This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2017.2709801, IEEE Transactions on Circuits and Systems II: Express Briefs

A Low Error Energy-Efficient Fixed-Width Booth Multiplier with Sign-Digit-Based Conditional Probability Estimation

Ziji Zhang, and Yajuan He, *Member, IEEE*

***Abstract*—Fixed-width multipliers are intensively used in many DSP applications, whose accuracy and energy efficiency affect the whole digital system to a large extent. To improve the computation accuracy, a Booth encoded sign-digit-based conditional probability estimation approach is proposed. A symmetric error distribution is obtained by taking the sign bit of the Booth-encoded multiplier into consideration when applying the conditional probability. In addition, a more generalized mux-based estimation method is formulated for the circuit implementation which reduces the delay time, as well as power dissipation. Simulation results show that the proposed multiplier exhibits the best computation accuracy with the least energy per operation. It performs even better for those operand lengths that are not multiples of 4. The maximum reduction on energy-delay-error product can reach 14.8% compared with all its contenders among various operand lengths.**

***Index Terms*—fixed-width multiplier, Booth encoding, approximate computation, mean square error, energy-efficient.**

I. Introduction

FIXED-WIDTH multipliers are widely used in modern energy-efficient multimedia and digital signal processing systems, which are desirable to utilize a fixed operand length [1]-[5]. In order to obtain the same bit width as input, the least significant bits of the output are normally truncated. Among various fixed-width multipliers, post-truncated multiplier (PTM) rounds off the product after a complete calculation, which makes its accuracy the best while the hardware expenses the most. Contrarily, direct-truncated multiplier (DTM) cuts off the least significant half partial products directly without any compensation, which results in the least computation accuracy and the hardware complexity as well. For a low error energy-efficient multiplication, the truncation error has to be well compensated with minimum overhead in order to maintain the required precision while keep the energy dissipation as low as possible.

Recently, modified Booth algorithm has been increasingly employed in the fixed-width multiplier design so as to contribute further to the speedup and logic reduction [6]-[11]. A partitioning method was proposed in [6] to partition the truncated part in Booth multiplication into two: one major and one minor depending upon the influence on the induced truncation error. This method is widely adopted in the subsequent designs. Later on, a simple compensation circuit is proposed in [7] to compensate the truncation error with limited accuracy. To lower the error, a simulation-based method in [8] is formulated by taking the information provided by the Booth encoding. However, the exhaustive simulation is time consuming. A probabilistic estimation bias (PEB) method is thus proposed in [9] to reduce the simulation time with acceptable accuracy. The error performance is further improved in [10] by applying a complex multi-level conditional probability model, which gains higher accuracy at the cost of increased area. A better accuracy-area trade-off solution is presented in [11], which applies both the conditional probability estimation and the computer simulation with a dynamic error-compensation method.

In this paper, a Booth-encoded sign-digit-based conditional probability (BSCP) method is proposed in order to achieve a lower mean square error since it is an important indicator of computation accuracy [1]. The compensation value is derived by applying the sign bit of Booth encoded multiplier to the conditional and expected probability. A simple mux-based estimation circuit is formulated from the proposed method, which yields a simplified compensation logic design.

The rest of the paper is organized as follows. Section II introduces the basics of fixed-width Booth multiplier. The proposed compensation algorithm and the corresponding circuit design are illustrated in Section III. Section IV evaluates the performance of proposed fixed-width multiplier and compares it with its rivals. Finally, Section V concludes this paper.

II. Preliminaries

Suppose $A(a_{N-1}a_{N-2}\ldots\ldots a_1a_0)$ and $B(b_{N-1}b_{N-2}\ldots\ldots b_1b_0)$ are two N-bit two's complement numbers, which represent multiplicand and multiplier, respectively. Thus, their product P $(p_{2N-1}p_{2N-2}\ldots\ldots p_1p_0)$ can be obtained as follows:

$$P = A\times B = \left(-a_{N-1}\cdot 2^{N-1}+\sum_{i=0}^{N-2}a_i\cdot 2^i\right)\left(-b_{N-1}\cdot 2^{N-1}+\sum_{j=0}^{N-2}b_j\cdot 2^j\right). \quad (1)$$

In fast multiplication design, modified Booth encoding is an efficient way to reduce the number of partial products by half. As shown in Table I, three consecutive bits ($b_{2j+1}b_{2j}b_{2j-1}$) from a multiplier are grouped to form a signed digit (d_j). c_j represents a correction bit due to Booth encoding in two's complement representations. It is '1' if d_j is negative, otherwise it is '0'. The encoded multiplier D from its binary variant B is shown in (2).

Manuscript received Mar. 26, 2017; accepted Apr. 30, 2017. This work is supported by the National Natural Science Foundation of China (61604036, 61534002).

The authors are with the School of Microelectronics and Solid State Electronics, University of Electronic Science and Technology of China, No.4, Sec. 2, North Jianshe Road, 610054, Chengdu, China. (e-mail: yjhe@uestc.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2017.2709801, IEEE Transactions on Circuits and Systems II: Express Briefs

$$D=\sum_{j=0}^{N/2-1} d_j \cdot 2^{2j} \tag{2}$$

Therefore, P in (1) can be rewritten as

$$P=A\times D=\left(-a_{N-1}\cdot 2^{N-1}+\sum_{i=0}^{N-2} a_i\cdot 2^i\right)\left(\sum_{j=0}^{N/2-1} d_j\cdot 2^{2j}\right). \tag{3}$$

TABLE I
MODIFIED BOOTH ENCODING

b_{2i+1}	b_{2i}	b_{2i-1}	d_i	c_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	2	0
1	0	0	-2	1
1	0	1	-1	1
1	1	0	-1	1
1	1	1	0	0

Fig. 1 illustrates a typical partial product matrix of an 8×8-bit Booth multiplier. $pp_{j,i}$ indicates the partial product located at the i-th column and j-th row, which is generated from the i-th bit of the multiplicand a_i and the j-th digit of the Booth encoded multiplier d_j. A simplified sign extension method introduced in [12] is utilized, in which s_j denotes the extended sign bit of the j-th partial product.

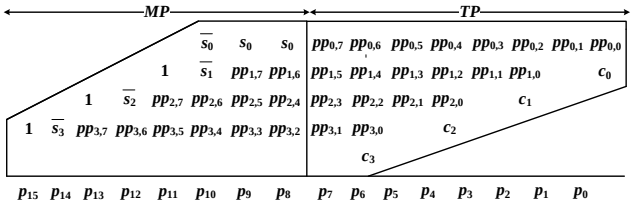

Fig. 1. Partial product matrix of an 8×8-bit Booth multiplier

For a fixed-width multiplier, the partial product matrix shown in Fig. 1 is typically partitioned into two parts: a main part (*MP*) for an accurate accumulation, and a truncation part (*TP*) for an approximate estimation. Thus, the product of the fixed-width multiplier, P_{FWM}, is given by

$$P_{FWM}=MP+TP\cdot 2^{-N}. \tag{4}$$

For an N-bit DTM, TP is directly truncated, which results in an error ε_{FWM} as indicated in (5). The maximum truncation error is thus derived as shown in (6).

$$\varepsilon_{FWM}=TP\cdot 2^{-N}=\sum_{j=0}^{N/2-1}\left(c_j\cdot 2^{2j-N}+\sum_{i=0}^{N-1-2j} pp_{j,i}\cdot 2^{i+2j-N}\right) \tag{5}$$

$$\varepsilon_{DTM,\max}=\sum_{j=0}^{N/2-1}\left(2^{2j-N}+\sum_{i=0}^{N-1-2j} 2^{i+2j-N}\right)=N/2 \tag{6}$$

On the other hand, for an N-bit PTM, both MP and TP are calculated and the result is then rounded to N bits, which gives the maximum error $\varepsilon_{PTM,\max}$ of 1/2.

III. PROPOSED FIXED-WIDTH MULTIPLIER DESIGN

Fig. 2 illustrates the composition of the proposed compensation algorithm. As indicated, TP_{major} and TP_{minor} are the sum of the most significant column of TP and the rest part of TP, respectively. The former is reserved for an accurate accumulation, while the latter is used for estimation. Suppose $f(D)$ is the compensation function, which is only related to the Booth-encoded multiplier D. As shown in Fig. 2, $f(D)$ represents the estimation result from TP_{minor}. Thus, the N-bit product $P_{A,D}$ can be expressed as

$$P_{A,D}=MP+(TP_{major}+f(D))/2. \tag{7}$$

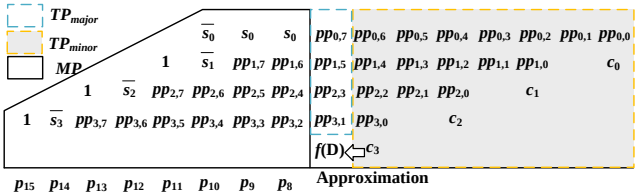

Fig. 2. Partial product matrix of fixed-width Booth multiplier with compensation algorithm

With (4) and (7), the truncation error after compensation $\varepsilon_{A,D}$ can be calculated as

$$\varepsilon_{A,D}=P_{FWM}-P_{A,D}=TP_{minor}\cdot 2^{-N}-f(D)/2. \tag{8}$$

A. Sign-Digit-Based Conditional Probability

As mean square error (MSE) is a key indicator of accuracy, it is important to keep the MSE as low as possible. Let ε_{mse} indicate the MSE of $\varepsilon_{A,D}$, and we use $TP_{A,D}$ replacing $TP_{minor}\cdot 2^{-N}$ in (8) for convenience, so we have:

$$\begin{aligned}\varepsilon_{mse}&=\frac{1}{2^{2N}}\sum_{D=-2^{N-1}}^{2^{N-1}-1}\sum_{A=-2^{N-1}}^{2^{N-1}-1}\varepsilon_{A,D}^2\\&=\frac{1}{2^{2N}}\sum_{D=-2^{N-1}}^{2^{N-1}-1}\sum_{A=-2^{N-1}}^{2^{N-1}-1}(TP_{minor}\cdot 2^{-N}-f(D)/2)^2\\&=\frac{1}{2^{2N}}\sum_{D=-2^{N-1}}^{2^{N-1}-1}\sum_{A=-2^{N-1}}^{2^{N-1}-1}(TP_{A,D}-f(D)/2)^2.\end{aligned} \tag{9}$$

For a given D, the MSE of $\varepsilon_{A,D}$ can be expressed as $\varepsilon_{mse,D}$. The compensation function $f(D)$ can always be formulated according to a specific D, which leads to the following:

$$\varepsilon_{mse,D}=\frac{1}{2^N}\sum_{A=-2^{N-1}}^{2^{N-1}-1}\varepsilon_{A,D}^2=\frac{1}{2^N}\sum_{A=-2^{N-1}}^{2^{N-1}-1}(TP_{A,D}-f(D)/2)^2. \tag{10}$$

Thus, (9) can be rewritten as:

$$\varepsilon_{mse}=\frac{1}{2^N}\sum_{D=-2^{N-1}}^{2^{N-1}-1}\varepsilon_{mse,D}. \tag{11}$$

Let $\sigma^2(\varepsilon_{A,D})$ and $\mu(\varepsilon_{A,D})$ represent the variance and mean of $\varepsilon_{A,D}$, respectively, according to the definition of MSE, we have:

$$\begin{aligned}\varepsilon_{mse,D}&=\sigma^2(\varepsilon_{A,D})+\mu^2(\varepsilon_{A,D})\\&=\sigma^2(TP_{A,D}-f(D)/2)+\mu^2(TP_{A,D}-f(D)/2).\end{aligned} \tag{12}$$

Since $f(D)$ is only related to D, the variance of $\varepsilon_{A,D}$ should be a constant and independent of $f(D)$. Therefore, the Min value of $\varepsilon_{mse,D}$ can be obtained as shown in (13), with the condition being satisfied in (14).

$$\varepsilon_{mse,D_{\min}}=\sigma^2(TP_{A,D}-f(D)/2) \tag{13}$$

$$\mu(TP_{A,D}-f(D)/2)=0 \tag{14}$$

With (11) and (13), the Min value of ε_{mse} can be derived as

$$\varepsilon_{mse\,\min}=\sum_{D=-2^{N-1}}^{2^{N-1}-1}\varepsilon_{mse,D_{\min}}=\sum_{D=-2^{N-1}}^{2^{N-1}-1}\sigma^2(TP_{A,D}-f(D)/2) \tag{15}$$

As the mean of $TP_{A,D}$ could be obtained from the expectations of each partial product together with the corresponding correction bit. Thus, the mean value of the j-th partial product in the matrix, $\mu_j(TP_{A,D})$, can be calculated as

$$\mu_j(TP_{A,D})=\sum_{i=0}^{N-2j-2}E(pp_{j,i})\cdot 2^{2j+i-N}+c_j\cdot 2^{2j-N} \tag{16}$$

where $E(\cdot)$ indicates the expectation. Noted that c_j here uses the actual bit generated from the encoded multiplier instead

of a traditional estimation.

Let n_j be an indication whether the j-th Booth encoded digit (d_j) is a non-zero digit or not. Based on all possible values of d_j, n_j, c_j, each partial product bit $pp_{j,i}$ and its corresponding expectation are summarized in Table II. a_i is the i-th bit of the multiplicand (a_{-1}=0) with a uniform input distribution. Thus, the probability of a_i being one is a half.

TABLE II
PARTIAL PRODUCT BITS AND THE CORRESPONDING EXPECTATIONS BASED ON ALL POSSIBLE BOOTH-ENCODED DIGITS

d_j	c_j	n_j	$pp_{j,i}$		$E(pp_{j,i})$	
			$i\neq0$	i=0	$i\neq0$	i=0
0	0	0	0	0	0	0
1	0	1	a_i	a_0	P(a_i=1)=1/2	P(a_0=1)=1/2
2	0	1	a_{i-1}	a_{-1}	P(a_{i-1}=1)=1/2	P(a_{-1}=1)=0
-1	1	1	$\overline{a_i}$	$\overline{a_0}$	P(a_i=0)= 1/2	P(a_0=0)= 1/2
-2	1	1	$\overline{a_{i-1}}$	$\overline{a_{-1}}$	P(a_{i-1}=0)=1/2	P(a_{-1}=0)=1

Observed from Table II, $E(pp_{j,i})$ can be expressed as a function of d_j and c_j according to n_j and the integer i as follows:

$$E(pp_{j,i})=\begin{cases}0 & n_j=0\\ 1/2 & n_j\neq0,\ i\neq0\\ 1-d_j/2-c_j & n_j\neq0,\ i=0\end{cases}. \tag{17}$$

From (17), $E(pp_{j,i})$ can be further derived as a function of d_j, c_j and n_j

$$E(pp_{j,i})=\begin{cases}n_j/2 & i\neq0\\ n_j-d_j/2-c_j & i=0\end{cases}. \tag{18}$$

Therefore, $\mu_j(TP_{A,D})$ in (16) can be calculated as

$$\begin{aligned}\mu_j(TP_{A,D}) &= \sum_{i=0}^{N-2j-2}E(pp_{j,i})\cdot2^{2j+i-N}+c_j\cdot2^{2j-N}\\ &= \sum_{i=1}^{N-2j-2}(n_j/2)\cdot2^{2j+i-N}+E(pp_{j,0})\cdot2^{2j-N}+c_j\cdot2^{2j-N}\\ &= n_j/4+(-d_j)\cdot2^{2j-1-N}\quad (j=0,1,\cdots,N/2-1).\end{aligned} \tag{19}$$

Let β be the total number of non-zero digits in D, we have:

$$\begin{aligned}\mu(TP_{A,D}) &= \sum_{j=0}^{N/2-1}\mu_j(TP_{A,D})=\sum_{j=0}^{N/2-1}n_j/4+(-d_j)\cdot2^{2j-1-N}\\ &= \beta/4+\sum_{j=0}^{N/2-1}(-d_j)\cdot2^{2j-1-N}\quad (0\le\beta\le N/2).\end{aligned} \tag{20}$$

With (14) and (20), $f(D)$ can be derived as:

$$f(D)=2\mu(TP_{A,D})=\beta/2+2\sum_{j=0}^{N/2-1}(-d_j)\cdot2^{2j-1-N}. \tag{21}$$

As the final product is supposed to be an integer, the compensation value should be rounded into an integer as well. In addition, the amplitude of $\sum_{j=0}^{N/2-1}(-d_j)\cdot2^{2j-1-N}$ is less than a quarter with its sign opposite to the multiplier, i.e., when b_{N-1} = 0, this term is negative, otherwise, it is positive. Considering the parity of β, the compensation integer $f(D)_{\text{int}}$ can be derived as:

$$f(D)_{\text{int}}=\begin{cases}\lfloor\beta/2\rfloor+1 & b_{N-1}=1\\ \lfloor(\beta-1)/2\rfloor+1 & b_{N-1}=0\end{cases}. \tag{22}$$

B. Compensation Circuits Design

Suppose $\omega_{k-1}\omega_{k-2}\ldots\ldots\omega_1\omega_0$ is a k-bit compensation binary that equals $f(D)_{\text{int}}$, k is determined by

$$k=1+\lfloor\log_2(\lfloor\max(\beta)/2\rfloor+1)\rfloor=1+\lfloor\log_2(\lfloor N/4\rfloor+1)\rfloor. \tag{23}$$

Table III illustrates the calculation on the compensation value based on β for the operand length N up to 16 bits according to (22).

TABLE III
COMPENSATION VALUES DERIVED BASED ON β

β	b_{N-1}	$f(D)_{\text{int}}$	ω_2	ω_1	ω_0
0	0	1	0	0	1
1	0 or 1	1	0	0	1
2	0	1	0	0	1
	1	2	0	1	0
3	0 or 1	2	0	1	0
4	0	2	0	1	0
	1	3	0	1	1
5	0 or 1	3	0	1	1
6	0	3	0	1	1
	1	4	1	0	0
7	0 or 1	4	1	0	0
8	0	4	1	0	0
	1	5	1	0	1

To obtain 'β' and 'β-1' at the same time, odd-even merge sorting algorithm is a more efficient way than traditional addition. In this algorithm, the non-zero bit array ($n_{N/2-1}n_{N/2-2}\ldots n_j\ldots n_1n_0$) is sorted in such an ascending order, $\Gamma(\gamma_{N/2-1}\gamma_{N/2-2}\ldots\gamma_j\ldots\gamma_1\gamma_0)$, so that there must be a demarcation point, γ_β, where γ_β=0 and $\gamma_{\beta-1}$=1. Therefore, we have:

$$\gamma_j=\begin{cases}1 & (0\le j<\beta)\\ 0 & (\beta\le j<N/2)\end{cases}. \tag{24}$$

A simple odd-even merge sorting circuit is exemplified in Fig. 3 with N=8 and k=2. ($\gamma_3\gamma_2\gamma_1\gamma_0$) is the order generated by the merge sorting circuits from the original array ($n_3n_2n_1n_0$).

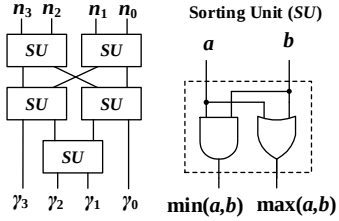

Fig. 3. A merge sorting circuit used for a fixed-width Booth multiplier

Once the order is generated, a 2-bit compensation value $f(D)_{\text{int}}$ ($\omega_1\omega_0$) can be derived according to Table III as

$$\begin{aligned}\omega_1&=\gamma_2\overline{b_7}+\gamma_1b_7=\gamma_2+\gamma_1b_7\\ \omega_0&=\overline{\gamma_2}\,\overline{b_7}+\gamma_3b_7+\overline{\gamma_1}b_7=\overline{\omega_1}+\gamma_3b_7.\end{aligned} \tag{25}$$

As noticed, the merge sorting circuit requires a balanced structure. Therefore, extra hardware resources would be incurred for those operand lengths which are not multiples of 4. In order to deal with this problem, we propose a simple method that firstly round down the operand length to be a multiple of 4, which excludes $n_{N/2-1}$ from the original array. Then, b_{N-1} and $n_{N/2-1}$ are used as two control signals to select the correct compensation value. As an example, when N=10, the two-bit binary $f(D)_{\text{int}}$ can be derived in this way as

$$\begin{aligned}\omega_1&=(\gamma_2\overline{n_4}+\gamma_1n_4)\overline{b_9}+(\gamma_1\overline{n_4}+\gamma_0n_4)b_9\\ \omega_0&=((\overline{\gamma_1}+\gamma_3)n_4+\overline{\gamma_2}\,\overline{n_4})\overline{b_9}+((\overline{\gamma_0}+\gamma_2)n_4+(\overline{\gamma_1}+\gamma_3)\overline{n_4})b_9.\end{aligned} \tag{26}$$

A more generalized derivation on the m-th bit compensation value, ω_m, is shown below. These two equations are differentiated by the operand length whether it is a multiple of 4 (27) or not (28).

$$\begin{aligned}\omega_m=b_{N-1}(\gamma_{2^{m+1}-3}\overline{\gamma_{2\cdot2^{m+1}-3}}+\gamma_{3\cdot2^{m+1}-3}\overline{\gamma_{4\cdot2^{m+1}-3}}+\ldots\ldots)+\\ \overline{b_{N-1}}(\gamma_{2^{m+1}-2}\overline{\gamma_{2\cdot2^{m+1}-2}}+\gamma_{3\cdot2^{m+1}-2}\overline{\gamma_{4\cdot2^{m+1}-2}}+\ldots\ldots)\end{aligned} \tag{27}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2017.2709801, IEEE Transactions on Circuits and Systems II: Express Briefs

$$\begin{aligned}\omega_m = {} & b_{N-1}(n_{N/2-1}(\gamma_{2^{m+1}-4}\overline{\gamma_{2\cdot 2^{m+1}-4}} + \gamma_{3\cdot 2^{m+1}-4}\overline{\gamma_{4\cdot 2^{m+1}-4}} + \ldots\ldots) + \\ & \overline{n_{N/2-1}}(\gamma_{2^{m+1}-3}\overline{\gamma_{2\cdot 2^{m+1}-3}} + \gamma_{3\cdot 2^{m+1}-3}\overline{\gamma_{4\cdot 2^{m+1}-3}} + \ldots\ldots)) + \\ & \overline{b_{N-1}}(n_{N/2-1}(\gamma_{2^{m+1}-3}\overline{\gamma_{2\cdot 2^{m+1}-3}} + \gamma_{3\cdot 2^{m+1}-3}\overline{\gamma_{4\cdot 2^{m+1}-3}} + \ldots\ldots) + \\ & \overline{n_{N/2-1}}(\gamma_{2^{m+1}-2}\overline{\gamma_{2\cdot 2^{m+1}-2}} + \gamma_{3\cdot 2^{m+1}-2}\overline{\gamma_{4\cdot 2^{m+1}-2}} + \ldots\ldots)).\end{aligned} \tag{28}$$

Based on these derivations, an estimation circuit used to generate the compensation value is proposed with one value generation block and one selection logic. The former is responsible to generate all possible compensation values, $\omega_{m,r}$. Fig. 4(a) shows its gate-level implementation, where r indicates the corresponding bit in Γ. Fig. 4(b) and 4(c) illustrates two cases for a slice of the estimation circuit with the operand length a multiple of 4 and not a multiple of 4, respectively. Once $\omega_{m,r}$ is obtained from each corresponding value generation block, a correct compensation value ω_m will be chosen by the selection logics with the control signal b_{N-1} for case (b) and b_{N-1} together with $n_{N/2-1}$ for case (c).

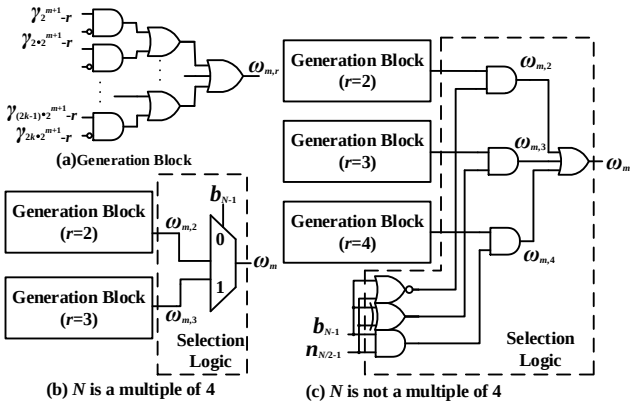

Fig. 4. Estimation circuit for producing one-bit compensation value ω_m

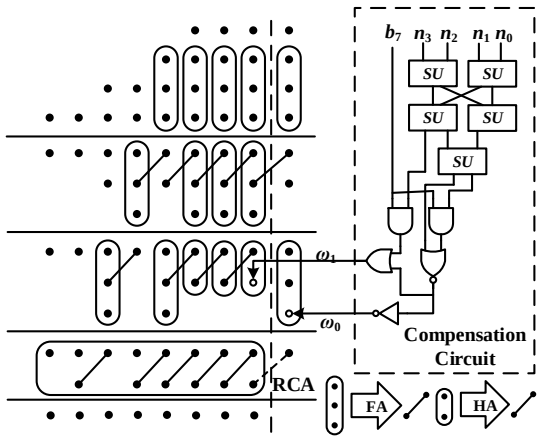

Fig. 5. Circuit structure of an 8×8-bit BSCP multiplier

Fig. 5 illustrates the circuit structure of the reduction procedure for an 8×8-bit BSCP multiplier. As shown in the figure, black dots indicate accurate bits corresponding to the partial products in *MP* and TP_{major} according to the partitioning method in Fig. 2. A gate-level circuit surrounded by the dashed line is used to generate the compensation value derived from $f(D)_{\text{int}}$. This compensation value is a two-bit number, $\omega_1\omega_0$, represented with white dots in Fig. 5. HA and FA represent a half adder and a full adder, respectively. A carry propagation adder is used to generate the final result.

IV. Performance Evaluations

This section evaluates the performance of our proposed fixed-width Booth multiplier and compares it with several competitive multipliers, including DTM, PTM, the probability and computer simulation (PACS) based multiplier [11], and the one with simple circuit generator (SCG) [8].

For a fair and legitimate comparison, all fixed-width Booth multipliers are implemented with the same Booth encoder and the compression tree structure.

A. Accuracy

The accuracy analysis of fixed-width Booth multiplier is performed for operand lengths of 8, 10, 12, 14 and 16 bits. The results are shown in Table IV, which lists the mean error (ME), maximum absolute error (MAE) and mean square error (MSE) for each multiplier.

As illustrated in Table IV, except PTM, the proposed BSCP multiplier exhibits the best accuracy with the minimum ME, MAE as well as MSE. Compared with the most competitive competitor SCG [8], although BSCP performs the same in terms of ME and MAE, it outperforms in MSE. This is due to the way which the compensation value is derived. With the actual sign information instead of a traditional estimation, the proposed method comes with a more symmetric error distribution which results in a better MSE.

TABLE IV
Error Performance of Different Fixed-Width Multipliers

Bit-width	Multiplier	ME	MAE	MSE
8	PTM	0	0.5000	0.0833
	DTM	1.50100	4.0000	2.6860
	PACS[11]	0.04492	1.1680	0.1370
	SCG[8]	0.00782	1.1680	0.1367
	BSCP (This work)	**0.00782**	**1.1680**	**0.1333**
10	PTM	0	0.5000	0.0833
	DTM	1.87520	5.0000	4.0563
	PACS[11]	0.06892	1.5000	0.1544
	SCG[8]	-0.00391	1.5000	0.1542
	BSCP (This work)	**-0.00391**	**1.5000**	**0.1498**
12	PTM	0	0.5000	0.0833
	DTM	2.25010	6.0000	5.7068
	PACS[11]	0.05847	1.6667	0.1672
	SCG[8]	0.00195	1.6667	0.1671
	BSCP (This work)	**0.00195**	**1.6667**	**0.1633**
14	PTM	0	0.5000	0.0833
	DTM	-2.62500	7.0000	7.6390
	PACS[11]	0.06442	2.0000	0.1822
	SCG[8]	-0.00098	2.0000	0.1821
	BSCP (This work)	**-0.00098**	**2.0000**	**0.1781**
16	PTM	0	0.5000	0.0833
	DTM	3.00000	8.0000	9.8525
	PACS[11]	0.06152	2.1667	0.1961
	SCG[8]	0.00049	2.1667	0.1961
	BSCP (This work)	**0.00049**	**2.1667**	**0.1922**

B. Circuit Performance

To evaluate the circuit performance, each multiplier mentioned in Table IV is described in Verilog HDL, and synthesized using the Synopsys Design Compiler. All experiments are carried out in a 32-nm CMOS technology. Table V summarizes the worst-case delay, power dissipation, normalized energy per operation and area of the fixed-width Booth multipliers. The average power dissipations are simulated by Synopsys Power Compiler with back annotated switching activity files generated from one million random input vectors at a 200MHz data rate with 50% switching activity to each design. Both area and energy per operation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2017.2709801, IEEE Transactions on Circuits and Systems II: Express Briefs

for each operand length are normalized so that DTM with the least area and energy consumption is one.

As indicated in Table V, the proposed BSCP multiplier has the minimum delay time except DTM. This is because the compensation function of this work uses a mux-based estimation circuits instead of modifying the partial products before compression as in [8] and [11]. This results in shorter critical paths with a slight overhead on area. As for power dissipation, the proposed BSCP multiplier dissipates the least for those bit widths which are not multiples of 4, although their corresponding area are not the smallest. That is due to our logic simplification on traditional merge sorting algorithm discussed in Section III.B, which decreases the chances of switching activity for the internal signals. When the operand length increases, the power savings becomes more obvious for all operand lengths. Benefit from the improved speed, and power dissipation, this work also exhibits the least energy consumption among all competitors.

TABLE V
CIRCUIT PERFORMANCE WITH DIFFERENT FIXED-WIDTH MULTIPLIER

Bit-width	Multiplier	Delay (ns)	Power (μW)	Nor. Energy	Nor. Area
8	PTM	1.29	170.4	2.625	2.016
	DTM	1.03	81.3	1	1
	PACS[11]	1.15	114.2	1.568	1.372
	SCG[8]	1.14	114.1	1.553	1.399
	BSCP (This work)	**1.11**	**116.5**	**1.544**	**1.404**
10	PTM	1.58	278.2	2.492	1.997
	DTM	1.32	133.6	1	1
	PACS[11]	1.39	178.9	1.410	1.307
	SCG[8]	1.40	180.6	1.434	1.337
	BSCP (This work)	**1.37**	**177.1**	**1.376**	**1.320**
12	PTM	1.79	401.1	2.726	2.021
	DTM	1.35	195.1	1	1
	PACS[11]	1.63	253.9	1.571	1.280
	SCG[8]	1.58	260.3	1.561	1.321
	BSCP (This work)	**1.52**	**254.8**	**1.470**	**1.296**
14	PTM	1.97	561.4	2.535	2.028
	DTM	1.58	276.1	1	1
	PACS[11]	1.71	349.0	1.368	1.251
	SCG[8]	1.74	355.1	1.416	1.289
	BSCP (This work)	**1.71**	**348.3**	**1.365**	**1.265**
16	PTM	2.00	829.2	2.469	2.041
	DTM	1.76	381.6	1	1
	PACS[11]	1.95	458.6	1.332	1.229
	SCG[8]	1.95	460.7	1.338	1.244
	BSCP (This work)	**1.95**	**452.8**	**1.315**	**1.240**

C. Energy-Delay-Accuracy Analysis

For benchmarking the energy efficiency of a circuit, the energy-delay product (*EDP*) is used to be a better metric than energy per operation. For lossy applications, to maintain a required accuracy, error performance should be taken into account as well as the energy efficiency. Thus, a new design metric energy-delay-error-product (EDE_{mse}) is adopted to better evaluate the design efficiency of the proposed approach, which is given by

$$EDE_{mse} = Energy(E) \times Delay(D) \times MeanSquareError(E_{mse}). \quad (29)$$

As there is no compensation algorithm involved in either DTM or PTM, these two multipliers are excluded from this comparison. The results are illustrated in Fig. 6.

As expected, the proposed design exhibits the least EDE_{mse} for all operand lengths. That means BSCP approach in the design of a fixed-width multiplier is the most energy efficient one with the highest accuracy. It is more efficient for those multipliers with the operand lengths that are not multiples of 4. It can reach maximum 14.8% reduction compared with all its contenders among various operand lengths.

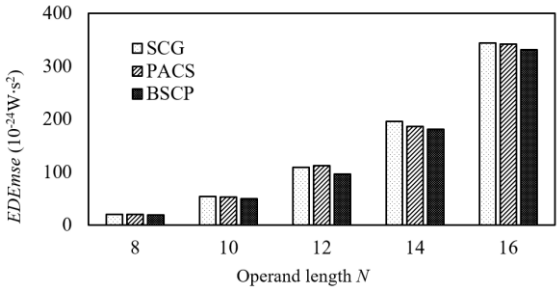

Fig. 6. EDE_{mse} with different fixed-width Booth multiplier

V. CONCLUSION

In this paper, a Booth encoded sign-digit-based conditional probability method for a fixed-width multiplier design is presented. A mux-based estimation circuit is proposed to simplify the compensation logics, and shorten the critical paths. The proposed approach can be easily adapted for an accuracy-adaptive design by putting variable columns of TP_{major} into considerations. Simulation results show that our proposed BSCP multiplier is the most energy efficient design with the highest accuracy. It can reach maximum 14.8% reduction on the energy-delay-error product compared with all its contenders among various operand lengths.

REFERENCES

[1] N. Petra, D. De 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.

[2] 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.

[3] S. N. Tang, J. W. Tsai, and T. Y. Chang, "A 2.4-Gs/s FFT processor for OFDM-based WPAN applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 6, pp. 451–455, Jun. 2010.

[4] S. W. Yen, S. Y. Hung, C. L. Chen, H. C. Chang, S. J. Jou, and C. Y. Lee, "A 5.79-Gb/s energy-efficient multirate LDPC codec chip for IEEE 802.15.3c applications," *IEEE J. Solid-State Circuits*, vol. 47, no. 9, pp. 2246–2257, Sep. 2012.

[5] H.-J. Ko and S.-F. Hsiao, "Design and application of faithfully rounded and truncated multipliers with combined deletion, reduction, truncation, and rounding," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 5, pp. 304-308, May 2011.

[6] 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.

[7] 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.*, May 2006, pp. 3538–3541.

[8] 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.

[9] 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.

[10] Y.-H. Chen, "An accuracy-adjustment fixed-width Booth multiplier based on multilevel conditional probability," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 1, pp. 203-207, Jan. 2015.

[11] W.-Q. He, Y.-H. Chen, and S.-J. Jou, "High-accuracy fixed-width Booth multipliers based on probability and simulation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 8, pp. 2052-2061, Aug. 2015.

[12] G. W. Bewick, "*Fast multiplication: algorithm and implementation*," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1994.