A Probabilistic Prediction-Based Fixed-Width Booth Multiplier for Approximate Computing

Yajuan He, *Member, IEEE*, Xilin Yi, *Graduate Student Member, IEEE*, Ziji Zhang, *Student Member, IEEE*, Bin Ma, and Qiang Li, *Senior Member, IEEE*

***Abstract*— In order to gain substantial circuit performance improvement in terms of power, speed and area cost, approximate computing is ideally applied in many error-resilient DSP and multimedia applications by relaxing the requirements of exact computation. Fixed-width multipliers are intensively used in those applications, which makes them a good demonstration for approximation through dynamic range reduction. In this paper, a probabilistic prediction based fixed-width Booth multiplier has been proposed. The error compensation function is derived from the probabilistic analysis on the partial products obtained from Booth encoding. By slightly modifying the partial product array with dedicated partitioning, a simple compensation circuit is thus formulated. Compared with the previous works, the proposed design exhibits the best performance-accuracy tradeoff. It achieves at least 27% reduction on the area-energy-delay-error product compared with the other multipliers among various operand lengths. Additionally, the evaluation on 2-Dimensional discrete cosine transform (DCT) circuits indicates that our proposed fixed-width Booth multiplier is applicable for lossy applications with a considerable reduction on hardware expenses and power dissipation while maintaining decent output quality.**

***Index Terms*— Probabilistic prediction, approximate computing, fixed-width multiplier, Booth encoding, discrete cosine transform (DCT).**

I. Introduction

APPROXIMATE computing is an emerging paradigm in the design of energy-efficient digital systems by exploiting the inherent error resilience of applications, such as multimedia processing, pattern recognition, machine learning and data mining [1]. By relaxing the requirement of traditional exact computation, approximate computing shows the potential for substantial energy efficiency and performance improvement. Datapath computation is demonstrated as a candidate of nature for approximation through dynamic range reduction [2], [3].

As multiplication is a fundamental operation and extensively used in many digital signal processing systems, the design efficiency on digital multipliers often impose a large impact on the entire system. Multiplication of two binary numbers with equal bit-width generates a product with twice of original bit-width. For most of the computational intensive applications, the bits of precision at the multiplier output are more than required [4]. It is usually desirable to utilize a fixed operand length that both the input and output of the datapath are with the same bit-width to facilitate the data storage in registers. Half of the less significant product bits are thus truncated in order to obtain the same bit-width as input, which leads to the design of fixed-width multipliers. Due to a wide range of applications such as filtering, convolution, and fast Fourier or discrete cosine transform, fixed-width multipliers have been extensively studied for approximate circuit implementations [5]–[19].

Among various fixed-width multipliers, a direct-truncated multiplier ignores half of the partial products array which contains the less significant bits. It reduces considerable area while introduces a large truncation error. On the contrary, post-truncated multiplier rounds off the product after a complete accumulation of all partial product bits, which makes its accuracy the best while the hardware expenses the most. To achieve a balanced design between accuracy and hardware expenses, various error-compensated circuits have been proposed to alleviate the truncation errors in Baugh–Wooley array multipliers [5]–[8] and Booth multipliers [12]–[19]. As only half of the partial products are generated with modified Booth encoding, Booth multipliers perform better in speed and area. Also due to reduced number of truncated partial products, fixed-width Booth multipliers present smaller truncation error than that of array multipliers [18]. Therefore, recent research works have focused on the compensated circuit in Booth multipliers [14]–[19].

A low cost compensation method was proposed in [12] by using statistical and linear regression analysis. The mean error is reduced hugely compared with direct-truncated Booth multiplier (DTBM). But other error metrics are still large due to the fact that the estimation information taken from the truncated part is limited. To lower the error, a partitioning method was proposed in [13] to divide the truncated partial products array into two: one major and one minor depending upon the impact on the induced truncation error. Based on this partitioning method, a simulation-based error estimation function was formulated in [14] by taking the information from Booth encoding. However, considerable time is consumed to establish the compensation function based on exhaustive simulations. A probabilistic estimation bias (PEB)

Manuscript received January 31, 2020; revised May 18, 2020; accepted May 31, 2020. Date of publication June 23, 2020; date of current version December 1, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61874023 and 61534002. This article was recommended by Associate Editor D. Zhao. *(Corresponding author: Yajuan He.)*

Yajuan He, Xilin Yi, Ziji Zhang, and Qiang Li are with the School of Electronic Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: yjhe@uestc.edu.cn; yi-xilin@outlook.com; 201611030130@std.uestc.edu.cn; qli@uestc.edu.cn).

Bin Ma is with the Department of Short Distance Wireless Communication, HiSilicon Technologies Company Ltd., Shanghai 200120, China (e-mail: mabin_uestc@163.com).

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

Digital Object Identifier 10.1109/TCSI.2020.3001654

1549-8328 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on December 19,2021 at 06:28:50 UTC from IEEE Xplore. Restrictions apply.

method is thus proposed in [15] to reduce the calculation time with acceptable accuracy. Later on, an adaptive conditional-probability estimator (ACPE) was proposed in [16] by using nonzero code to estimate and compensate truncation errors. The error performance is further improved in [17] by applying a complex multi-level conditional probability model (MLCP), which gains higher accuracy at the cost of increased area. A better accuracy-area tradeoff solution was presented in [18], which applies both the conditional probability estimation and computer simulation with a dynamic error-compensation method. By applying the sign bit of Booth encoded multiplier to the conditional and expected probability (BSCP), a more accurate compensation function was generated to achieve an energy-efficient multiplier design in [19].

In this paper, we present a simple yet accurate error compensation function derived from the probabilistic analysis on the partial products obtained from Booth encoding. By slightly modifying the partial product array with dedicated partitioning, a simple compensation circuit is formulated. Compared with the previous works, the proposed fixed-width Booth multiplier exhibits the best performance-accuracy tradeoff. The evaluation on 2-D DCT circuit further demonstrates the proposed design is beneficial to lossy applications with considerable reduction on hardware expenses and power dissipation while maintaining decent output quality.

The rest of the paper is organized as follows. Section II gives the introduction on the design of a fixed-width Booth multiplier utilizing the negative zero encoding. The proposed compensation function of probabilistic prediction method is presented in Section III. Section IV presents the hardware implementation of the proposed fixed-width Booth multiplier. Section V evaluates the performance of the proposed design and compares it with its rivals. The chip implementation and the evaluation on DCT application are also presented in Section V. Finally, the conclusions of this paper are drawn in Section VI.

II. Fixed-Width Booth Multiplier (FWBM)

Modified Booth encoding is an efficient way in the fast multiplication to halve the number of partial products, with which all bits in the multiplier are partitioned into triplets. Table I shows the modified Booth encoding utilizing a negative zero representation, which is slightly different from the traditional Booth algorithm. As indicated, d_k is a signed digit computed by each triplet and $d_k \in \{\pm 0, \pm 1, \pm 2\}$. The integer k denotes the k-th encoded digit in the multiplier and c_k is the correction bit due to modified Booth encoding in two's complement representations. Thus, the product of an $n \times n$-bit Booth multiplier with full accuracy, P_A, can be expressed as:

$$\begin{aligned} P_A &= X \times Y \\ &= (-x_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} x_i \cdot 2^i)(-y_{n-1} \cdot 2^{n-1} \\ &\quad + \sum_{i=0}^{n-2} y_i \cdot 2^j) \\ &= (-x_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} x_i \cdot 2^i) \cdot \sum_{k=0}^{\frac{n}{2}-1} d_k \cdot 2^{2k} \\ &\quad (k = 0, 1, \ldots, \frac{n}{2} - 1) \end{aligned} \tag{1}$$

TABLE I
MODIFIED BOOTH ENCODING

y_{2k+1}	y_{2k}	y_{2k-1}	d_k	c_k
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	1

TABLE II
8-BIT PARTIAL PRODUCTS BASED ON ALL POSSIBLE BOOTH ENCODED DIGITS

d_k	$s_k(p_{k,8})$	$p_{k,7}$	$p_{k,6}$	$p_{k,5}$	$p_{k,4}$	$p_{k,3}$	$p_{k,2}$	$p_{k,1}$	$p_{k,0}$	c_k
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	$\overline{x_7}$	$\overline{x_7}$	$\overline{x_6}$	$\overline{x_5}$	$\overline{x_4}$	$\overline{x_3}$	$\overline{x_2}$	$\overline{x_1}$	$\overline{x_0}$	1
2	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	0	0
-2	$\overline{x_7}$	$\overline{x_6}$	$\overline{x_5}$	$\overline{x_4}$	$\overline{x_3}$	$\overline{x_2}$	$\overline{x_1}$	$\overline{x_0}$	1	1
-0	1	1	1	1	1	1	1	1	1	1

where $X(x_{n-1}x_{n-2}\ldots x_1x_0)$ and $Y(y_{n-1}y_{n-2}\ldots y_1y_0)$ are two n-bit signed numbers, representing multiplicand and multiplier, respectively. For the convenience of Booth encoding, n is assumed to be an even number.

Table II further exemplifies the k-th Booth encoded partial product, PP_k, generated from each possible Booth encoded digit for an 8×8-bit multiplication, where $p_{k,i(i=0,1,\ldots,8)}$ represents the i-th bit of the k-th partial product.

A decomposition of an $n \times n$-bit fixed-width Booth multiplier is illustrated in Fig. 1. A simplified sign extension method introduced in [14] is utilized, in which s_k indicates an extended sign bit of the k-th partial product, $p_{k,i}$ means the i-th bit of the k-th partial product in the array, c_k is the correction bit shown in Table I. The partial products array is divided into two: accurate part (AP) and truncation part (TP). The former is used for an accurate accumulation, while the latter is reserved for an approximate estimation. Since for a fixed-width multiplier, its output, P_T, should be quantized to n bits as its input. Therefore, we have:

$$P_T = AP + \sigma \approx X \times Y \tag{2}$$

where σ represents the compensation value estimated from TP and been added in the least significant column of AP. As shown in Fig. 1, AP is accumulated as:

$$\begin{aligned} AP &= (s_0 + p_{1,n-2} + \cdots + p_{n/2-1,2})2^0 \\ &\quad + (s_0 + p_{1,n-1} + \cdots + p_{n/2-1,3})2^1 \\ &\quad + \cdots + (1 + p_{n/2-1,n-1})2^{n-3} + (\overline{s_{n/2-1}})2^{n-2} + 2^{n-1}. \end{aligned} \tag{3}$$

For a post-truncated Booth multiplier (PTBM), both AP and TP are taken into calculation. It rounds off the product after a complete accumulation of all partial products, thus, σ_{PTBM} is derived as:

$$\sigma_{PTBM} = \left\lfloor TP + 1 \cdot 2^{-1} \right\rfloor \tag{4}$$

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on December 19,2021 at 06:28:50 UTC from IEEE Xplore. Restrictions apply.

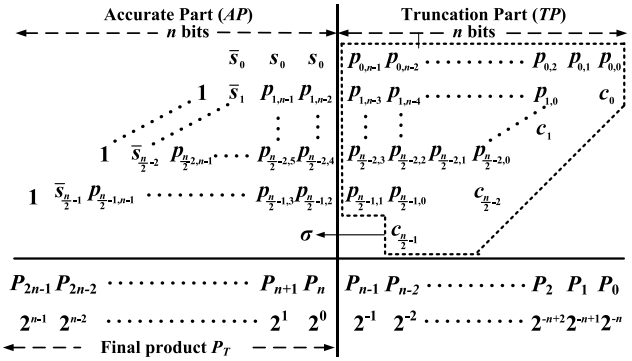


Fig. 1. Partial products array of an $n \times n$-bit Booth multiplier.

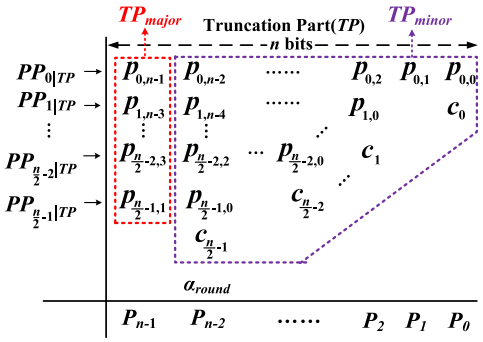


Fig. 2. Partitioning on TP array.

where $TP = (p_{0,0}+c_0)2^{-n}+(p_{0,1})2^{-n+1}+\ldots+(p_{0,n-1}+\ldots+p_{n/2-1,1})2^{-1}$, $\lfloor\cdot\rfloor$ is the floor operator mapping a real number to the largest integer of calculated result, and the extra bit "1" represents a rounding constant to be added in the most significant column of TP.

For a direct-truncated Booth multiplier (DTBM), TP is directly discarded without any compensation, so we have:

$$\sigma_{DTBM} = 0. \tag{5}$$

It is obvious that the simpler formula σ is, the less hardware complexity incurs, and also, the lower accuracy of the calculation could achieve. Therefore, PTBM and DTBM can be regarded as two boundary cases among various fixed-width Booth multipliers. The former leads to the highest accuracy with the most hardware expenses; while the latter results in the largest truncation error but the least hardware resources. Obtaining an appropriate compensation value could be a better tradeoff between these two boundary cases.

III. Proposed Probabilistic Prediction Formula (PPF)

Fig. 2 illustrates the partitioning on the truncation array in the proposed algorithm. As shown, TP in Fig. 1 is further divided into two parts: TP_{major} reserved for an accurate calculation same as AP; TP_{minor} evaluated with a probabilistic estimation in terms of its mean value, $\mu(TP_{minor})$; in addition, a rounding constant, $\alpha_{round} = 1$, is necessary for the truncation in the most significant column of TP_{minor}. Therefore, the compensation formula σ can be expressed as:

$$\sigma = TP_{major} + \mu\left(TP_{minor}\right) + \alpha_{round} \cdot 2^{-2}. \tag{6}$$

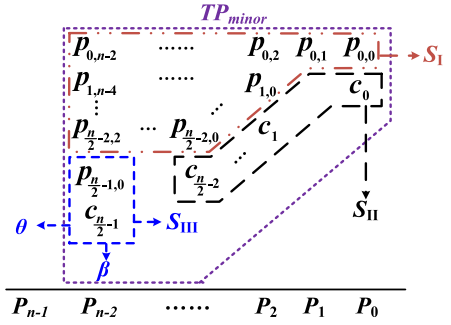


Fig. 3. Further partitioning on TP_{minor} array.

As $\mu(TP_{minor})$ in (6) can be obtained from the expectations of each partial product ($PP_{0|TP_{minor}} \sim PP_{\frac{n}{2}-1|TP_{minor}}$) and the corresponding correction bits ($c_0 \sim c_{\frac{n}{2}-1}$), thus we have:

$$\mu(TP_{minor}) = \sum_{k=0}^{\frac{n}{2}-1}\sum_{i=0}^{n-2-2k} E\left\{p_{k,i}\right\}\cdot 2^{i+2k-n} + \sum_{k=0}^{\frac{n}{2}-1} E\left\{c_k\right\}\cdot 2^{2k-n} \tag{7}$$

where $E\{\cdot\}$ indicates the expectation, $p_{k,i}$ represents the partial product bit in the k-th row and the i-th column, c_k is the correction bit as mentioned in Table I.

In order to make the array of TP_{minor} more compact, a pre-calculation is involved in the proposed algorithm. Thus, TP_{minor} is further partitioned into 3 sections, $S_{\mathrm{I}} \sim S_{\mathrm{III}}$, as indicated in Fig. 3. Thus, (7) can be rewritten as (8). Value estimation for each individual section, $S_{\mathrm{I}} \sim S_{\mathrm{III}}$, are elaborated one by one after that.

$$\begin{aligned}\mu\left(TP_{minor}\right) = &\sum_{k=0}^{\frac{n}{2}-2}\sum_{i=0}^{n-2-2k} E\{p_{k,i}\}\cdot 2^{i+2k-n} \\ &+\sum_{k=0}^{\frac{n}{2}-2} E\{c_k\}\cdot 2^{2k-n} \\ &+\left(E\left\{p_{\frac{n}{2}-1,0}\right\}+E\left\{c_{\frac{n}{2}-1}\right\}\right)\cdot 2^{-2}.\end{aligned} \tag{8}$$

①S_{I}: This section includes the partial products from $PP_{0|TP_{minor}}$ to $PP_{\frac{n}{2}-2|TP_{minor}}$. As the probability of each input bit is assumed to be uniformly distributed, each combination of three adjacent input bits has the equal possibilities of being 1/8. As a "0" is always supplemented to the least significant bit (LSB) of the multiplier in modified Booth encoding, half of the combinations in the first partial product are thus excluded. Consequently, the possibilities of PP_0 being $0X$, $\pm 1X$, and $-2X$ are all 1/4. Therefore, the expected value of $p_{0,0}$ can be calculated as follows:

$$\begin{aligned}E\left\{p_{0,0}\right\} &= \sum\nolimits_{j=0,\pm1,-2} P\left\{p_{0,0}=1|d_0=j\right\}\cdot P\left\{d_0=j\right\} \\ &= 0\cdot\frac{1}{4}+\frac{1}{2}\cdot\frac{1}{4}+\frac{1}{2}\cdot\frac{1}{4}+1\cdot\frac{1}{4}=\frac{1}{2}\end{aligned} \tag{9}$$

where $P\{\cdot\}$ indicates an expected-probability operation. The rest bits of $PP_{0|TP_{minor}}$ can be similarly calculated as:

$$\begin{aligned}E\left\{p_{0,j}\right\} &= \sum\nolimits_{j=0,\pm1,-2} P\left\{p_{0,i}=1|d_0=j\right\}\cdot P\left\{d_0=j\right\} \\ &= 0\cdot\frac{1}{4}+\frac{1}{2}\cdot\frac{1}{4}+\frac{1}{2}\cdot\frac{1}{4}+\frac{1}{2}\cdot\frac{1}{4}=\frac{3}{8} \\ &\quad (i=1,2,\cdots,n-2)\end{aligned} \tag{10}$$

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on December 19,2021 at 06:28:50 UTC from IEEE Xplore. Restrictions apply.

For the other partial product bits in S_{I}, the expected values can be calculated through (11) and (12), for the LSBs and non LSBs, respectively.

$$E\left\{p_{k,0}\right\} = \sum\nolimits_{j=\pm0,\pm1,\pm2} P\left\{p_{k,0}=1|d_k=j\right\}\cdot P\left\{d_k=j\right\}$$
$$= 0\cdot\frac{1}{8}+1\cdot\frac{1}{8}+\frac{1}{2}\cdot\frac{1}{4}+\frac{1}{2}\cdot\frac{1}{4}+0\cdot\frac{1}{8}+1\cdot\frac{1}{8}=\frac{1}{2}$$
$$(k=1,2,\cdots,\frac{n}{2}-2) \tag{11}$$

$$E\left\{p_{k,i}\right\} = \sum\nolimits_{j=\pm0,\pm1,\pm2} P\left\{p_{k,i}=1|d_k=j\right\}\cdot P\left\{d_k=j\right\}$$
$$= 0\cdot\frac{1}{8}+1\cdot\frac{1}{8}+\frac{1}{2}\cdot\frac{1}{4}+\frac{1}{2}\cdot\frac{1}{4}+\frac{1}{2}\cdot\frac{1}{8}+\frac{1}{2}\cdot\frac{1}{8}=\frac{1}{2}$$
$$\left(k=1,2,\cdots,\frac{n}{2}-2;i=1,2,\cdots,n-2-2k\right). \tag{12}$$

②S_{II}: This section deals with the correction bits for each partial product except $c_{\frac{n}{2}-1}$. Similarly, the expected values of the correction bits can be calculated through (13) and (14), for the LSBs and non LSBs, respectively.

$$E\left\{c_0\right\} = \sum\nolimits_{j=0,\pm1,-2} P\left\{c_0=1|d_k=j\right\}\cdot P\left\{d_k=j\right\}$$
$$= 0\cdot\frac{1}{4}+0\cdot\frac{1}{4}+1\cdot\frac{1}{4}+1\cdot\frac{1}{4}=\frac{1}{2} \tag{13}$$

$$E\left\{c_k\right\} = \sum\nolimits_{j=\pm0,\pm1,\pm2} P\left\{c_k=1|d_k=j\right\}\cdot P\left\{d_k=j\right\}$$
$$= 0\cdot\frac{1}{8}+1\cdot\frac{1}{8}+0\cdot\frac{1}{4}+1\cdot\frac{1}{4}+0\cdot\frac{1}{8}+1\cdot\frac{1}{8}=\frac{1}{2}$$
$$\left(k=1,2,\cdots,\frac{n}{2}-2\right). \tag{14}$$

③S_{III}: This section involves a pre-calculation. As shown in Fig. 3, instead of figuring out the expected values of $PP_{\frac{n}{2}-1|TP_{minor}}$ and its correction bit $c_{\frac{n}{2}-1}$, an addition operation is performed on them which intends to reduce the number of partial products in TP_{minor} and make the array more compact.

Suppose β and θ are sum and carry-in bit from the addition of $p_{\frac{n}{2}-1,0}$ and $c_{\frac{n}{2}-1}$. Thus, θ and β stay in TP_{major} column and the most significant column of TP_{minor} array, respectively, as shown in Fig. 3. Based on the $(\frac{n}{2}-1)$-th Booth encoded digit, $d_{\frac{n}{2}-1}$, corresponding β and θ can be obtained. Table III summarizes all possible cases of $d_{\frac{n}{2}-1}$ according to the triplet $\{y_{n-1},y_{n-2},y_{n-3}\}$, which are the three MSBs of the multiplier Y. Therefore, the logical expression of θ is derived as:

$$\theta = \overline{(x_0\cdot(y_{n-2}\oplus y_{n-3}))}\cdot y_{n-1} \tag{15}$$

where x_0 is the LSB of the multiplicand X.

Since β has an equal possibility of being "0" and x_0 according to Table III, its expected value is calculated as:

$$E\left\{\beta\right\} = 0\cdot\frac{1}{2}+E\left\{x_0\right\}\cdot\frac{1}{2} = 0\cdot\frac{1}{2}+\frac{1}{2}\cdot\frac{1}{2}=\frac{1}{4}. \tag{16}$$

Summarized from the above three sections, $S_{\mathrm{I}} \sim S_{\mathrm{III}}$, all expected values in TP_{minor} array are therefore figured out and annotated as shown in Fig. 4.

TABLE III
A PRE-ADDITION TRUTH TABLE BASED ON ALL POSSIBLE VALUES OF $(\frac{n}{2}-1)$-TH BOOTH ENCODED DIGIT

y_{n-1}	y_{n-2}	y_{n-3}	$d_{\frac{n}{2}-1}$	$c_{\frac{n}{2}-1}$	$p_{\frac{n}{2}-1,0}$	β	θ
0	0	0	0	0	0	0	0
0	0	1	1	0	x_0	x_0	0
0	1	0	1	0	x_0	x_0	0
0	1	1	2	0	0	0	0
1	0	0	-2	1	1	0	1
1	0	1	-1	1	$\overline{x_0}$	x_0	$\overline{x_0}$
1	1	0	-1	1	$\overline{x_0}$	x_0	$\overline{x_0}$
1	1	1	0	1	1	0	1

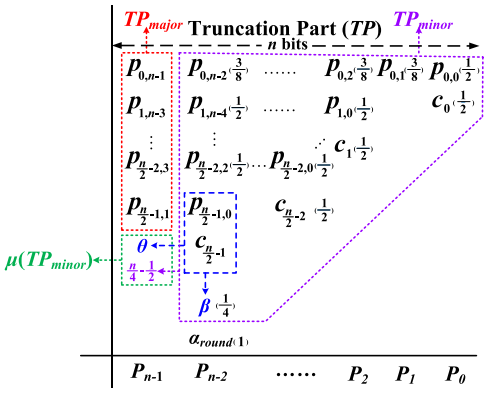


Fig. 4. Expected values annotated in TP_{minor}.

Consequently, $\mu(TP_{minor})$ in (8) can be further derived as:

$$\mu\left(TP_{minor}\right) = \sum\nolimits_{k=0}^{\frac{n}{2}-2}\sum\nolimits_{i=0}^{n-2-2k} E\left\{p_{k,i}\right\}\cdot 2^{i+2k-n}$$
$$+\sum\nolimits_{k=0}^{\frac{n}{2}-2} E\left\{c_k\right\}\cdot 2^{2k-n}+E\left\{\beta\right\}\cdot 2^{-2}+\theta\cdot 2^{-1}$$
$$= \left[\frac{3}{8}\cdot(2^{-2}+\cdots+2^{-n+1})+\frac{1}{2}\cdot(2^{-n}+2^{-n})\right]$$
$$+\left(\frac{n}{4}-\frac{7}{8}+\theta\right)\cdot 2^{-1}. \tag{17}$$

As observed in Fig. 4, $p_{0,0}$ and c_0 have negligible contributions to the compensation formula, in that sense, 3/8 instead of 1/2 is taken as an approximation for the purpose of simplification. Thus, (17) can be derived as:

$$\mu\left(TP_{minor}\right) \approx \left[\frac{3}{8}\cdot(2^{-2}+\cdots+2^{-n+1})+\frac{3}{8}\cdot(2^{-n}+2^{-n})\right]$$
$$+\left(\frac{n}{4}-\frac{7}{8}+\theta\right)\cdot 2^{-1}$$
$$= \left(\frac{n}{4}-\frac{1}{2}+\theta\right)\cdot 2^{-1}. \tag{18}$$

With (18), the compensation formula in (6) is derived as:

$$\sigma = TP_{major}+\mu(TP_{minor})+\alpha_{round}\cdot 2^{-2}$$
$$= TP_{major}+\left(\frac{n}{4}+\theta\right)\cdot 2^{-1} \tag{19}$$

where $TP_{major} = (p_{0,n-1}+p_{1,n-3}+\ldots+p_{\frac{n}{2}-1,1})\cdot 2^{-1}$.

As θ is obtained from (15), an integer constant compensation value σ, can thus be easily figured out based on the

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on December 19,2021 at 06:28:50 UTC from IEEE Xplore. Restrictions apply.

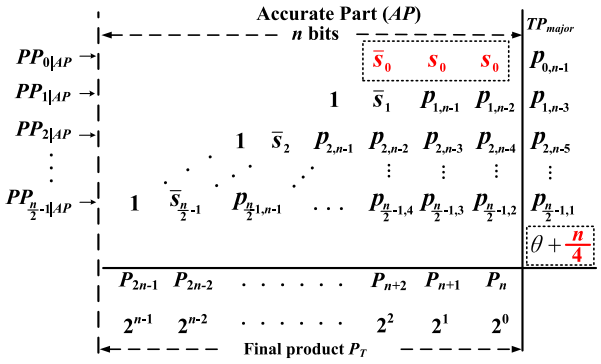


Fig. 5. Partial products array of PPF for compression.

TABLE IV
MAPPING TABLE FOR OPERAND LENGTHS BEING MULTIPLES OF 8

n	Binary number	$PP_{0\|AP}$
8	{0, 0, 1}	$\{1, 0, \overline{s_0}\}$
16	{0,1,0}	$\{1, \overline{s_0}, s_0\}$
24	{0,1,1}	$\{1, 1, \overline{s_0}\}$
32	{1,0,0}	$\{\overline{s_0}, s_0, s_0, s_0\}$
40	{1,0,1}	$\{1, 0, 0, \overline{s_0}\}$
48	{1,1,0}	$\{1, 0, \overline{s_0}, s_0\}$
56	{1,1,1}	$\{1, 0, 1, \overline{s_0}\}$
64	{1,0,0,0}	$\{1, \overline{s_0}, s_0, s_0\}$

operand length n. It implies that when n is a multiple of 4, the truncation error could be well compensated through a simple one-column compression with nearly zero mean error.

Fig. 5 shows the partial products array of proposed fixed-width Booth multiplier for compression, where a single column TP_{major} is used for compensating the truncation error. For a given operand length, as "$\frac{n}{4}$" is a constant, the partial product of $PP_{0|AP}$ ($\{\overline{s}_0, s_0, s_0\}$) can be reorganized by taking both "$\frac{n}{4}$" and θ into account at the same time.

For an arbitrary even number n, the reorganization of $PP_{0|AP}$ is discussed in the following 3 cases:

① n is a multiple of 8

In this case, $PP_{0|AP} = \{\overline{s}_0, s_0, s_0\} + \{\frac{n}{8}\}$. Table IV maps the partial products for various operand lengths vary from 8 to 64 bits. It can be easily extended for even larger operand lengths.

② In this case, $PP_{0|AP} = \{\overline{s}_0, s_0, s_0|0\} + \{0, 0, 0|\frac{n}{4}\} + \{0, 0, 0|\theta\}$ where $\{*|\bullet\}$ means a partial product, '$*$' and '$\bullet$' indicate its AP and TP_{major} part, respectively. Table V maps the partial products for various operand lengths vary from 4 to 60 bits. It can be also extended for even larger operand lengths, which are multiples of 4.

③ The other cases

For the rest even numbers, as a rounding constant, α_{round}, is already used in the derivation of compensation formula, the remaining bits in the most significant column of TP_{minor} are ignored. Thus, they can be rounded to the nearest integers being multiples of 8. Consequently, in this case, $PP_{0|AP}$ can be obtained according to Table IV as well.

TABLE V
MAPPING TABLE FOR OPERAND LENGTHS BEING MULTIPLES OF 4

n	Binary number	$PP_{0\|AP}$
4	{0,0,0|1}	$\{\theta + \overline{s_0}, \overline{\theta} \cdot s_0, \theta \oplus s_0\}$
12	{0,0,1|1}	$\{1, \theta \cdot \overline{s_0}, \theta \odot s_0\}$
20	{0,1,0|1}	$\{1, \theta + \overline{s_0}, \theta \oplus s_0\}$
28	{0,1,1|1}	$\{\theta \cdot \overline{s_0}, \overline{\theta} + s_0, \overline{\theta} + s_0, \theta \odot s_0\}$
36	{1,0,0|1}	$\{\theta + \overline{s_0}, \overline{\theta} \cdot s_0, \overline{\theta} \cdot s_0, \theta \oplus s_0\}$
44	{1,0,1|1}	$\{1, 0, \theta \cdot \overline{s_0}, \theta \odot s_0\}$
52	{1,1,0|1}	$\{1, 0, \theta + \overline{s_0}, \theta \oplus s_0\}$
60	{1,1,1|1}	$\{1, \theta \cdot \overline{s_0}, \overline{\theta} + s_0, \theta \odot s_0\}$

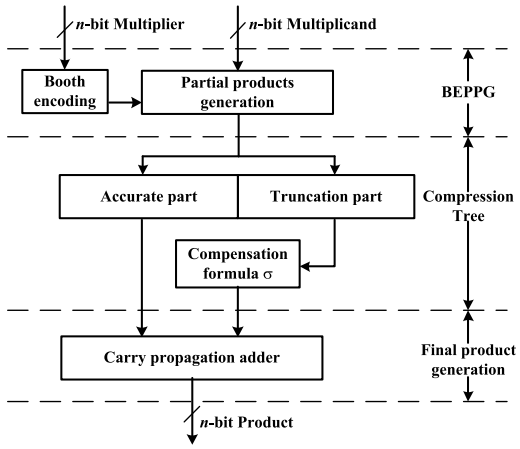


Fig. 6. The architecture of proposed fixed-width Booth multiplier.

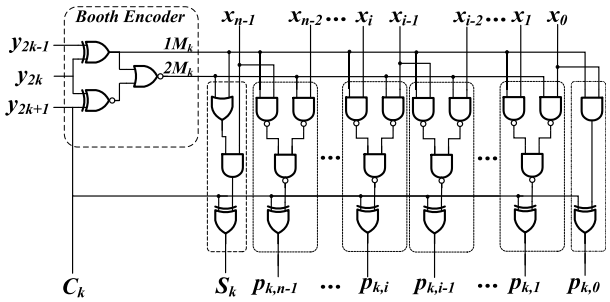


Fig. 7. The k-th partial product generation with Booth encoding.

IV. PROBABILISTIC PREDICTION BASED FWBM DESIGN

A. Architecture of Proposed FWBM

Fig. 6 illustrates the architecture of proposed fixed-width Booth multiplier consists of three major building blocks, namely, Booth encoding and partial product generation (BEPPG), compression tree, and carry propagation adder (CPA). The BEPPG module generates the partial product bits according to the Booth encoded multiplier and the corresponding multiplicand. These partial products form an array, which is implemented with a compression tree including one compensation component. Finally, a two-operand CPA is employed to output the final product. The choice of fast adder structures depends on the number of operand lengths.

B. Hardware Implementation

Fig. 7 shows the gate-level circuit implementation of the k-th partial product generation with Booth encoding.

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on December 19,2021 at 06:28:50 UTC from IEEE Xplore. Restrictions apply.

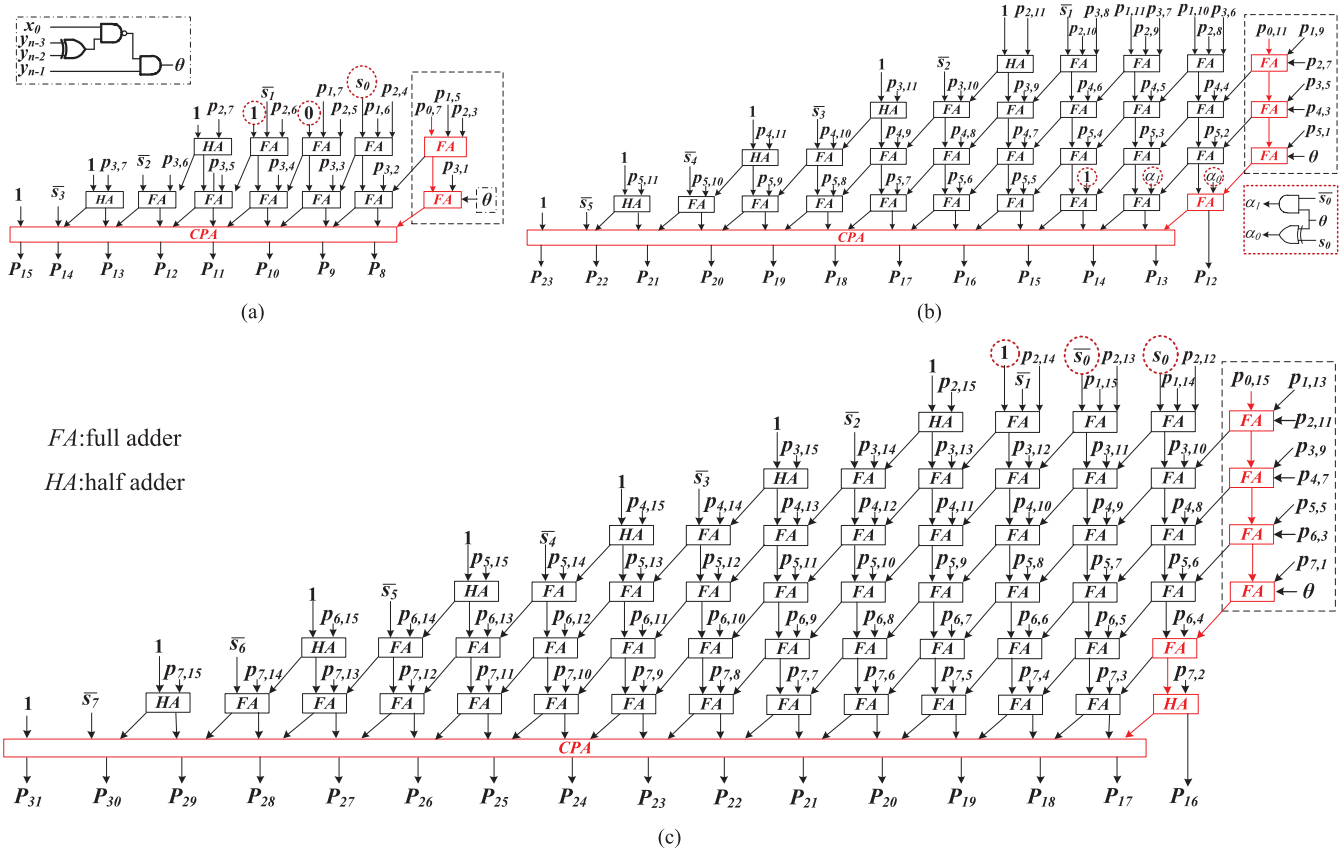


Fig. 8. Circuit implementations of proposed compensation formula for (a) operand length $n = 8$ (b) operand length $n = 12$ (c) operand length $n = 16$.

Notice that S_k and $p_{k,0}$ indicate the MSB and LSB of the k-th partial product, respectively.

Based on the proposed formula in (19), the compensation value σ can be derived once the two operands and their bit widths are determined. Corresponding $PP_{0|AP}$ is thus generated according to the mapping Tables IV and V.

Figs. 8(a)-(c) illustrate the implementations of compression trees with proposed compensation formula for the operand lengths being 8, 12, and 16 bits, respectively. A carry-save reduction array is employed with both full-adders and half adders. A parallel prefix carry propagation adder can be used for fast addition. The critical paths for each design are highlighted in red. Notice that θ is implemented with three simple logic gates as indicated in the top left corner of Fig. 8, which makes it not locating in the critical path. In addition, the red dotted circles in Figs 8(a)-(c) indicate the reorganized partial product of $PP_{0|AP}$?Cwhich prevents an additional constant partial product from being generated due to the term of "$n/4$" in the compensation formula. This not only makes the array more compact, but also leads to a bit faster accumulation.

V. Performance Evaluation

In this section, experimental results of various recently proposed fixed-width Booth multiplier are presented for comparisons and analyses. The chip implementation and the measurement results of our proposed design, as well as its DCT application are provided for further performance evaluation.

A. Error Performance

To evaluate the accuracy of different fixed-width Booth multipliers, the error metrics in terms of normalized mean error ($\bar{\varepsilon}$), maximum absolute error ($|\varepsilon|_{max}$), and mean absolute error ($\overline{|\varepsilon|}$) are introduced and defined as follows:

$$\begin{aligned} \bar{\varepsilon} &= E\{P_A - P_T\}/2^n \\ |\varepsilon|_{max} &= max\{|P_A - P_T|\}/2^n \\ \overline{|\varepsilon|} &= E\{|P_A - P_T|\}/2^n \end{aligned} \tag{20}$$

where P_A and P_T indicate accurate and truncated product, respectively; $\bar{\bullet}$, $|\bullet|$, and $max\{\bullet\}$ are the average, absolute and maximum operators, respectively. Notice that the higher computation accuracy in multipliers, the smaller error metrics should be obtained. The smaller $\bar{\varepsilon}$ and $\overline{|\varepsilon|}$ indicate a more symmetric and centralized error distribution with respect to zero, which are important for many multimedia and DSP applications whose final output data are normally generated by accumulating a number of multiplication products [14]. On the other hand, a larger maximum error may cause unbearable random malfunctioning. Therefore, $|\varepsilon|_{max}$ is an equally important performance parameter especially in safety critical applications in which reliability is emphasized [20].

In order to obtain the error performance, exhaustive simulations are performed on various fixed-width Booth multipliers, including PTBM, DTBM, JFM [12], PEB [15], SCG [14], MLCP [17], and BSCP [19], with the operand lengths being 8, 12, and 16 bits. Table VI summarizes the results. As the result shows, the proposed PPF Booth multiplier exhibits the

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on December 19,2021 at 06:28:50 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
ERROR PERFORMANCE OF DIFFERENT FIXED-WIDTH BOOTH MULTIPLIERS

n	Multiplier	$\bar{\varepsilon}$	$\lvert\varepsilon\rvert_{max}$	$\overline{\lvert\varepsilon\rvert}$
8	PTBM	0.000	0.500	0.249
	DTBM	1.501	4.000	1.501
	BSCP[19]	0.008	1.168	0.299
	MLCP[17]	0.133	1.500	0.330
	SCG[14]	0.008	1.168	0.302
	PEB[15]	0.000	1.500	0.347
	JFM[12]	0.001	1.731	0.418
	PPF (This work)	**0.000**	**1.246**	**0.343**
12	PTBM	0.000	0.500	0.250
	DTBM	2.250	6.000	2.250
	BSCP[19]	0.002	1.667	0.328
	MLCP[17]	0.127	2.000	0.357
	SCG[14]	0.002	1.667	0.332
	PEB[15]	0.124	2.000	0.404
	JFM[12]	0.000	2.530	0.509
	PPF (This work)	**0.000**	**1.750**	**0.405**
16	PTBM	0.000	0.500	0.250
	DTBM	3.000	8.000	3.000
	BSCP[19]	0.001	2.167	0.354
	MLCP[17]	0.125	2.500	0.381
	SCG[14]	0.001	2.167	0.358
	PEB[15]	0.250	2.500	0.477
	JFM[12]	0.000	3.330	0.585
	PPF (This work)	**0.000**	**2.250**	**0.461**

TABLE VII
CIRCUIT PERFORMANCE OF DIFFERENT FIXED-WIDTH BOOTH MULTIPLIER

n	Multipliers	Area (10^3·μm^2)	Delay (ns)	Power (mW)	Energy (pJ)
8	PTBM	3.035	3.89	1.61	6.26
	DTBM	1.443	2.64	0.66	1.74
	BSCP[19]	2.413	2.89	0.91	2.63
	MLCP[17]	2.423	2.91	0.98	2.85
	SCG[14]	2.441	2.91	0.95	2.76
	PEB[15]	2.275	3.36	1.01	3.39
	JFM[12]	2.223	3.29	0.83	2.73
	PPF (This work)	**1.725**	**2.76**	**0.80**	**2.21**
12	PTBM	6.655	5.63	4.15	23.36
	DTBM	3.165	3.79	1.85	7.01
	BSCP[19]	5.061	4.37	2.21	9.66
	MLCP[17]	4.859	4.33	2.48	10.74
	SCG[14]	4.995	4.51	2.34	10.55
	PEB[15]	4.652	5.04	2.54	12.80
	JFM[12]	4.532	4.85	2.17	10.52
	PPF (This work)	**3.592**	**3.82**	**2.11**	**8.06**
16	PTBM	11.200	7.13	5.57	39.71
	DTBM	5.532	4.61	3.00	13.83
	BSCP[19]	8.238	5.28	3.82	20.17
	MLCP[17]	8.088	5.68	3.56	20.22
	SCG[14]	8.290	5.38	3.98	21.41
	PEB[15]	7.933	6.22	3.98	24.76
	JFM[12]	7.641	5.98	3.36	20.09
	PPF (This work)	**6.177**	**4.94**	**3.24**	**16.01**

least mean error as much as PTBM achieves. This is due to the way that the error compensation function is derived as discussed in Section III. The proposed design also provides a superior performance in $|\varepsilon|_{max}$ compared with PEB and MLCP. However, it shows 22.81% performance loss in $\overline{|\varepsilon|}$ on average compared with BSCP. That is due to the reason that BSCP takes more information from the sign bit of Booth encoding for producing the compensation function, which in turn increases the hardware expenses.

B. Circuit Performance

Circuit performance is evaluated in terms of area cost, propagation delay, and power dissipation, since these design metrics are critical to a multiplier design. Table VII presents the result of various multipliers with the operand lengths being 8, 12, and 16 bits.

For a fair and legitimate comparison, all fixed-width Booth multipliers are implemented with the same Booth encoding method and compression tree structure. Each multiplier is described at gate level in Verilog HDL. They are synthesized and mapped to a 0.18-μm CMOS standard-cell library using the Synopsys Design Compiler (DC) with a specified wire load model. The option for logic structuring was turned off to prevent the tool from changing the structure of the unit cells. All simulations are carried out at a supply voltage of 1.8V and a room temperature of 25°C. The average power dissipations are simulated by Synopsys Power Compiler with back annotated switching activity files (SAIF) generated from 100,000 pseudo random input vectors at a 100MHz data rate.

As indicated in Table VII, the proposed Booth multiplier occupies the least area and dissipates the least power when compared with the existing works. This is due to our simple one-column compression compensation function. Also thanks to the partial product array reorganization strategy, the proposed multipliers run faster than the others. Benefit from the improved speed, and power dissipation, the proposed design exhibits the least energy consumption among all competitors with various bit lengths.

C. Design Efficiency Analysis

In general, all fixed-width multiplier designs with compensation function present tradeoffs between accuracy and hardware expenses. For better characterizing the multipliers in comparison, a combined parameter is proposed to evaluate the design efficiency, which includes the delay time, energy consumption, and area cost since they are important design metrics for hardware implementation. While for error-tolerant applications, to maintain a required accuracy, error metric should also be taken into account. Therefore, $P_{\mathrm{AED}}\varepsilon$ is made as a combined measure, which is given as follows:

$$P_{AED\varepsilon} = Area(A) \times Energy(E) \times Delay(D) \times \overline{|\varepsilon|}. \quad (21)$$

Notice that different error metrics may be chosen for different scenarios in error-tolerant applications. $\overline{|\varepsilon|}$ is used in (21) for the purpose of delivering a more objective evaluation since our proposed design performs relatively poor in mean absolute error.

Fig. 9 shows the comparison results for various FWBM with the operand lengths being 8, 12, and 16 bits. Each

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on December 19,2021 at 06:28:50 UTC from IEEE Xplore. Restrictions apply.

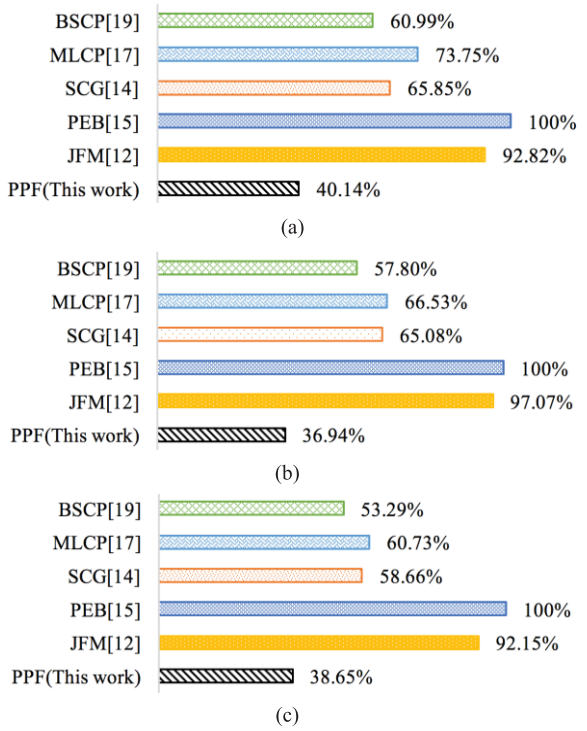


Fig. 9. $P_{AED\varepsilon}$ with different fixed-width Booth multiplier. (a) operand length $n = 8$ (b) operand length $n = 12$ (c) operand length $n = 16$.

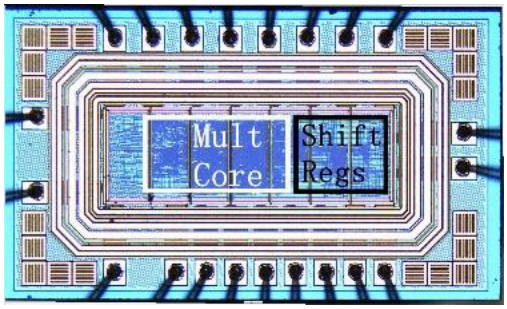


Fig. 10. Chip photomicrograph of proposed PPF Booth multiplier.

design is normalized with respect to PEB since its $P_{AED\varepsilon}$ is the largest for each operand length. As the results indicate, the proposed design outperforms the others and exhibits the best performance-accuracy tradeoff for all operand lengths. When compared with PEB, which uses a similar probabilistic estimation method, the proposed design achieves at least 60% reduction on $P_{AED\varepsilon}$. Compared with the most competitive BSCP, the proposed design achieves 34%, 36% and 27% reduction on $P_{AED\varepsilon}$ for operand lengths of 8, 12, and 16 bits, respectively.

D. Chip Implementation

In order to verify the advantages of the proposed design, a 16×16-bit PPF FWBM is fabricated using a 0.18-μm standard CMOS process. The chip photomicrograph is illustrated in Fig. 10. Series input/output and serial-to-parallel/parallel-to-serial buffers are employed to reduce the number of I/O ports. Several input/output ports are reserved for debug purpose in order to improve the testability of the chip. The functionality verification is performed at clock frequencies starting from

TABLE VIII
MEASUREMENT RESULTS

Process Technology	0.18-μm CMOS, 1P6M
Supply Voltage	1.8V
Core Size	832μm×270μm
Max Frequency	100MHz
Power Consumption	3.30mW @ 100MHz

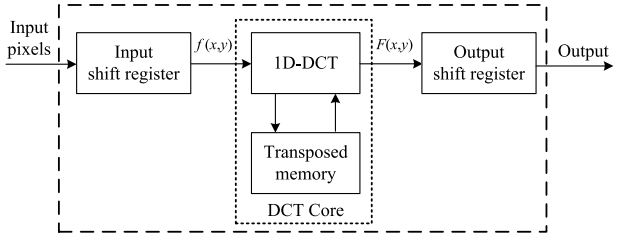


Fig. 11. Block diagram of the 2-D DCT implementation.

80MHz upwards in steps of 5MHz with exhaustive input data patterns generated by SignalTap II logic analyzer. Table VIII presents the measurement results. The resulting chip is capable of a maximum operating frequency of 100 MHz with power consumption of only 3.30mW. As noticed, a separated pair of supply is used dedicated for the proposed PPF multiplier core, the current is thus measured and the power dissipation is obtained accordingly.

E. Application to 2-D DCT Computations

The proposed design is applied to 2-D DCT computation for further demonstrating the performance in real applications. The 2-D DCT circuit is implemented using fast row/column decomposition algorithm with shift register arrays and one 1-D DCT core, which contains 28 12-bit Booth multipliers and a transposed memory [21]. The block diagram of the 2-D DCT implementation is shown in Fig. 11.

To check the computation accuracy, ten commonly used test images are employed, each of which is comprised of 512×512 pixels. An 8-bit 256 grayscale data represents each image pixel. The original test image pixels are fed into the 2-D DCT module to obtain the transformed output data. The inverse DCT is then computed with these data using 64-bit floating point double-precision operations through MATLAB. The computation accuracy is evaluated by the peak signal-to-noise ratio (PSNR). Seven Booth multipliers including PTBM, DTBM, BSCP, MLCP, SCG, PEB, and our proposed PPF are employed in the 2-D DCT computation one by one. System-level simulations are performed to calculate the PSNR for various DCTs. The results are summarized in Table IX.

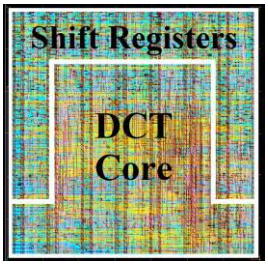
To evaluate the circuit performance, 1-D DCT circuits employed with seven different Booth multipliers are also synthesized and mapped to a 0.18-μm CMOS standard-cell library using the Synopsys DC. All these circuits are simulated at a frequency of 50MHz. The area cost and power dissipation for each design are also summarized in Table IX.

As indicated, the 2-D DCT design using our proposed PPF Booth multiplier presents the closest PSNR to PTBM design.

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on December 19,2021 at 06:28:50 UTC from IEEE Xplore. Restrictions apply.

TABLE IX
COMPARISONS OF DCT COMPUTATION USING DIFFERENT FIXED-WIDTH BOOTH MULTIPLIERS

		PTBM	DTBM	BSCP [19]	MLCP [17]	SCG [14]	JFM [12]	PEB [15]	**PPF (This work)**
PSNR (dB)	Aerial	49.97	29.79	46.48	46.48	46.48	45.15	46.18	**46.61**
	Airport	50.07	29.93	46.25	46.25	46.25	45.59	46.08	**46.44**
	Baboon	49.86	29.65	46.17	46.17	46.17	44.8	45.84	**46.27**
	Barbara	49.92	29.59	46.06	46.06	46.06	44.74	45.85	**46.21**
	Bridge	50.18	29.73	46.24	46.24	46.24	44.58	45.90	**46.37**
	Couple	49.88	29.64	45.96	45.96	45.96	44.99	45.72	**46.11**
	Dollar	49.86	29.92	46.34	46.34	46.34	44.72	46.06	**46.47**
	Goldhill	50.25	29.38	45.67	45.67	45.67	44.55	46.46	**46.88**
	Lena	49.91	28.73	45.92	45.92	45.92	45.14	45.73	**46.11**
	Peppers	49.85	29.75	46.07	46.07	46.07	45.23	45.83	**46.25**
	Average	49.98 (100%)	29.61 (59.2%)	46.12 (92.3%)	46.12 (92.3%)	46.12 (92.3%)	44.95 (89.9%)	45.97 (92.0%)	**46.37 (92.8%)**
Area(10^4·μm^2)		22.82	13.84	16.73	16.25	16.33	17.35	16.61	**15.92**
Power(mW)		9.89	6.33	7.59	7.73	7.61	7.31	7.45	**7.17**



Characteristics	
Technology	0.18-μm CMOS, 1P6M
Supply Voltage	1.8V
Core Size	598μm × 597μm
Max Frequency	50 MHz
Power Consumption	17.60mW @ 50MHz

Fig. 12. Core layout and simulated characteristics of the proposed 2-D DCT.

For 1-D DCT circuit, the proposed approach achieves the least area and power dissipation except DTBM. Therefore, the PPF Booth multiplier has the best tradeoff performance in accuracy and hardware expenses for DCT application. Compared with direct truncation method, it improves more than 16.76 dB PSNR in average at the cost of 15% area increase. Compared with post truncation method, it saves 30% area cost and reduces 28% power dissipation with merely 3.61 dB accuracy loss.

Furthermore, to implement the 2-D DCT circuit with the proposed multiplier on a chip, the RTL design is synthesized, the placement and routing are also performed for the physical design. Fig. 12 shows the core layout and the simulated characteristics of the proposed 2-D DCT circuit. With a 0.18-μm 1P6M standard CMOS process, the proposed 2-D DCT is capable of operating at the frequency of 50MHz with power consumption of 17.6mW.

VI. CONCLUSION

In this paper, a probabilistic prediction based fixed-width Booth multiplier has been proposed. The error compensation function is derived from the probabilistic analysis on the partial products obtained from Booth encoding. By slightly modifying the partial product array with dedicated partitioning, a simple compensation circuit is thus formulated. The proposed approach can be easily adapted for an accuracy-adaptive design by putting variable columns of TP_{major} into considerations. Compared with the previous works, the proposed fixed-width Booth multiplier exhibits the best performance-accuracy tradeoff. It achieves at least 27% reduction on the area-energy-delay-error product compared with all its contenders among various operand lengths. Additionally, the experimental results indicate the proposed Booth multiplier in the DCT application obtains an improvement of PSNR by 16.76 dB with only 15% area penalty when compared with direct-truncated method; while it saves 30% area cost and reduces 28% power dissipation with merely 3.61 dB accuracy loss when compared with post-truncated method. Therefore, the proposed fixed-width Booth multiplier is applicable for lossy applications in approximate computing with considerable reduction on hardware expenses while maintaining decent output quality.

REFERENCES

[1] V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Analysis and characterization of inherent application resilience for approximate computing," in *Proc. 50th Annu. Design Autom. Conf. (DAC)*, Austin, TX, USA, Jun. 2013, pp. 1–9.

[2] Y. Emre and C. Chakrabarti, "Energy and quality-aware multimedia signal processing," *IEEE Trans. Multimedia*, vol. 15, no. 7, pp. 1579–1593, Nov. 2013.

[3] C. Liu, J. Han, and F. Lombardi, "An analytical framework for evaluating the error characteristics of approximate adders," *IEEE Trans. Comput.*, vol. 64, no. 5, pp. 1268–1281, May 2015.

[4] N. Petra, D. De 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.

[5] 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.

[6] 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.

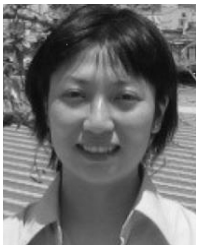
[7] N. Petra, D. De Caro, V. Garofalo, E. Napoli, and A. G. M. Strollo, "Design of fixed-width multipliers with linear compensation function," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 5, pp. 947–960, May 2011.

[8] I.-C. Wey and C.-C. Wang, "Low-error and hardware-efficient fixed-width multiplier by using the dual-group minor input correction vector to lower input correction vector compensation error," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 10, pp. 1923–1928, Oct. 2012.

[9] 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.

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on December 19,2021 at 06:28:50 UTC from IEEE Xplore. Restrictions apply.

[10] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal processing and classification applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 6, pp. 1180–1184, Jun. 2015.
[11] S.-F. Hsiao, J.-H. Zhang Jian, and M.-C. Chen, "Low-cost FIR filter designs based on faithfully rounded truncated multiple constant multiplication/accumulation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 5, pp. 287–291, May 2013.
[12] 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.
[13] 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.
[14] 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.
[15] 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.
[16] Y. H. Chen and T. Y. Chang, "A high-accuracy adaptive conditional-probability estimator for fixed-width booth multipliers," *IEEE Trans. Circuits Syst. I, Reg. Papers, Reg. Papers*, vol. 59, no. 3, pp. 594–603, Mar. 2012.
[17] 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.
[18] 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.
[19] Z. Zhang and Y. He, "A low-error energy-efficient fixed-width booth multiplier with sign-digit-based conditional probability estimation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 2, pp. 236–240, Feb. 2018.
[20] V. Garofalo, N. Petra, and E. Napoli, "Analytical calculation of the maximum error for a family of truncated multipliers providing minimum mean square error," *IEEE Trans. Comput.*, vol. 60, no. 9, pp. 1366–1371, Sep. 2011.
[21] N. Rathore, "JPEG image compression," *Int. J. Eng. Res. Appl.*, vol. 4, no. 3, pp. 435–440, 2014.



Yajuan He (Member, IEEE) received the B.S. degree from East China Normal University (ECNU), Shanghai, China, in 2001, and the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2008.

Since 2009, she has been with the University of Electronic Sciences and Technology of China (UESTC), where she is currently an Associate Professor. Her current research interests include ultralow power design techniques and approximate computing in VLSI design.

Xilin Yi (Graduate Student Member, IEEE) received the B.S. degree in integrated circuit design and integrated system from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2017, where he is currently pursuing the master's degree in integrated circuit engineering.

His research interests include approximate computing and VLSI design.

Ziji Zhang (Student Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013 and 2016, respectively, where he is currently pursuing the Ph.D. degree in electronic engineering.

His research interests include approximate computing and ultralow power design techniques.

Bin Ma received the B.S. and M.S. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2014 and 2017, respectively.

He is currently working with the Department of Short Distance Wireless Communication, HiSilicon Technologies Company Ltd.

Qiang Li (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2001, and the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2007.

Since 2001, he has been working on analog/RF circuits in both academia and industry, holding positions of engineer, project leader, and technical consultant in Singapore and Associate Professor in Denmark. He is currently a Full Professor with the University of Electronic Science and Technology of China (UESTC), heading the Institute of Integrated Circuits and Systems. His research interests include low-voltage and low-power analog/RF circuits, data converters, and mixed-mode circuits for biomedical and sensor interfaces.

Dr. Li is a member of the Student Research Preview (SRP) Committee of ISSCC and the Technical Program Committee of ESSCIRC and ASSCC. He was a recipient of the Young Changjiang Scholar Award in 2015, the UESTC Teaching Excellence Award in 2011, and the Service Excellence Award in 2018. He was the TPC Chair of the IEEE 2018 APCCAS. He is the Founding Chair of the IEEE SSCS/CASS Chengdu Chapter. He serves as a Guest Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS (TCAS-I) and an Associate Editor for the IEEE OPEN JOURNAL OF CIRCUITS AND SYSTEMS (OJCAS).

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on December 19,2021 at 06:28:50 UTC from IEEE Xplore. Restrictions apply.