 50, no. 11, pp. 1470–1474, Nov. 2003.
[16] M. A. Song, L. D. Van, and S. Y. Kuo, "Adaptive low-error fixed-width Booth multipliers," *IEICE Trans. Fundam.*, vol. E90-A, no. 6, pp. 1180–1187, Jun. 2007.
[17] H. A. Huang, Y. C. Liao, and H. C. Chang, "A self-compensation fixed-width Booth multiplier and its 128-point FFT applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2006, pp. 3538–3541.
[18] Y. H. Chen, T. Y. Chang, and R. Y. Jou, "A statistical error-compensated Booth multiplier and its DCT applications," in *Proc. IEEE Region 10 Conf.*, 2010, pp. 1146–1149.
[19] T. B. Juang and S. F. Hsiao, "Low-error carry-free fixed-width multipliers with low-cost compensation circuits," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 6, pp. 299–303, Jun. 2005.
[20] C. Y. Li, Y. H. Chen, T. Y. Chang, and J. N. Chen, "A probabilistic estimation bias circuit for fixed-width Booth multiplier and its DCT applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 4, pp. 215–219, Apr. 2011.
[21] 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.
[22] J. P. Wang, S. R. Kuang, and S. C. Liang, "High-accuracy fixed-width modified Booth multipliers for lossy applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 1, pp. 52–60, Jan. 2011.
[23] Y. He and C. H. Chang, "A new redundant binary Booth encoding for fast 2^n-bit multiplier design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 6, pp. 1192–1201, Jun. 2009.
[24] C. H. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 10, pp. 1985–1997, Oct. 2004.
[25] D. Villeger and V. G. Oklobdzija, "Evaluation of Booth encoding techniques for parallel multiplier implementation," *Electron. Lett.*, vol. 29, no. 23, pp. 2016–2017, Nov. 1993.
[26] V. G. Oklobdzija, D. Villeger, and S. S. Liu, "A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach," *IEEE Trans. Comput.*, vol. 45, no. 3, pp. 294–306, Mar. 1996.
[27] G. Goto, A. Inoue, R. Ohe, S. Kashiwakura, S. Mitarai, T. Tsuru, and T. Izawa, "A 4.1-ns compact 54 × 54-b multiplier utilizing sign-select Booth encoders," *IEEE J. Solid-State Circuits*, vol. 32, no. 11, pp. 1676–1682, Nov. 1997.
[28] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*. Oxford, U.K.: Oxford Univ. Press, 2000.
[29] S. R. Kuang, J. P. Wang, and C. Y. Guo, "Modified Booth multipliers with a regular partial product array," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 5, pp. 404–408, May 2009.
[30] H. Stark and J. W. Woods, *Probability and Random Processes With Applications to Signal Processing*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.
[31] F. B. Hanson, *Applied Stochastic Processes and Control for Jump-Diffusions: Modeling, Analysis and Computation*. Philadelphia, PA: SIAM Books, 2007.

Yuan-Ho Chen (S'10) received the B.S. degree in from Chang Gung University, Taoyuan, Taiwan, in 2002, and the M.S. and Ph.D. degrees in electrical engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2004, and 2011, respectively.

He is now a postdoctoral research fellow at IC Design Laboratory in the Department of Engineering and System Science, NTHU. His research interests include video/image and digital signal processing, VLSI architecture design, VLSI implementation, computer arithmetic, fault-tolerant design, and signal processing for wireless communication.

Tsin-Yuan Chang (S'87–M'90) received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1982, and the M.S. and Ph.D. degrees from Michigan State University, East Lansing, in 1987 and 1989, respectively, both in electrical engineering.

He is an Associate Professor with the Department of Electrical Engineering, National Tsing Hua University. His research interests include IC design and testing, computer arithmetic, and VLSI DSP.