Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing

Weiqiang Liu, *Senior Member, IEEE*, Liangyu Qian, Chenghua Wang, Honglan Jiang, Jie Han, *Senior Member, IEEE*, and Fabrizio Lombardi, *Fellow, IEEE*

Abstract—Approximate computing is an attractive design methodology to achieve low power, high performance (low delay) and reduced circuit complexity by relaxing the requirement of accuracy. In this paper, approximate Booth multipliers are designed based on approximate radix-4 modified Booth encoding (MBE) algorithms and a regular partial product array that employs an approximate Wallace tree. Two approximate Booth encoders are proposed and analyzed for error-tolerant computing. The error characteristics are analyzed with respect to the so-called approximation factor that is related to the inexact bit width of the Booth multipliers. Simulation results at 45 nm feature size in CMOS for delay, area and power consumption are also provided. The results show that the proposed 16-bit approximate radix-4 Booth multipliers with approximate factors of 12 and 14 are more accurate than existing approximate Booth multipliers with moderate power consumption. The proposed R4ABM2 multiplier with an approximation factor of 14 is the most efficient design when considering both power-delay product and the error metric NMED. Case studies for image processing show the validity of the proposed approximate radix-4 Booth multipliers.

Index Terms—Radix-4 multiplier, booth encoder, approximate computing, low power

1 Introduction

MULTIPLIERS are widely used in arithmetic units of microprocessors, multimedia and digital signal processors; moreover, high performance and low power multipliers are in high demand for embedded systems. It is becoming extremely difficult to further improve performance and reduce the power consumption of multipliers under the requirement of full accuracy; however, the requirements of high precision and exactness are not so strict for many applications related to human perception, such as multimedia signal processing and machine learning. High precision and exactness in the operations of digital logic circuits are related to the generally accepted requirement of correctness of information processing; numerous error-tolerant applications can be found in computing and by relaxing the requirement of strict accuracy, performance and power consumption can be substantially improved [1]. This design principle is generally known as *approximate or inexact computing* [2].

As the basic operations of an arithmetic processor, addition and multiplication are very important for achieving high performance. Addition has been extensively studied for approximate computing for reduction in power consumption and delay [3], [4], [5]. New metrics including error distance (ED), mean error distance (MED) and normalized error distance (NED) have been proposed for evaluating the designs of approximate adders [6].

Approximate multiplication has not been extensively studied despite its importance for arithmetic processing and systems; multiplication is more complex compared with addition, because it requires the accumulation of partial product rows. The most widely used high performance multiplier consists of a modified Booth encoding (MBE) to reduce the number of partial product rows by half as the first step [7], [8].

The current designs for an approximate multiplier can be categorized as truncation and non-truncation schemes. A truncation-based design relies on a simple approximation in which either the lower part of the partial products is removed, or the least significant partial products are estimated by a constant (so referred to as fixed-width multiplier design [9]); however, the error generated by the truncated partial product rows can be rather large. Therefore, error compensation strategies have been proposed to increase the accuracy of truncated multipliers; an inexact array multiplier has been proposed by ignoring some of the least significant columns of the partial products and considering them as a constant [3]. In [10], the truncated multiplier utilizes a correction constant selected according to both the reduction and rounding errors. However, this truncated multiplier incurs a very large error if the partial products in the least significant columns are all ones or all zeros; therefore, a truncated multiplier with variable correction has been also proposed in [11]. Recently, a few error compensation strategies have been proposed to further improve the accuracy of fixed-width Booth multipliers [9], [12], [13], [14], [15]. An error is compensated with the outputs of the Booth encoders in [9]; the error compensation circuit proposed in [12] mainly uses a simplified sorting network. An adaptive conditional-probability estimator has been proposed in [15] to compensate the quantization error of a fixed-width Booth multiplier. These truncated Booth multipliers use error compensation circuits to improve the accuracy. However, the extra compensation circuits require additional hardware; approximate computing can be employed to reduce such overhead.

A non-truncation scheme utilizes approximate circuits to assemble an approximate multiplier. An approximate 2×2 multiplier has been proposed in [16] by simplifying its logic expression using a Karnaugh-Map (K-map); this circuit is then used as a basic block for larger size multipliers. Compressors or counters are widely used to accelerate the accumulation of partial products in the design of a high-speed multiplier [17], [18]; an inexact 4:2 counter has been used to design an approximate 4×4 Wallace multiplier for assembling larger size multipliers [17]. Two approximate 4-2 compressors are proposed in [18] and used in a Dadda tree of 8×8 array multipliers. An 8×8 multiplier using approximate adders that ignore carry propagation between partial products, has been proposed in [19]. Hybrid Approximate Multiplier (HAM) schemes are proposed by exploring perforation, logic approximation and voltage over-scaling techniques in [20]. It is found that higher power savings for the same error values can be achieved by applying hybrid approximation techniques than using only single design techniques. A dynamic range unbiased multiplier (DRUM) with a dynamic range selection scheme is proposed in [21]; this scheme based on approximation in operands has an unbiased error distribution. In [22], approximate radix-8 Booth multipliers are proposed by using an approximate 2-bit adder for generating the odd multiples (×3) of the multiplicand.

However, there is little study in the technical literature on the approximate design of a radix-4 Booth multiplier as one of the most popular schemes for signed multiplication. In the first step of a radix-4 Booth multiplier, a radix-4 modified Booth encoding (MBE) is used to generate the partial products [23]; a radix-4 MBE

- *W. Liu, L. Qian and C. Wang are with the College of Electronic Information and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China. E-mail: {liuweiqiang, qliangyu, chwang}@nuaa.edu.cn.*
- *H. Jiang and J. Han are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada. E-mail: {honglan, jhan8}@ualberta.ca.*
- *F. Lombardi is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115. E-mail: lombardi@ece.neu.edu.*

Manuscript received 26 Oct. 2016; revised 18 Jan. 2017; accepted 18 Feb. 2017. Date of publication 22 Feb. 2017; date of current version 17 July 2017.
Recommended for acceptance by J.D. Bruguera.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2017.2672976

can reduce the number of partial products by a factor of two. The implementation of the MBE significantly affects the area, delay and power consumption of Booth multipliers [24]; in the traditional MBE algorithm, an extra partial product bit is generated at the least significant bit (LSB) position of each partial product row due to the negative encoding. This leads to an irregular partial product array as requiring a complex reduction tree.

In this paper, two approximate radix-4 MBE algorithms are proposed and analyzed. Booth multipliers are designed based on the proposed radix-4 MBEs, in which a regular partial product array is achieved by using the proposed approximate Wallace tree structure. The error characteristics are analyzed with an approximation factor that is related to the inexact bit width of the Booth multipliers. Simulation results at 45 nm CMOS technology on delay, area, power consumption are also provided. Case studies for image processing are presented to show the validity of the proposed approximate radix-4 Booth multipliers.

This paper is an extension of our previous work presented in [25]; the main differences and novel contributions are summarized as follows:

1) A more efficient approximate radix-4 Booth encoder is proposed in this paper. The designs of both approximate radix-4 Booth encoders are presented and extensively analyzed.
2) Approximate Booth multipliers are proposed using approximate Booth encoders, in which the features of an approximate regular tree structure are illustrated in detail.
3) An approximation factor is proposed to assist in the design of the approximate Booth multipliers and facilitate its error analysis.
4) The proposed approximate Booth multipliers are comprehensively evaluated with respect to both hardware implementation and error analysis.
5) The proposed approximate Booth multipliers are applied in image processing.

The paper is organized as follows. Section 2 presents the design of approximate radix-4 Booth multipliers based on two approximate Booth encoders and an approximate Wallace tree, which are also discussed in detail in this section. Section 3 presents the error analysis and the simulation results on area, delay and power consumption. The application of the proposed approximate Booth multipliers to image processing is presented in Section 4. Conclusion is provided in Section 5.

2 Approximate Radix-4 Booth Multipliers

A Booth multiplier consists of three parts: partial product generation using a Booth encoder, partial product accumulation using compressors and final product generation using a fast adder. The approximate design of radix-4 Booth encoding is studied in this section. A more efficient approximate radix-4 Booth encoding method is proposed in this section by carefully considering the error characteristics. Furthermore, an approximate partial product array produced by the Booth encoding is also designed to make it regular, such that a reduction stage is saved. Approximate Booth multipliers are designed based on the approximate Booth encoder and the regular approximate partial product array.

2.1 Review of Radix-4 Booth Multiplication

Booth encoding has been proposed for improving the performance of multiplication of two's complement binary numbers [7]; it has been further improved by the MBE or radix-4 Booth encoding [8]. The Booth encoder plays an important role in the Booth multiplier, which reduces the number of partial product rows by half.

TABLE 1
K-Map of R4ABE1

$a_j a_{j-1}$ \ $b_{2i+1}b_{2i}b_{2i-1}$	000	001	011	010	110	111	101	100
00	0	0	0	0	1	0	1	⓪
01	0	0	⓪	0	1	0	1	0
11	0	1	⓪	1	0	0	0	0
10	0	1	0	1	0	0	0	⓪

Consider the multiplication of two N-bit integers, i.e., a multiplicand A and a multiplier B in two's complement; this is given as follows:

$$A = -a_{N-1}2^{N-1} + \sum_{i=0}^{N-2} a_i 2^i, \tag{1}$$

$$B = -b_{N-1}2^{N-1} + \sum_{i=0}^{N-2} b_i 2^i. \tag{2}$$

In a Booth encoder, each group is decoded by selecting the partial products as $-2A$, $-A$, 0, A, or $2A$. The negation operation is performed by inverting each bit of A and adding a '1' (defined as Neg) to the LSB [23], [24].

The circuit diagrams of the radix-4 Booth encoder and decoder are provided in [23]. The output (i.e., the partial product, pp_{ij}) of the Booth encoder is given as follows:

$$\begin{aligned} pp_{ij} = {} & (b_{2i} \oplus b_{2i-1})\big(b_{2i+1} \oplus a_j\big) \\ & + \overline{(b_{2i} \oplus b_{2i-1})}(b_{2i+1} \oplus b_{2i})\big(b_{2i+1} \oplus a_{j-1}\big). \end{aligned} \tag{3}$$

2.2 Approximate Radix-4 Booth Encoding Method 1

The K-map of the first approximate radix-4 Booth encoding (R4ABE1) method is shown in Table 1, where ⓪ denotes an entry in which a '1' is replaced by a '0'. Only four entries are modified to simplify the Booth encoding; the strategy for the first approximate design is to make the truth table as symmetrical as possible and introduce a small error. Thus, the advantage of the R4ABE1 design is that a very small error occurs, as only four entries are modified; however, all modifications change a '1' to a '0', so the absolute value of approximate product is always smaller than its exact counterpart.

The modified truth table shows two XOR functions; therefore, the output of R4ABE1 is given as follows:

$$\begin{aligned} pp_{ij} = {} & a_j\overline{b_{2i+1}b_{2i}}b_{2i-1} + a_j\overline{b_{2i+1}}b_{2i}\overline{b_{2i-1}} \\ & + \overline{a_j}b_{2i+1}b_{2i}\overline{b_{2i-1}} + \overline{a_j}b_{2i+1}\overline{b_{2i}}b_{2i-1} \\ = {} & (b_{2i} \oplus b_{2i-1})\big(b_{2i+1} \oplus a_j\big) \end{aligned} \tag{4}$$

Compared with the exact MBE (Eq. (3)), R4ABE1 can significantly reduce both the complexity and the critical path delay of Booth encoding. The error rate (defined as the probability that at least a bit differs from the exact result, given a uniform distribution of inputs), denoted by P_e, is given by:

$$P_e = 4/32 = 12.5\%. \tag{5}$$

The gate level structure of R4ABE1 is shown in Fig. 1a; the conventional design of MBE [23] consists of four XNOR-2 gates, one XOR-2 gate, one OR-3 gate, one OR-2 gate and one NAND-2 gate. The R4ABE1 design only requires two XOR-2 gates and one AND-2 gate. If the 2-input XOR and XNOR gates are implemented using transmission gates and the other gates are implemented as a complex gate, the transistor count of the MBE for a full CMOS implementation is 34 [23], while the transistor count of the proposed R4ABE1 is only 12, i.e., a reduction of over 64% in terms of circuit complexity is achieved.

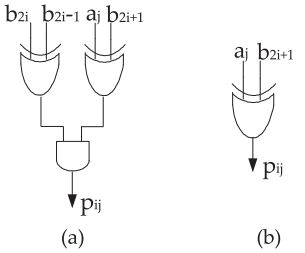

Fig. 1. The gate-level circuit of: (a) R4ABE1 and (b) R4ABE2.

By using a normalized gate delay model of [23] (Table 2), the conventional MBE has a normalized delay of 2.5; therefore, its critical path delay is reduced to 1.7 unit delay in the R4ABE1 design, i.e. an improvement of 32 percent in delay.

2.3 Approximate Radix-4 Booth Encoding Method 2

The truth table of the second approximate radix-4 Booth encoding (R4ABE2) method is shown in Table 3, where ① denotes a '0' entry that has been replaced by a '1'; eight entries in the K-map are modified to simplify the logic of the Booth encoding. The strategy for R4ABE2 is that in addition to having a symmetric truth table at a small error, the number of prime implicants (identified by rectangle) should be as small as possible too. Although the error introduced by R4ABE2 is nearly doubled compared with R4ABE1, the modification is achieved by not only changing a '1' to a '0', but also changing a '0' to a '1'. Thus, the approximate product can be either larger or smaller than the exact product and errors can complement each other in the partial product reduction process. Therefore, when using R4ABE2 in a Booth multiplier, the error may not be larger than for a Booth multiplier with R4ABE1. This is further discussed in Section 3.2.

The modified truth table contains only two prime implicant rectangle; therefore, the output of R4ABE2 is given as follows:

$$pp_{ij} = a_j\overline{b_{2i+1}} + \overline{a_j}b_{2i+1} = b_{2i+1} \oplus a_j. \tag{6}$$

R4ABE2 further reduces the complexity and critical path delay compared with R4ABE1 (Eq. (4)). The error rate is now given by:

$$P_e = 8/32 = 25\%. \tag{7}$$

The gate-level circuit of R4ABE2 is shown in Fig. 1b. R4ABE2 only requires one XOR-2 gate by using transmission gates, so the transistor count of R4ABE2 is 4. R4ABE2 reduces the complexity of the Booth encoder by over 88 percent compared with MBE. By using the normalized gate delay model in Table 2 [23], the critical path delay of R4ABE2 is 1.0; so, it improves the delay by 60 percent compared with MBE.

2.4 Approximate Regular Partial Product Array

For a more regular partial product array (requiring a smaller reduction stage), the Neg term in the $(N/2+1)$th row of the approximate design of a Booth multiplier can be ignored (shown as △ in Fig. 2a). For an *N*-bit radix-4 Booth multiplier when *N* is a power of 2, removing the extra Neg term significantly reduces the critical path, area and power when the 4-2 compressor is used for the partial product accumulation. In the approximate partial product array (Fig. 2b), one reduction stage is saved; this significantly

TABLE 2
Normalized Gate Delay [23]

Normalized Delay	**Gate Type**
1.0	XOR-2, XNOR-2, OR-3
0.7	AND-2, OR-2
0.5	NAND-2

TABLE 3
K-Map of R4ABE2

$a_j a_{j-1}$ \ $b_{2i+1}b_{2i}b_{2i-1}$	000	001	011	010	110	111	101	100
00	0	0	0	0	1	①	1	1
01	0	0	⓪	0	1	①	1	①
11	①	1	1	1	0	0	0	0
10	①	1	①	1	0	0	0	⓪

reduces the complexity and critical path delay. The error rate of the approximate partial product array with the ignored Neg bit is 37.5 percent and its logic function is given by:

$$Neg_{N/2-1} = b_{2N+1}\overline{b_{2N}} + b_{2N+1}\overline{b_{2N-1}} = b_{2N+1}\overline{b_{2N}b_{2N-1}}. \tag{8}$$

2.5 Design of Approximate Booth Multipliers

R4ABE1 and R4ABE2 are applied to a Booth multiplier design. In the approximate Booth multiplier, the proposed approximate Booth encoders, i.e. R4ABE1 and R4ABE2, are used in the first part to generate the inexact partial products. The approximate Booth

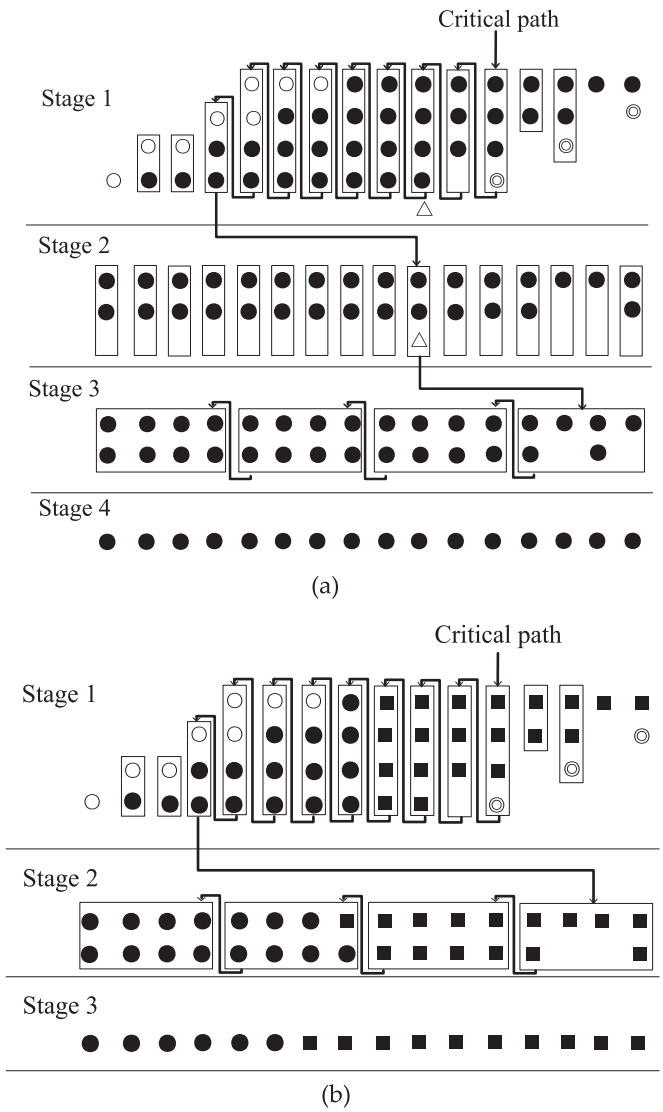

Fig. 2. The 8×8 Booth multiplier using: (a) exact irregular partial product array, (b) approximate regular partial product array by ignoring the Neg term in the 5th partial product row. (The exact partial product term is represented by •, while the approximate partial product term is represented by ■. ◯ and ◎ represent the sign extension bit and the Neg term.)

TABLE 4
Comparison Between Exact and Approximate Booth Encoders to Generate One Partial Product Bit

Booth Encoder	Power (μW)	Delay (ns)	Area (μm^2)	PDP (aJ)	Error Rate
MBE	1.99	0.08	9.84	159.20	0%
R4ABE1	0.93	0.07	3.99	65.10	12.5%
R4ABE2	0.23	0.05	1.60	11.50	25%

encoders can then be used in all or only part of the partial product generation process; therefore, an approximation factor $p\,(p = 1, 2, \ldots, 2N)$ is defined as the number of least significant partial product columns that are generated by the approximate Booth encoders. The approximate partial products are accumulated with the exact 4-2 compressor. The last part uses an exact carry-lookahead adder (CLA) to compute the final product result.

Two types of approximate Booth multipliers are proposed:

1. The first approximate radix-4 Booth multiplier (R4ABM1) uses R4ABE1 (to generate the p least significant partial product columns) and the regular approximate partial product array. The exact MBE is used for generating the $2N$-p most significant partial product columns. The exact 4-2 compressors are used to accumulate both approximate and exact partial products.
2. Similar as R4ABM1, the second approximate radix-4 Booth multiplier (R4ABM2) uses R4ABE2 (to generate the p least significant partial product columns), the regular approximate partial product array and the exact 4-2 compressors.

As the error is controlled by the approximation factor p, a reasonable accuracy can be achieved for different applications; Fig. 2b shows an approximate 8×8 Booth multiplier using either R4ABE1, or R4ABE2 with $p = 8$.

For $p < 7$ of any word length, the delay of an approximate Booth multiplier is only improved by the approximate regular partial product array, since the critical path starts from $p = 7$. For $p > 6$, the delay of the entire multiplier can be further reduced due to the approximate Booth encoder. However, for both cases, the approximate design of a Booth encoder can significantly reduce the area and power consumption.

3 Evaluation and Analysis

3.1 Encoder Design Evaluation

The proposed two approximate Booth encoders, i.e. R4ABE1 and R4ABE2, are described at gate-level in Verilog HDL and verified by Synopsys VCS. Both designs are then synthesized by the Synopsys Design Compiler using the NanGate 45 nm Open Cell Library. In the simulation of each design, a supply voltage of 1.25 V and room temperature are assumed. Standard buffers of a 2X strength are used for both the input driver and the output load; the average power consumption is found using the Synopsys Power Compiler with a back annotated switching activity file generated from the random input vectors.

Table 4 summarizes the power, delay, area, power-delay product (PDP) and error rates of the exact and approximate Booth encoders.

The approximate Booth encoders significantly reduce the PDP; R4ABE2 reduces the PDP by over an order of magnitude compared with MBE, as the best performing Booth encoder. However, the error rate introduced by R4ABE2 is also the largest; R4ABE1 generates half of the error of R4ABE2 and reduces the PDP by up to 59 percent compared with MBE. The error characteristics of approximate multipliers using these approximate Booth encoders are studied next.

3.2 Error Analysis and Evaluation

Although the error rate of each approximate circuit (or module) has been presented, the error characteristics of the entire approximate Booth multiplier must be also considered. For approximate designs, several metrics have been proposed to measure the error of approximate adders and multipliers including the mean error distance (MED), the relative error distance (RED) and the normalization of MED (NMED) [6], [22]. Several error metrics are used to fairly compare different approximate designs of various sizes:

- The NMED is defined as the normalized MED by the maximum output of the accurate design.
- RED is defined as the ED over the absolute accurate result. Mean RED (MRED) and P_{RED} are usually used to evaluate the error distribution of approximate multipliers, where P_{RED} is the probability of obtaining a RED smaller than a specific percentage value that is assumed to be 2 percent throughout the paper.

TABLE 5
Error of Proposed 8-bit and 16-bit Approximate Multipliers at Different Approximation Factors (ER: Error Rate)

Approximate Design	p	8-bit Approximate Booth Multipliers				16-bit Approximate Booth Multipliers			
		NMED (10^{-2})	MRED (10^{-2})	P_{RED} (%)	ER (%)	NMED (10^{-5})	MRED (10^{-2})	P_{RED} (%)	ER (%)
R4ABM1	2	0.073	0.051	83.72	53.12	0.2542	0.0264	99.80	50.00
	4	0.082	0.092	80.05	62.89	0.2543	0.0529	99.79	60.42
	6	0.137	0.641	66.45	71.92	0.2545	1.37	99.78	69.40
	8	0.427	5.256	40.30	75.89	0.2547	15.8	99.71	76.74
	10	1.269	19.917	30.23	78.15	0.2557	133	99.44	82.46
	12	3.369	55.125	28.44	78.69	0.3082	916	98.51	85.72
	14	7.022	104.73	28.14	78.81	0.9303	5326	95.89	93.75
	16	-	-	-	-	3.1026	$1.76^{*}10^{4}$	90.61	93.98
	18	-	-	-	-	12.295	$1.37^{*}10^{5}$	81.64	94.04
	20	-	-	-	-	46.134	$4.61^{*}10^{5}$	63.59	94.05
	24	-	-	-	-	580.47	$3.42^{*}10^{6}$	32.08	94.05
	28	-	-	-	-	5558.2	$1.12^{*}10^{7}$	12.15	94.06
	32	-	-	-	-	11115	$2.53^{*}10^{7}$	12.15	94.06
R4ABM2	2	0.073	0.048	83.81	60.93	0.2543	0.344	99.79	73.95
	4	0.076	0.134	80.29	78.75	0.2543	0.689	99.79	77.34
	6	0.104	2.142	70.20	88.87	0.2543	9.61	99.76	88.27
	8	0.409	24.120	37.16	94.16	0.2545	125	99.66	94.03
	10	1.400	124.692	16.01	96.12	0.2550	1165	99.23	96.99
	12	4.089	469.695	10.08	96.57	0.2671	8837	97.81	98.49
	14	10.138	1291.11	8.56	96.66	0.6243	$5.78^{*}10^{4}$	93.44	99.61
	16	-	-	-	-	3.0173	$3.39^{*}10^{5}$	86.07	99.86
	18	-	-	-	-	11.241	$1.58^{*}10^{6}$	77.12	99.96
	20	-	-	-	-	40.973	$6.83^{*}10^{6}$	57.11	99.99
	24	-	-	-	-	542.48	$1.03^{*}10^{8}$	34.80	99.99
	28	-	-	-	-	7270.3	$1.07^{*}10^{9}$	0.217	99.99
	32	-	-	-	-	17879	$2.52^{*}10^{9}$	0.123	99.99

The error metrics of the proposed approximate multipliers are provided in Table 5 for 8-bit and 16-bit designs; Table 5 shows that with an increase of p, the accuracy of both approximate multipliers decreases. MRED increases with an increase of p and the MRED of R4ABM2 is generally larger than that of R4ABM1. P_{RED} (the probability of getting a RED smaller than 2 percent) of both the 8-bit R4ABM1 and R4ABM2 decrease rapidly when $4 < p < 12$. P_{RED} of the 16-bit designs is larger than 90 percent when $p < 16$. The error rates for 8-bit and 16-bit approximate Booth multipliers are also shown in Table 5; the error rate is increased by increasing p. The error rate of R4ABM1 is smaller than for R4ABM2 at the same value of p because the error rate of R4ABE-1 (12.5 percent) is significantly smaller than for R4ABE2 (25 percent). However, for error analysis, the most important error metric is in this case the difference between the exact and the approximate values, so related to the error distance (i.e. NMED).

Consider next the NMED; although the error rate of R4ABM1 is smaller than for R4AMB2, the NMED of R4ABM1 is larger than for R4ABM2 when $2 < p < 10$ for 8-bit and when $6 < p < 28$ for 16-bit.

TABLE 6
Designs (45 nm Technology) of 8-bit and 16-bit Approximate Multipliers at Different Approximation Factors

Booth Multiplier Designs	p	8-bit Booth Multipliers				16-bit Booth Multipliers			
		Power (μW)	Delay (ns)	Area (μm^2)	PDP (pJ)	Power (μW)	Delay (ns)	Area (μm^2)	PDP (pJ)
R4EBM	0	188.0	0.70	788.4	0.131	634.0	1.02	2645	0.646
R4ABM1	2	167.5	0.62	671.9	0.103	609.9	0.96	2426	0.585
	4	160.6	0.60	648.8	0.096	600.9	0.96	2425	0.576
	6	148.3	0.60	625.1	0.088	589.4	0.96	2345	0.565
	8	138.0	0.58	581.7	0.080	557.5	0.95	2278	0.529
	10	109.7	0.58	515.5	0.063	547.3	0.95	2242	0.519
	12	104.3	0.57	500.6	0.059	535.0	0.95	2209	0.508
	14	101.3	0.57	495.0	0.057	516.6	0.95	2169	0.490
	16	-	-	-	-	479.9	0.94	2066	0.451
	18	-	-	-	-	448.4	0.94	2002	0.421
	20	-	-	-	-	408.1	0.93	1875	0.379
	24	-	-	-	-	371.8	0.93	1780	0.345
	28	-	-	-	-	351.2	0.92	1756	0.323
	32	-	-	-	-	342.2	0.92	1733	0.314
R4ABM2	2	164.8	0.62	667.9	0.102	601.3	0.96	2403	0.577
	4	162.6	0.60	649.3	0.097	600.2	0.96	2412	0.576
	6	139.1	0.58	575.0	0.080	571.8	0.95	2314	0.543
	8	127.4	0.58	538.6	0.073	563.2	0.95	2276	0.535
	10	111.7	0.57	480.4	0.063	551.8	0.94	2217	0.518
	12	93.3	0.56	423.2	0.052	515.4	0.94	2077	0.484
	14	88.4	0.55	405.3	0.048	479.7	0.92	2004	0.441
	16	-	-	-	-	448.7	0.92	1875	0.412
	18	-	-	-	-	437.4	0.89	1788	0.389
	20	-	-	-	-	394.6	0.89	1642	0.351
	24	-	-	-	-	358.3	0.89	1489	0.318
	28	-	-	-	-	343.1	0.89	1423	0.305
	32	-	-	-	-	339.0	0.89	1408	0.301

This occurs because there are only positive errors in R4ABE1, while both positive and negative errors are introduced in R4ABE2. Hence, they can complement each other, such that smaller errors are generated overall when using R4ABE2 in an approximate multiplier. This is further demonstrated with the peak signal-noise ratio (PSNR) results for the image application in Section 4. An 8-bit approximate Booth multiplier design with a value of p not larger than 8 is a good choice for an error-tolerant application; for a 16-bit approximate Booth multiplier design, p should not be larger than 20 for a reasonable tradeoff between accuracy and low power.

3.3 Approximate Multiplier Hardware Evaluation

Evaluation by simulation is pursued for the proposed approximate multipliers under the same conditions as in Section 3.1. The critical path delay, area, power consumption and PDP are reported in Tables 6 and 7 for 8-bit, 16-bit and 32-bit exact radix-4 Booth multipliers (R4EBM), R4ABM1 and R4ABM2 at different values of the approximation factor. With an increase of the approximation factor p, the power consumption and PDP decrease significantly. The PDPs of both R4ABM1 and R4ABM2 are significantly lower than for R4EBM; the critical path delay is also reduced, leading to an overall better performance. R4ABM2 is better than R4ABM1 in terms of power, delay, area and PDP; this is consistent with the comparison results for R4ABE1 and R4ABE2. Not surprisingly, the use of a simpler approximate Booth encoder in R4ABM2 leads to a high-performance and power-efficient approximate design.

3.4 Comparison of Approximate Booth Multipliers

The proposed designs at large p values have large NMEDs, but small PDPs; moreover, the proposed 16-bit R4ABM1 and R4AMB2 with $p < 20$ show a good tradeoff between PDP and NMED. Therefore, both proposed 16-bit designs with $p = 12$, 14, 16 and 18 are compared with previous approximate 16-bit Booth multipliers that have been proposed in [9] (R4BM04), [12] (R4ABM11), [15] (R4ABM12), and [22] (R8ABM1, R8ABM2-C9 and R8ABM2-C15). R8ABM1, R8ABM2-C9 and R8ABM2-C15 are all radix-8 approximate Booth multipliers with no truncation, 9-bit truncation and 15-bit truncation, respectively. R8ABM2-C9 and R8ABM2-C15 are also designed with the compensation circuits. Also in these cases, all designs are described in Verilog as combinational multipliers and synthesized by the Synopsys Design Compiler using the NanGate 45 nm Open Cell Library.

TABLE 7
Designs (45 nm Technology) of 32-bit Approximate Multipliers at Different Approximation Factors

32-bit Booth Multiplier Designs	p	R4ABM1				R4ABM2			
		Power (μW)	Delay (ns)	Area (μm^2)	PDP (pJ)	Power (μW)	Delay (ns)	Area (μm^2)	PDP (pJ)
R4EBM	0	2489.8	1.59	10222.1	3.958	2489.8	1.59	10222.1	3.958
R4ABM	8	2386.2	1.54	9598.1	3.674	2449.1	1.52	9572.8	3.722
	16	2309.7	1.52	9289.7	3.510	2292.6	1.48	9258.3	3.393
	24	2241.8	1.52	8901.1	3.407	2086.3	1.48	8540.1	3.087
	28	2133.5	1.51	8557.4	3.221	2034.2	1.48	8103.9	3.010
	32	2050.0	1.51	8200.5	3.095	1870.9	1.46	7684.5	2.731
	36	2008.8	1.49	7960.3	2.993	1728.7	1.46	7145.5	2.523
	44	1845.2	1.49	7452.5	2.749	1537.0	1.45	6413.5	2.228
	52	1774.4	1.47	7162.3	2.608	1402.8	1.45	5973.6	2.034
	60	1730.4	1.47	7007.7	2.543	1343.3	1.45	5768.7	1.947

The power consumption, delay, area, PDP, NMED and error rate are reported in Table 8. The proposed R4ABM1 and R4ABM2 can reduce the power consumption significantly (in a range of 25∼31 percent) compared with the accurate design, i.e., R4EBM. The area of the proposed designs is also much smaller (up to 32 percent) than that of their exact counterpart. The delay of the proposed approximate designs can be improved by up to 12 percent when $p = 18$. Therefore, the PDPs of both R4ABM1 and R4ABM2 are significantly smaller than R4EBM (up to 39 percent).

Although the approximate radix-8 designs show good performance in terms of power consumption and PDP, their delay is generally higher than the proposed designs. The power and PDP results of R8ABM2-C15 are the best among all approximate Booth multipliers; however, it also has a large NMED and error rate. The most accurate approximate radix-8 design is R8ABM1; and its accuracy cannot be further increased. This is also the drawback of the fixed-width Booth multipliers, i.e., R4ABM04, R4ABM11, and R4ABM12, where accuracy is not adjustable and their NMED and ER are significantly larger than the proposed designs when $p < 16$. However, the accuracy of our proposed designs can be changed along with PDP to meet various application requirements.

TABLE 8
Comparative Performance of Approximate Booth Multipliers

16-bit Booth Multipliers	Power (μW)	Delay (ns)	Area (μm^2)	PDP (pJ)	NMED (10^{-5})	ER (%)
R4EBM [23]	634.0	1.02	2645	0.646	0	0
R4ABM04 [9]	427.3	0.95	1939	0.406	5.31	99.99
R4ABM11 [12]	404.4	0.94	1859	0.380	2.18	99.98
R4ABM12 [15]	394.6	0.95	1808	0.374	2.26	99.98
R8ABM1 [22]	376.7	1.23	1516	0.463	1.92	44.06
R8ABM2-C9 [22]	332.6	1.22	1332	0.406	4.43	99.74
R8ABM2-C15 [22]	217.3	1.18	912	0.256	5.73	99.99
R4ABM1 ($p = 12$)	535.0	0.95	2209	0.508	0.31	85.72
R4ABM1 ($p = 14$)	516.6	0.95	2169	0.490	0.93	93.75
R4ABM1 ($p = 16$)	479.9	0.94	2066	0.451	3.10	93.98
R4ABM1 ($p = 18$)	448.4	0.94	2002	0.421	12.29	94.04
R4ABM2 ($p = 12$)	515.4	0.94	2077	0.484	0.27	98.49
R4ABM2 ($p = 14$)	479.7	0.92	2004	0.441	0.62	99.61
R4ABM2 ($p = 16$)	448.7	0.92	1875	0.412	3.02	99.86
R4ABM2 ($p = 18$)	437.4	0.89	1788	0.389	11.24	99.96

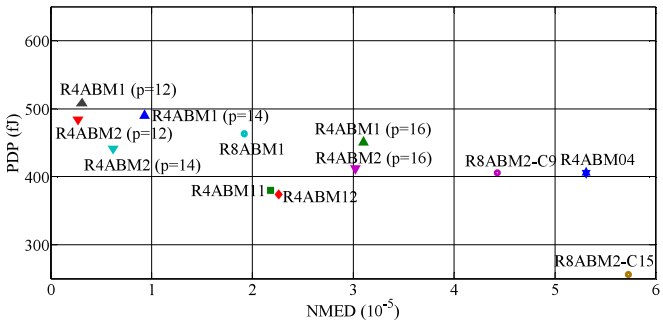

Fig. 3. PDP versus NMED for approximate Booth multipliers.

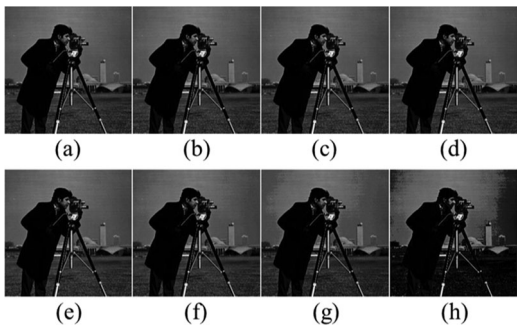

Fig. 4. Image multiplication using R4ABM1 with different p: (a) $p = 0$, (b) $p = 2$, (c) $p = 4$, (d) $p = 6$, (e) $p = 8$, (f) $p = 10$, (g) $p = 12$, and (h) $p = 14$.

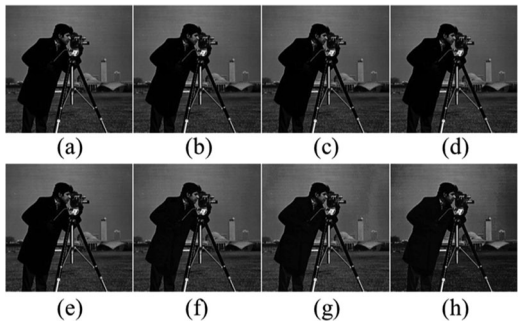

Fig. 5. Image multiplication using R4ABM2 with different p: (a) $p = 0$, (b) $p = 2$, (c) $p = 4$, (d) $p = 6$, (e) $p = 8$, (f) $p = 10$, (g) $p = 12$, and (h) $p = 14$.

TABLE 9
PSNR of Processed Images Using Approximate Booth Multipliers at Different Approximation Factors

Approximation Factor (p)	R4ABM1 (dB)	R4ABM2 (dB)
2	57.043	57.043
4	56.733	56.885
6	54.485	55.022
8	52.514	53.598
10	37.818	39.646
12	30.340	33.918
14	17.576	24.643

Among all approximate Booth multipliers, only the proposed designs with $p = 12$ and $p = 14$ have NMEDs that are smaller than 10^{-5}; and their accuracy can be further improved by applying a smaller p. This is achieved with no error compensation or recovery. The choice of p is dependent on different specific applications that are error-tolerant; for applications that can tolerant more error, the proposed approximate Booth multipliers with larger p can be used to further reduce the power and delay.

The approximate Booth multipliers are further compared in terms of both PDP and NMED. Fig. 3 shows that the proposed R4ABM2 ($p = 14$) is the most efficient design when considering both PDP and NMED. Both R4ABM1 and R4ABM2 with $p = 12$ show better NMED but larger PDP, while R4ABM11 and R4ABM12 have better PDP but larger NMED. Although an approximate radix-8 design, especially R8ABM2-C15, has a small PDP, its large NMED makes it less attractive when considering both power consumption and accuracy. These results confirm that both proposed designs with $p = 12$ and $p = 14$ are more accurate than previous approximate Booth multipliers with moderate power consumption.

4 Case Study: Image Processing

In this section, 8-bit R4ABM1 and R4ABM2 with various p are applied to image processing; two images are multiplied on a pixel-by-pixel basis to blend them into a single output image. As Booth multipliers perform signed multiplication, the pixel values have been shifted from 0∼255 to −128∼127. The approximate designs are modeled in C that is called by Matlab for image processing. The peak signal-to-noise ratio (PSNR) is used to assess the quality of the output image.

As shown in Figs. 4 and 5, the quality of the processed image deteriorates with an increase of p. The output image is still viable when p is smaller than 12, thus confirming the error analysis presented previously. The PSNRs of all subfigures shown in Figs. 4 and 5 are provided in Table 9; the PSNR drops significantly when p is larger than 10. The PSNR of the output images processed by R4ABM2 is larger compared with R4ABM1; this finding is again consistent with the previously presented error analysis in Section 3.2. Both R4ABM1 and R4ABM2 with $p < 10$ can be used in image processing applications with no significant loss of quality.

5 Conclusion

Designs of approximate radix-4 Booth multipliers have been studied in this paper. Two approximate Booth encoders have been proposed to reduce the complexity of MBE by introducing incorrect terms in the truth table, resulting in a reduction of the PDP by up to 59 percent. An approximation factor has been proposed to indicate the inexact bit width of the Booth multiplier and facilitate the error analysis of proposed designs. Based on the two proposed approximate Booth encoders, i.e., R4ABE1 and R4ABE2, two approximate Booth multipliers have been designed with different approximation factors. An approximate Wallace tree structure has been designed for use in the partial product accumulation array. The power, delay and PDP of the proposed designs have been evaluated at 45nm CMOS technology.

The error characteristics of the proposed designs have been studied at different values of the approximation factor (p) to establish a suitable balance between various figures of merit such as PDP and NMED. The proposed designs have been compared with previous approximate Booth multipliers; results presented in this paper have shown that the proposed R4ABM2 with $p = 14$ is the most efficient design when considering both PDP and NMED. Both proposed designs with $p = 12$ and $p = 14$ are more accurate than previous approximate Booth multipliers with moderate power consumption. The proposed designs have also been applied to image processing, resulting in a very small loss of accuracy.

Acknowledgments

This work is supported by grants from the National Natural Science Foundation China (61401197) and Natural Science Foundation of Jiangsu Province (BK20151477).

REFERENCES

[1] S. Venkataramani, S. T. Chakradhar, and K. Roy, "Computing approximately, and efficiently," in *Proc. Design Automation Test Europe Conf. Exhib.*, 2015, pp. 748–751.

[2] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *Proc. 18th IEEE Eur. Test Symp.*, 2013, pp. 1–6.

[3] H. Mahdiani, A. Ahmadi, S. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[4] V. Gupta, D. Mohapatra, S. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise adders for low-power approximate computing," in *Proc. Int. Symp. Low Power Electron. Des.*, 2011, pp. 1–3.

[5] W. Liu, L. Chen, C. Wang, M. O'Neill, and F. Lombardi, "Design and analysis of floating-point adders," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 308–314, Jan. 2016.

[6] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Trans. Comput.*, vol. 63, no. 9, pp. 1760–1771, Sep. 2013.

[7] A. Booth, "A signed binary multiplication technique," *Quarterly J. Mech. Appl. Math.*, vol. 4, pp. 236–240, Jun. 1951.

[8] O. MacSorley, "High-speed arithmetic in binary computers," *Proc. IRE*, vol. 49, pp. 67–91, 1961.

[9] K.-J. Cho, K.-C. Lee, J.-G. Chung, and K. K. Parhi, "Design of low error fixed-width modified Booth multiplier," *IEEE Trans. VLSI Syst.*, vol. 12, no. 5, pp. 522–531, May 2004.

[10] M. J. Schulte and E. E. Swartzlander Jr., "Truncated multiplication with correction constant," in *Proc. Workshop VLSI Signal Process. VI*, 1993, pp. 388–396.

[11] E. J. King and E. E. Swartzlander Jr., "Data dependent truncated scheme for parallel multiplication," in *Proc. 31st Asilomar Conf. Signals, Circuits Syst.*, 1998, pp. 1178–1182.

[12] J.-P. Wang, S.-R. Kuang, and S.-C. Liang, "High-accuracy fixed-width modified Booth multipliers for lossy applications," *IEEE Trans. VLSI Syst.*, vol. 19, no. 1, pp. 52–60, Jan. 2011.

[13] 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: Express Briefs*, vol. 58, no. 4, pp. 215–219, Apr. 2011.

[14] Y.-H. Chen, C.-Y. Li, and T.-Y. Chang, "Area-effective and power efficient fixed-width Booth multipliers using generalized probabilistic estimation bias," *IEEE J. Emerging Selected Topics Circuits Syst.*, vol. 1, no. 3, pp. 277–288, Sep. 2011.

[15] 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*, vol. 59, no. 3, pp. 594–603, Mar. 2012.

[16] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power with an under designed multiplier architecture," in *Proc. 24th Int. Conf. VLSI Des.*, 2011, pp. 346–351.

[17] C.-H. Lin and C. Lin, "High accuracy approximate multiplier with error correction," in *Proc. IEEE Int. Conf. Comput. Des.*, 2013, pp. 33–38.

[18] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 984–994, Apr. 2015.

[19] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in *Proc. Design Autom. Test Europe*, 2014, pp. 95–98.

[20] G. Zervakis, S. Xydis, K. Tsoumanis, D. Soudris, and K. Pekmestzi, "Hybrid approximate multiplier architectures for improved power-accuracy trade-offs," in *Proc. IEEE Int. Symp. Low Power Electron. Des.*, 2015, pp. 79–84.

[21] S. Hashemi, R. Bahar, and S. Reda, "DRUM: A dynamic range unbiased multiplier for approximate applications," in *Proc. IEEE/ACM Int. Conf. Comput. Des.*, 2015, pp. 418–425.

[22] H. Jiang, J. Han, F. Qiao, and F. Lombardi, "Approximate radix-8 Booth multipliers for low-power and high performance operation," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2638–2644, Aug. 2016.

[23] W. Yeh and C. Jen, "High-speed Booth encoded parallel multiplier design," *IEEE Trans. Comput.*, vol. 49, no. 7, pp. 692–701, Jul. 2000.

[24] S. Kuang, J. Wang, and C. Guo, "Modified Booth multiplier with a regular partial product array," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 56, no. 5, pp. 404–408, May 2009.

[25] L. Qiang, C. Wang, W. Liu, F. Lombardi, and J. Han, "Design and evaluation of an approximate Wallace-Booth multiplier," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2016, pp. 1974–1977.