Design and Analysis of Approximate 4-2 Compressors for High-Accuracy Multipliers

Tianqi Kong and Shuguo Li, *Member, IEEE*

***Abstract*— Approximate multipliers are applicable in error-resilient applications with relaxed precision constraints, including image processing, multimedia, and data recognition. Such multipliers that sacrifice some accuracy can gain a corresponding increase in electrical performance. This article presents an analysis of the architectures of previously proposed compressors to investigate their performance and accuracy. In this article, we propose five high-accuracy approximate 4-2 compressors with better delay, area, power, and better performance–accuracy tradeoff. Pro1–Pro4 rely on the critical path optimization, while Pro5 derives from the modified sorting technique. This article implements 8 × 8 and 16 × 16 multipliers by employing the proposed approximate compressors in TSMC 28 nm. The experimental results indicate that our designs have about 18% delay, 43%–52% area-delay product (ADP) reduction compared to the exact multiplier, and 20%–55% ADP optimization compared to compressors with the same accuracy. This article further verifies the efficacy of the proposed compressors through image blending and matrix multiplication applications.**

***Index Terms*— 4-2 Compressors, approximate arithmetic, approximate compressors, approximate multipliers, critical path.**

I. Introduction

AS AN extensively used arithmetic unit, multiplier is widely used in microprocessors, digital signal processing (DSP) applications, and so on. In order to speed up the processing operation and optimize the performance of system, some high-performance approximate multipliers with reduced area and power consumption have emerged [1]–[3]. Approximate multipliers can be used in error-resilient [4] applications with relaxed precision constraints, such as multimedia, data recognition, image processing, and convolutional neural networks (CNNs) [36]–[41]. Artificial intelligence as a cutting-edge research field brings various algorithms with large amount of computation. Due to the computationally expensive algorithms, many works have focused on approximate computation to achieve better performance [42]–[50]. Such multipliers that sacrifice some accuracy can gain a corresponding increase in electrical performance. Several studies have focused on high-performance approximate multipliers with low error rate (ER).

The implementation of multiplier included three steps: 1) partial products generation; 2) partial products array compression; and 3) addition. Accordingly, the approaches of approximate multipliers could also be classified by these three aspects: 1) approximate generation; 2) approximate partial products compression; and 3) approximate addition. Other approaches, such as logarithmic multipliers [5], approximate redundant binary multipliers [6], [7], and runtime configuration [8], have also been researched and applied.

For multipliers with inaccurate partial products, the approximate approaches commonly rely on logic simplification, which simplifies the generation logic of products through the Karnaugh map (K-map) or other methods to reduce the circuit complexity [9]–[12]. Whether or not these approaches use the Booth encoding algorithm [13], [14], they all achieve better performance by reducing some elements in partial product array with simpler partial products generators [34]. Naturally, the reduced elements and simpler generators will introduce corresponding errors, and some included error compensation circuit to increase the accuracy is proposed in [15]–[17].

Approximate multipliers proposed in [18]–[26] and [35] are based on approximate compressors. After the partial products generation, the products array is usually compressed into two rows and then added to the final result. As a bottleneck in the multiplication, the partial products accumulation has been extensively studied. The approaches of approximate compression commonly use approximate compressors with reduced complexity to substitute a part of exact compressors. Similarly, while saving delay, area, and power, these approaches also introduce some errors. Take the 4-2 compressor as an example: the compressor proposed in [18] introduces four errors and compressor in [20] introduces only a single error, while the electrical performance in [18] is better than [20]. It shows that there is an electrical performance–accuracy tradeoff.

There are also some approximate multipliers constructed by approximate addition. In [27]–[29], the addition approximate approaches are proposed with carry prediction and adder partition. The adder in [29] is partitioned into several non-overlapped summation blocks and the logic is simplified to reduce the circuit complexity.

In this article, we aim at the research of approximate 4-2 compressors. We carried out sufficient investigation and survey of previous high-accuracy 4-2 compressors, analyzed their critical paths, and presented a comparison. Keeping the

Manuscript received April 20, 2021; revised July 8, 2021; accepted August 1, 2021. Date of publication August 26, 2021; date of current version October 6, 2021. This work was supported by the National Natural Science Foundation of China under Grant 61974083. *(Corresponding author: Shuguo Li.)*

The authors are with the Institute of Microelectronics, Tsinghua University, Beijing 100084, China (e-mail: ktq18@mails.tsinghua.edu.cn; lisg@tsinghua.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TVLSI.2021.3104145.

Digital Object Identifier 10.1109/TVLSI.2021.3104145

1063-8210 © 2021 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: VIT University. Downloaded on January 25,2023 at 08:37:47 UTC from IEEE Xplore. Restrictions apply.

TABLE I
NORMALIZED GATE DELAY

Gate Type	The Normalized Delay	Description of Gate
XOR-2	1.0	the 2-input XOR gate
NXOR-2	1.0	the 2-input NXOR gate
OR-2	0.7	the 2-input OR gate
NOR-2	0.4	the 2-input NOR gate
AND-2	0.6	the 2-input AND gate
NAND-2	0.3	the 2-input NAND gate
INV	0.2	the inverter
MUX-2	1	the 2-to-1 Multiplexer
Full-Adder	1	1-Bit Full Adder
AO222	1	the compound And-Or gate
AO22	0.8	the compound And-Or gate
AOI22	0.5	the compound And-Or-inverter gate
OAI22	0.5	the compound And-Or-inverter gate

high-accuracy characteristic, our work recombines the inputs of 4-2 compressor and designs the logic architecture based on the analysis of critical paths. To optimize the sorting technique, this article sorts the inputs of compressor with opposite idea to shorten the logic path. The 4-2 compressors we proposed with at most two errors introduced can achieve better performance–accuracy tradeoff due to its short critical paths.

The primary contributions of this article are listed in the following.

1) This article investigates the sufficient survey of previous state-of-the-art approximate 4-2 compressors and extensively analyzes the respective critical paths in their architectures. It also presents a comprehensive comparison in terms of performance and accuracy.
2) This article proposes five architectures (Pro1–Pro5) of high-accuracy approximate 4-2 compressors with shorter critical paths by decomposition and recombination methods that can build multipliers with a better tradeoff.
3) This article modifies the sorting technique to assist in the design of approximate 4-2 compressors. The compressor "Pro5" we proposed is designed by this technical routine.
4) Approximate multipliers with three varied schemes both symmetric and asymmetric are designed, and experiments indicate that the proposed compressors are suitable for high-accuracy multipliers.
5) This article presents an area-delay product (ADP) versus accuracy tradeoff graph that can be helpful for the selection of the approximate compressor that is best suited for a specific application.

This article is divided into four major sections as follows. Section II shows the previous developments of approximate 4-2 compressors and gives our analysis of them. Section III proposes our architectures of compressors and reports the error characteristics. The comparison and analysis of compressors, VLSI implementations, image processing, and matrix multiplication applications are shown in Section IV.

II. PREVIOUS 4-2 COMPRESSORS

In this article, we use the normalized gate delay model to analyze the architectures of approximate 4-2 compressors. According to the TSMC standard cell library data sheet [51], this article sets the delay of two-input XOR gate as one unit delay to simplify the calculation. Based on this, the normalized delays of all gates mentioned in this work are calculated in Table I.

A. High-Accuracy Approximate Compressors

This kind of compressor is much needed in the high-accuracy multipliers. The experimental results in Section IV indicated that the compressor with less than four errors can deliver better accuracy to multipliers. Therefore, in this article, we define this kind of compressors as high-accuracy compressor. Compared with the exact 4-2 compressor $(\text{cout}, \text{carry}, \text{sum}) = a + b + c + d + \text{cin}$, the approximate 4-2 compressor usually omits the "cin" and "cout" pins as $(\text{carry}, \text{sum}) = a+b+c+d$. This kind of omission can break the carry propagation between compressors, so it speeds up the products accumulation. The compressors proposed in [20] (simplified as "Yang" in the following) only introduce four errors at most and the most accurate of them ("Yang1") only introduces a single error as shows in the truth table (Table II). The analysis of critical paths in "Yang1–3" compressors is shown in Fig. 1(a)–(c). The red lines in these architectures mark the critical path of compressor circuit and the calculations of normalized gate delays are shown in the following:

$$\begin{aligned} &\text{Yang1} : 0.7 + 0.3 + 1 + 0.3 = 2.3 \\ &\text{Yang2} : 1 + 1 + 0.7 = 2.7 \\ &\text{Yang3} : 1 + 1^{>} = 2^{>} \end{aligned} \tag{1}$$

where ">" indicates that the value is slightly larger than its number.

The 4-2 compressor proposed in [21] (referred to as "Lin" in the following) with two errors has a shorter critical path than "Yang1–3." "Lin" has an XOR-2, an inverter, and an MUX-2 on the critical path, as shown in Fig. 1(d). The normalized critical path delay almost is equal to two XOR-2 gates. In [22] (referred as "Ha"), compressor with four errors has two XOR-2 gates on its critical path, as shown in Fig. 1(e). These two compressors have the same characteristic, that is, the calculated results are always less than exact results

$$\begin{aligned} &\text{Lin} : 1 + 0.2 + 1 + 0.8 = 2 \\ &\text{Ha} : 1 + 1 = 2. \end{aligned} \tag{2}$$

The work [23] (referred as "Strollo") uses the "stacking circuit" to propose architectures of compressor. As shown in Fig. 1(f)–(h), there are a common compound gate "full adder" in these three architectures. However, the front-end circuit complexity of compressors in [23] increases with accuracy. The truth tables of "Strollo1-3" are shown in Table II and the calculation of critical paths is shown in the following equation:

$$\begin{aligned} &\text{Strollo1} : 0.7 + 1^{>} = 1.7^{>} \\ &\text{Strollo2} : 0.6 + 0.7 + 1^{>} = 2.3^{>} \\ &\text{Strollo3} : 0.6 + 0.6 + 0.7 + 1^{>} = 2.9^{>}. \end{aligned} \tag{3}$$

B. Low-Accuracy Approximate Compressors

This section introduces approximate 4-2 compressors with more than four errors. With relatively high ER, these low-accuracy compressors have significant performance advantages. The architectures of them are presented in Fig. 2, and the behavior of these simple architectures is reported in Table II.

Authorized licensed use limited to: VIT University. Downloaded on January 25,2023 at 08:37:47 UTC from IEEE Xplore. Restrictions apply.

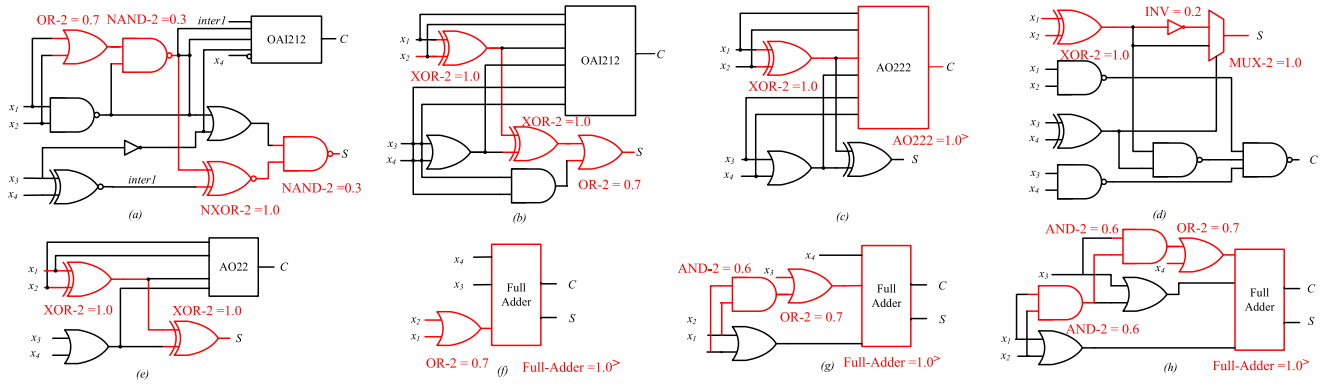

Fig. 1. Analysis of high-accuracy approximate 4-2 compressor. (a)–(c) Yang *et al.* [20]. (d) Lin and Lin [21]. (e) Ha and Lee [22]. (f)–(h) Strollo *et al.* [23].

TABLE II
TRUTH TABLE OF APPROXIMATE 4-2 COMPRESSORS

$x_4...x_1$	[A]		[B]		Yang3		[C]		[D]		Strollo2		Momeni		Venka		Akbar1		Akbar2		Sabetz		Ahma		Ranjbar1		Ranjbar2		Ranjbar3	
	CS	E	CS	E	CS	E	CS	E	CS	E	CS	E	CS	E	CS	E	CS	E	CS	E	CS	E	CS	E	CS	E	CS	E	CS	E
0000	00		00		00		00		00		00		01	+1	00		00		00		01	+1	00		00		00		00	
0001	01		01		01		01		01		01		01		01		01		01		01		01		10	+1	01		10	+1
0010	01		01		01		01		01		01		01		01		01		01		01		01		10	+1	10	+1	01	
0011	10		10		10		10		10		10		01	-1	10		00	-2	10		01	-1	01	-1	10		10		10	
0100	01		01		01		01		01		01		01		01		01		01		01		01		01		10	+1	10	+1
0101	10		10		10		10		10		10		10		01	-1	01	-1	11	-1	11	+1	11	+1	10		10		10	
0110	10		10		10		10		10		10		10		01	-1	01	-1	11	-1	01	-1	11	+1	10		10		10	
0111	11		11		11		11		11		10	-1	11		11		01	-2	11		11		11		11		11		11	
1000	01		01		01		01		01		01		01		01		11	+2	01		01		01		01		01		01	
1001	10		10		10		10		10		10		10		01	-1	11	+1	11	+1	11	+1	11	+1	10		01	-1	10	
1010	10		10		10		10		10		10		10		01	-1	11	+1	11	+1	01	-1	11	+1	10		10		01	-1
1011	11		11		11		11		11		11		11		11		11		11		11		11		11		11		11	
1100	10		11	+1	11	+1	10		01	-1	10		01	-1	10		10		10		11	+1	01	-1	01	-1	10		10	
1101	11		11		10	-1	11		10	-1	11		11		11		11		11		11		11		11		11		11	
1110	11		11		10	-1	11		10	-1	11		11		11		11		11		11		11		11		11		11	
1111	11	-1	11	-1	11	-1	10	-2	11	-1	11	-1	11	-1	11	-1	10	-2	10	-2	11	-1	11	-1	11	-1	11	-1	11	-1

[A]:Yang1,Strollo3 and Proposed1,5; [B]:Yang2,Proposed3,Proposed4; [C]:Lin and Proposed2; [D]:Ha and Strollo1

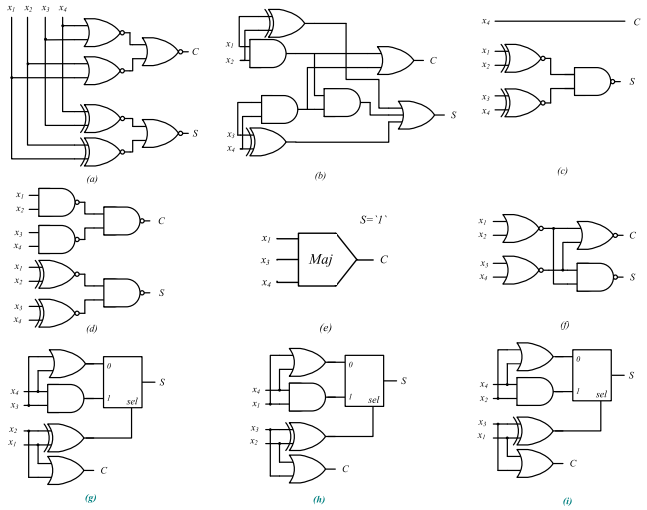

Fig. 2. Low-accuracy approximate 4-2 compressor. (a) Momeni *et al.* [18]. (b) Venkatachalam and Ko [19]. (c) and (d) Akbari *et al.* [24]. (e) Sabetzadeh *et al.* [25]. (f) Ahmadinejad *et al.* [26]. (g)–(i) Ranjbar *et al.* [35].

III. PROPOSED APPROXIMATE MULTIPLIERS

In this section, we propose five approximate 4-2 compressors and apply them to approximate multipliers. Based on the truth table, the compressors in Sections III-A–III-D are all designed to shorten the critical path. Moreover, we also optimize the sorting technique with reversed logic gates to construct a high-accuracy 4-2 compressor with better performance.

The novelty of the proposed compressors "Pro1–Pro5" is as follows.

1) The first four architectures "Pro1–Pro4" rely on the truth table with at most two errors introduced are constructed by decomposition-recombination methods. These compressors we proposed have shorter critical paths than previous compressors with the same accuracy.
2) We modify the sorting technique to construct a compressor with shorter critical path, and "Pro5" we proposed is designed by this technical routine.
3) All of the compressors we proposed not only have shorter critical paths but also can obtain high accuracy with at most two errors introduced. Therefore, our design can obtain a better performance–accuracy tradeoff.

A. Optimized Critical Path: "Pro1"

Relying on the truth table of the most accurate compressor (Table II, A), there are nine logical terms that can output Sum = 1 and 11 logical terms output Carry = 1. For instance, the logical term $(\bar{a}\,\bar{b}\,\bar{c}\,d)$ means that if input "0001," the output of compressor will be $S = 1$ and $C = 0$ (we use S and C denote Sum and Carry in the following, respectively).

Authorized licensed use limited to: VIT University. Downloaded on January 25,2023 at 08:37:47 UTC from IEEE Xplore. Restrictions apply.

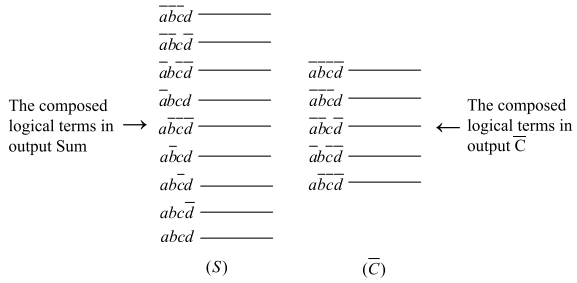

Fig. 3. Decomposition of output S and $\overline{C}$.

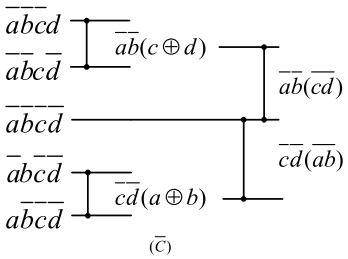

Fig. 4. Recombination of logic terms of $\overline{C}$.

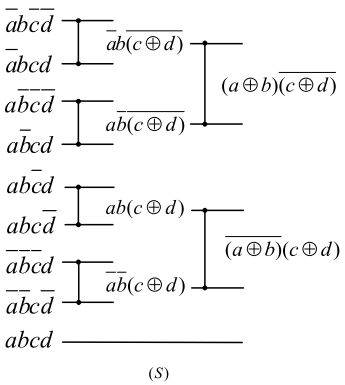

Fig. 5. Recombination of decomposed S in "Pro1."

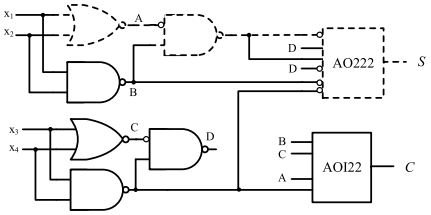

Fig. 6. Circuit architecture of "Pro1."

Therefore, the output logic expressions of $S = 1$ and $C = 1$ can be decomposed to several logical terms. Because the number of terms in $\overline{C} = 0$ is much less than $C = 1$, we decompose $\overline{C}$ equivalently. The logical terms decomposition of output 'S' and '$\overline{C}$' is presented in Fig. 3.

The recombination of decomposed $\overline{C}$ is shown in Fig. 4. In decomposition form, the output can be simple represented as the OR of all logical terms. Through recombination, we recombine the logical terms to obtain a better logical expression with various gates. Recombining ($\overline{a}\,\overline{b}\,\overline{c}\,d$ and $\overline{a}\,\overline{b}\,c\,\overline{d}$), ($\overline{a}\,b\,\overline{c}\,\overline{d}$ and $a\,\overline{b}\,\overline{c}\,\overline{d}$), we find that two XOR gates are generated. To optimize the XOR logic, we combine $\overline{a}\,\overline{b}\,\overline{c}\,\overline{d}$ with them respectively to generate a simpler logic expression of $\overline{C}$.

According to the logic expression of the output $\overline{C}$, the corresponding circuit can be constructed with only one AND-gate and two NOR-gates on its critical path as follows:

$$\text{Proposed } C = \overline{(\overline{a|b}\&\overline{c\&d})|(\overline{c|d}\&\overline{a\&b})}. \tag{4}$$

The recombination of decomposed S in "Pro1" is shown in Fig. 5. The logic terms decomposed from S are simplified to $\overline{(a \oplus b)}(c \oplus d) \,|\, (a \oplus b)\overline{(c \oplus d)} \,|\, abcd$ after two stages recombination. The simplified logic expression of output S is given as follows:

$$\begin{aligned} \text{Pro1} - S &= \overline{(a \oplus b)}(c \oplus d) \,|\, (a \oplus b)\overline{(c \oplus d)} \,|\, abcd \\ \text{Pro1} - C &= \overline{(\overline{a|b}\&\overline{c\&d})|(\overline{c|d}\&\overline{a\&b})}. \end{aligned} \tag{5}$$

According to the logic expressions of output S and C, the first proposed approximate 4-2 compressor (referred to as "Pro1" in the following) can be designed, as shown in Fig. 6. This 4-2 compressor only introduces a single error at "1111," the accuracy of it is relatively high. In the circuit architecture of "Pro1," the compound AND–OR gate AO222 and AND–OR–inverter gate AOI22 are used, and the normalized gate delay of them i "1" and "0.5," respectively. Moreover, we use several inverters in the architecture, and the normalized gate delay of them is "0.2," as shown in Table I.

As shown in Fig. 6, the dotted lines in graph mark the critical path of the approximate compressor "Pro1." There are one NOR-2, one NAND-2, two inverters, and one AO222 on the critical path. Calculate the critical path delay of "Pro1" as follows, and this compressor reduces the delay for 8% and 27% compared to "Yang1" and "Strollo3" with the same accuracy, respectively:

$$\text{Pro1} : 0.4 + 0.3 + 0.2 + 0.2 + 1 = 2.1. \tag{6}$$

B. Optimized Critical Path: "Pro2"

In the recombination of the decomposed output S, if $abcd$ is omitted, another error will be introduced at "1111." This kind of compressor behavior corresponds to the circuit proposed in "Lin" (Table II, C). Recombine the terms in decomposed S except the term $abcd$, as shown in Fig. 7. The logic expression of the second proposed approximate compressor (referred as "Pro2" in the following) is shown in the following equation:

$$\begin{aligned} \text{Pro2} - S &= (a \oplus b)\overline{(c \oplus d)}|\overline{(a \oplus b)}(c \oplus d) \\ \text{Pro2} - C &= \overline{(\overline{a|b}\&\overline{c\&d})|(\overline{c|d}\&\overline{a\&b})}. \end{aligned} \tag{7}$$

Similar to "Pro1," we select the compound AND–OR gate AO22 and AND–OR–inverter gate AOI22 to construct circuit architecture of "Pro2." As dotted path in Fig. 8, the critical path of "Pro2" has one NOR-2, one NAND-2, two inverters, and one AO22 gate. The normalized delay calculation is presented

Authorized licensed use limited to: VIT University. Downloaded on January 25,2023 at 08:37:47 UTC from IEEE Xplore. Restrictions apply.

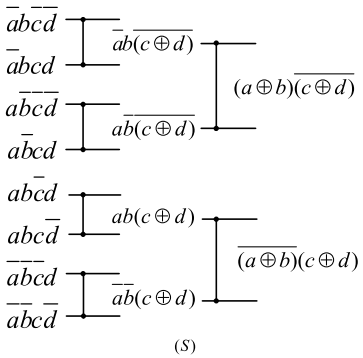

Fig. 7. Recombination of S in "Pro2."

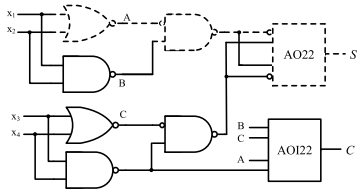

Fig. 8. Circuit implementation of "Pro2."

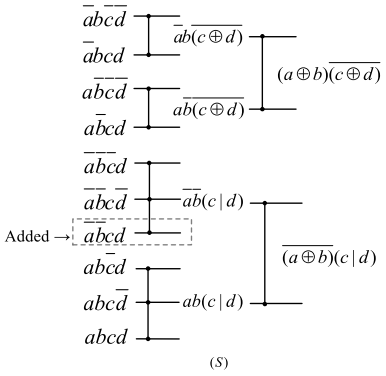

Fig. 9. Add a term in the recombination of "Pro3."

in (8), and the critical path delay of "Pro2" is 5% faster than "Lin" with the same truth table

$$\text{Pro2} : 0.4 + 0.3 + 0.2 + 0.2 + 0.8 = 1.9. \tag{8}$$

C. Optimized Critical Path: "Pro3"

Consider adding a term $\bar{a}\,\bar{b}\,cd$ to the decomposed S, and three-term recombination can be carried out in the first stage, as shown in Fig. 9. The XOR-gate "$(c \oplus d)$" in "Pro2" can be replaced by an OR-gate "$(c|d)$." Furthermore, we use De Morgan's laws to optimize the critical path of approximate 4-2 compressor "Pro3"

$$\begin{aligned}\text{Pro3} - S &= (a\oplus b)\overline{(c\oplus d)}\,|\,\overline{(a\oplus b)}(c|d)\\ &= \overline{(\overline{(a\oplus b)}\,|\,(c\oplus d))\&(\overline{(a\oplus b)}\,\overline{(c|d)})}\\ \text{Pro3} - C &= \overline{(a|\bar{b}\&c\&d)\,|\,(\bar{c}|d\&a\&b)}.\end{aligned} \tag{9}$$

Due to the term $\bar{a}\,\bar{b}\,cd$ added to the recombination of S, this third proposed approximate compressor "Pro3" introduces an error "+1" at "1100" like "Yang2" (Table II, B). The critical path of the architecture of "Pro3" in Fig. 10 has one NOR-2, one NAND-2, two inverters, and one compound AND–OR–inverter gate OAI22, and the normalized delay values 1.6. Compared to "Yang2," it achieves 40% critical path delay reduction

$$\text{Pro3} : 0.4 + 0.3 + 0.2 + 0.2 + 0.5 = 1.6. \tag{10}$$

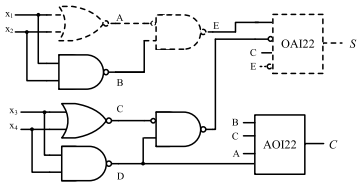

Fig. 10. Architecture of approximate compressor "Pro3."

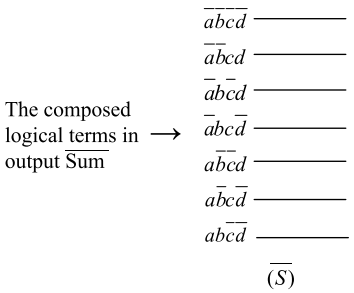

Fig. 11. Decomposed $\overline{S}$ with fewer terms.

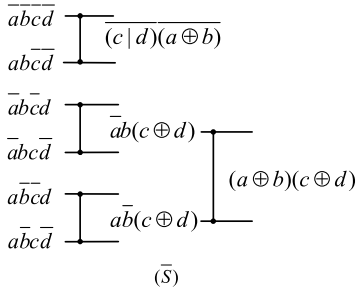

Fig. 12. Recombination of decomposed $\overline{S}$.

D. Optimized Critical Path: "Pro4"

All compressors proposed above are designed by recombination of the terms in decomposed S. In this section, we decompose the reversed logic $\overline{S}$ with fewer terms, as shown in Fig. 11.

Fig. 12 shows the recombination of the terms in output $\overline{S}$. By omitting the term $\bar{a}\,\bar{b}\,cd$, the logical expression of $\overline{S}$ can be expressed as follows:

$$\begin{aligned}\text{Pro4} - \overline{S} &= (a\oplus b)\overline{(c\oplus d)}\,|\,\overline{(a\oplus b)}(c|d)\\ \text{Pro4} - S &= \overline{\overline{(c|d)}\,\overline{(a\oplus b)}\,|\,(a\oplus b)(c\oplus d)}\\ \text{Pro4} - C &= \overline{(a|\bar{b}\&c\&d)\,|\,(\bar{c}|d\&a\&b)}.\end{aligned} \tag{11}$$

The critical path in circuit architecture (Fig. 13) of "Pro4" compressor has one NOR-2, one NAND-2, two inverters, and one compound AND–OR–inverter gate AOI22. The normalized delay of this critical path is equal to "Pro3" and also has 40% faster than "Yang2" with the same truth table, as shown in Table II (B)

$$\text{Pro4} : 0.4 + 0.3 + 0.2 + 0.2 + 0.5 = 1.6. \tag{12}$$

Authorized licensed use limited to: VIT University. Downloaded on January 25,2023 at 08:37:47 UTC from IEEE Xplore. Restrictions apply.

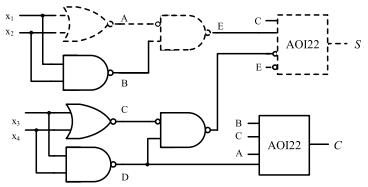

Fig. 13. Circuit architecture of "Pro4."

TABLE III
BEHAVIOR OF FLAG BITS O_4, O_3, O_2, AND O_1

Inputs "0" number	O_4, O_3, O_2, O_1
0	0000
1	0001
2	0011
3	0111
4	1111

E. Optimize Sorting Technique: "Pro5"

Since the reversed logic gates behave faster (as NAND-2 consumes less delay than AND-2), we optimized the sorting technique from sorting "1" in the inputs to sorting "0" to employ more reversed logical gates. The optimization corresponded to a circuit architecture with more reversed gates, delivering a shorter critical path.

Approximate 4-2 compressor have four inputs "a, b, c, and d." We first design "O_1, O_2, O_3, and O_4" to mark the number of "0" in the inputs. The behavior of these flag bits is presented in Table III, for example, when there are at least three "0" in the inputs, the flag bit "O_3" is equal to "1."

To detect the number of "0" in the inputs, the NOR and NAND gates are employed to construct the detection bits "d_1, d_2, d_3, and d_4." The output of NOR-2 is equal to "1," which means that both two inputs are "0," and the output of NAND-2 gate is equal to "1," which that there are at least one "0" in the inputs. Through the logic combination of detection bits in (13), the corresponding expression for "0" number in the inputs can be obtained, as shown in Table IV

$$\begin{aligned} d_1 &= \overline{a|b} \\ d_2 &= \overline{c|d} \\ d_3 &= \overline{a\&b} \\ d_4 &= \overline{c\&d}. \end{aligned} \tag{13}$$

Accordingly, the expressions of flag bits "O_1, O_2, O_3, and O_4" can be constructed by "d_1, d_2, d_3, and d_4" as shown in the following:

$$\begin{aligned} O_4 &= d_1\&d_2 \\ O_3 &= (d_1|d_2)\&(d_3\&d_4) \\ O_2 &= (d_1|d_2)|(d_3\&d_4) \\ O_1 &= d_3|d_4. \end{aligned} \tag{14}$$

According to the truth table of most accurate approximate 4-2 compressor (Table II, A), the output C is equal to "1" as long as there are more than one "1" in the inputs, which means that there are less than three "0." Therefore, the output C is equal to the reversed flag bit $\overline{O_3}$. The sum bit of the approximate compressor is equal to 1 when the inputs exist "1" (except for the case of two "1" inputs). Therefore, the logic expression of output S and C can be constructed by the flag bit "O_1, O_2, O_3, and O_4" as follows:

$$\begin{aligned} \text{Pro5} - C &= \overline{O_3} \\ \text{Pro5} - S &= (O_3\&\overline{O_4})|\overline{O_2}. \end{aligned} \tag{15}$$

In the architecture of "Pro5" in Fig. 14 we can find, the critical path of this circuit has one NOR-2, one OR-2, one NAND-2, and one compound AND–OR–inverter gate OAI22. The behavior of "Pro5" approximate 4-2 compressor corresponds to "Yang1," and it only introduces a single error at "1111," as shown in Table II (A). The normalized delay of critical path in "Pro5" values is only 1.9, and its about 17% reduction compared to "Yang1" with the same accuracy

$$\text{Pro5} : 0.4 + 0.7 + 0.3 + 0.5 = 1.9. \tag{16}$$

TABLE IV
EXPRESSION FOR "0" NUMBER

Inputs "0" number	Logic Expression
none	$\overline{d_3\|d_4}$
at least 1	$d_3\|d_4$
at least 2	$(d_1\|d_2)\|(d_3\&d_4)$
at least 3	$(d_1\|d_2)\&(d_3\&d_4)$
4	$d_1\&d_2$

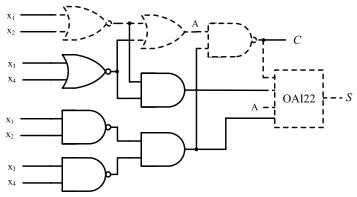

Fig. 14. Circuit implementation of "Pro5."

IV. IMPLEMENTATION AND PERFORMANCE ANALYSIS

In this section, we implement approximate 8 × 8 and 16 × 16 multipliers with the proposed compressors "Pro1–Pro5" and all mentioned previous compressors and analyze the error characteristic of these multipliers. In order to obtain better performance–accuracy tradeoff, the hybrid multipliers are proposed. Synthesized by TSMC standard cell library in 28-nm CMOS, our designs have better electrical performance with high accuracy. Furthermore, we verify the efficacy of the proposed compressors through image blending and matrix multiplication applications.

A. Multiplier Design

The partial product accumulation scheme in Fig. 15 is used to implement 8 × 8 approximate multipliers with the proposed compressors "Pro1–Pro5." In order to make the fair comparison of each compressor, the same partial products accumulation structure is introduced in all multipliers we implemented. For a comprehensive electrical performance comparison, we implemented multipliers equipped with all mentioned compressors whether high accuracy or not. All of

Authorized licensed use limited to: VIT University. Downloaded on January 25,2023 at 08:37:47 UTC from IEEE Xplore. Restrictions apply.

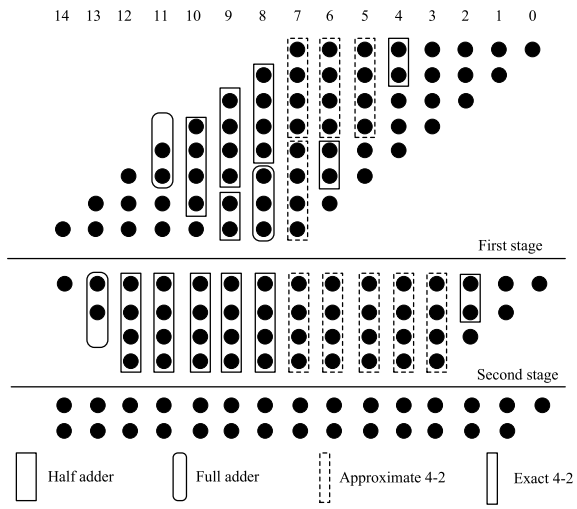

Fig. 15. Partial product accumulation scheme for all multipliers equipped with all mentioned approximate compressors.

TABLE V
ERROR METRICS OF APPROXIMATE 8 × 8 MULTIPLIERS

Compressor	Error Metrics				
	ER	NMED	MRED	NoEB	PRED
Momeni [18]	0.93	1.63×10^{-3}	9.03×10^{-2}	8.92	2.56×10^{-1}
Venka [19]	0.63	1.25×10^{-3}	1.28×10^{-2}	9.05	1.43×10^{-1}
Akbar1 [24]	0.84	2.93×10^{-3}	4.82×10^{-2}	8.01	3.52×10^{-1}
Akbar2 [24]	0.63	1.29×10^{-3}	1.29×10^{-2}	8.98	1.47×10^{-1}
Sabetz [25]	0.98	1.93×10^{-3}	9.52×10^{-2}	8.71	2.83×10^{-1}
Ahma [26]	0.77	1.47×10^{-3}	1.70×10^{-2}	8.94	1.86×10^{-1}
Yang1 [20]	0.04	4.79×10^{-5}	2.40×10^{-4}	11.60	2.32×10^{-3}
Yang2 [20]	0.20	2.50×10^{-4}	2.44×10^{-3}	10.47	2.69×10^{-2}
Yang3 [20]	0.28	3.68×10^{-4}	3.05×10^{-3}	10.13	3.36×10^{-2}
Strollo1 [23]	0.28	4.53×10^{-4}	3.68×10^{-3}	9.85	4.41×10^{-2}
Strollo2 [23]	0.09	1.30×10^{-4}	7.65×10^{-4}	10.86	8.06×10^{-3}
Strollo3 [23]	0.04	4.79×10^{-5}	2.40×10^{-4}	11.60	2.32×10^{-3}
Lin [21]	0.04	9.16×10^{-5}	4.68×10^{-4}	10.75	6.15×10^{-3}
Ha [22]	0.28	4.53×10^{-4}	3.68×10^{-3}	9.85	4.41×10^{-2}
Ranjbar1 [35]	0.88	2.15×10^{-3}	4.92×10^{-2}	8.49	3.09×10^{-1}
Ranjbar2 [35]	0.90	1.88×10^{-3}	4.29×10^{-2}	8.73	2.95×10^{-1}
Ranjbar3 [35]	0.88	2.15×10^{-3}	4.78×10^{-2}	8.49	3.05×10^{-1}
Pro1,5	0.04	4.79×10^{-5}	2.40×10^{-4}	11.60	2.32×10^{-3}
Pro2	0.04	9.16×10^{-5}	4.68×10^{-4}	10.75	6.15×10^{-3}
Pro3,4	0.20	2.50×10^{-4}	2.44×10^{-3}	10.47	2.69×10^{-2}
Hybrid1	0.11	1.99×10^{-4}	1.86×10^{-3}	10.60	1.97×10^{-2}
Hybrid2	0.14	2.21×10^{-4}	2.13×10^{-3}	10.53	2.41×10^{-2}

multipliers use the same half adders, full adders, and exact 4-2 compressors except for approximate 4-2 compressors.

There are three schemes in multipliers we designed.

1) C-N Scheme: There are only N less significant approximate columns in the partial product array of the multiplier. Moreover, only one type of approximate 4-2 compressor is used in one single multiplier. The multipliers constructed with "Pro1–Pro5," "Yang1–Yang3," "Strollo1–Strollo3," "Lin," "Ha," "Momeni," "Venka," "Akbar1-2," "Sabetz," "Ahma," and "Ranjbar1-3" are all C-N scheme ($N = 8$ in 8×8 and $N = 16$ in 16×16).

2) C-NH Scheme: Only N less significant columns of the multiplier use approximate compressors such as C-N scheme. However, compared with C-N, C-NH scheme is a hybrid scheme. In this scheme, we adopt two type approximate compressors in different columns, "H" represents 0–H columns that use one type of compressor and H-N that uses another type.

Because of the accuracy–performance tradeoff, the lower accuracy compressor obtains better electrical performance. To illustrate the C-NH scheme more clearly, for instance, we adopt "Pro1" (with only a single error) and "Pro3" (introduces two errors) in one single multiplier "Hybrid1" with C-NH scheme ($N = 8$ and $H = 6$). Obviously, in partial product array, errors produced by various columns have different weights. The error produced by column 3 has less effect on the final result than the error produced by column 7. Thus, in "Hybrid1" multiplier, "Pro1" only adopted in column 7 due to its accuracy, whereas "Pro3" is adopted in column 0–6 for a better performance–accuracy tradeoff.

3) C-NS Scheme: C-NS scheme is also a hybrid multiplier scheme with only N less significant columns imprecise as C-N and C-NH schemes. However, this scheme adopts various approximate compressors in different compression stages instead of different columns.

According to the truth table, compressors "Pro1, Pro2, and Pro5" are symmetric, which means that the sequence of compressor inputs does not affect the ER of output. When using an asymmetric compressor such as "Pro3 and Pro4," extra configuration of the interstage connection is required to obtain higher accuracy.

In hybrid multiplier "Hybrid2," we adopt the C-NS scheme ($N = 8$) and use "Pro1" and "Pro3" to gain a better tradeoff between accuracy and electrical performance without extra configuration. In the first stage, the asymmetric "Pro3" compressors are adopted, and the inputs of first-stage compressors are uniformly random, so the outputs of first-stage compressors are irrelevant to input connection. The inputs of second-stage compressors "Pro1" are the outputs of "Pro3" since the compressor "Pro1" only introduces error at "1111," and without any configuration, the outputs of compressors in the second stage are still irrelevant to the connection.

B. Error Metrics

Some metrics in [30]–[33] are proposed to assess the accuracy of approximate 8×8 and 16×16 multipliers. We define M as the exact multiplication result and M' as the result calculated by approximate multiplier. The calculation method of metrics we used are shown.

1) *ER:* The rate of $|M - M'| \neq 0$.
2) *Normalized Mean Error Distance (NMED):* The average value of $|M - M'|$ is divided by $(2^n - 1)^2$.
3) *Mean Relative Error Distance (MRED):* The average $|M - M'|/|M|$.
4) *Number of Effective Bits (NoEB):* $2n - \log_2(1 +$ The root mean square $of\ |M - M'|)$.
5) Probability of Relative Error Distance (PRED): The probability of $(|M - M'|/|M|) > 0.02$.

All of the 8×8 approximate multipliers we implemented have been carried out the traversal simulations and the 16×16 approximate multipliers have been simulated with 100 million uniform distributed random vectors. The error metrics of implemented 8×8 and 16×16 multipliers are summarized in Tables V and VI. Our design "Pro1" and "Pro5" have the

Authorized licensed use limited to: VIT University. Downloaded on January 25,2023 at 08:37:47 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
ERROR METRICS OF APPROXIMATE 16 × 16 MULTIPLIERS

Compressor	Error Metrics				
	ER	NMED	MRED	NoEB	PRED
Momeni [18]	1.00	1.64×10^{-5}	3.79×10^{-1}	1.75	3.79×10^{-3}
Venka [19]	0.94	5.72×10^{-6}	1.28×10^{-1}	1.10	1.43×10^{-3}
Akbar1 [24]	0.99	7.51×10^{-6}	2.97×10^{-1}	1.67	2.71×10^{-3}
Akbar2 [24]	0.94	6.20×10^{-6}	1.39×10^{-1}	1.95	1.50×10^{-3}
Sabetz [25]	1.00	2.39×10^{-5}	1.59×10^{-1}	1.24	3.00×10^{-3}
Ahma [26]	0.77	1.47×10^{-5}	1.70×10^{-2}	3.94	1.86×10^{-3}
Yang1 [20]	0.39	7.16×10^{-6}	1.32×10^{-2}	8.75	1.33×10^{-4}
Yang2 [20]	0.72	5.62×10^{-6}	3.51×10^{-2}	3.58	6.81×10^{-4}
Yang3 [20]	0.87	2.15×10^{-5}	9.20×10^{-2}	1.71	1.50×10^{-3}
Strollo1 [23]	1.00	2.50×10^{-5}	6.34×10^{-1}	1.59	6.33×10^{-3}
Strollo2 [23]	0.70	1.77×10^{-5}	3.33×10^{-2}	3.02	1.11×10^{-3}
Strollo3 [23]	0.39	7.16×10^{-6}	1.32×10^{-2}	8.75	1.33×10^{-4}
Lin [21]	0.39	1.21×10^{-5}	3.17×10^{-2}	8.56	5.01×10^{-4}
Ha [22]	1.00	2.50×10^{-5}	6.34×10^{-1}	1.59	6.33×10^{-3}
Ranjbar1 [35]	1.00	8.15×10^{-5}	4.53×10^{-1}	1.60	3.55×10^{-3}
Ranjbar2 [35]	0.99	6.12×10^{-5}	2.19×10^{-1}	1.05	1.97×10^{-2}
Ranjbar3 [35]	1.00	6.56×10^{-5}	3.04×10^{-1}	1,96	2.55×10^{-3}
Pro1,5	0.39	7.16×10^{-6}	1.32×10^{-2}	8.75	1.33×10^{-4}
Pro2	0.39	1.21×10^{-5}	3.17×10^{-2}	8.56	5.01×10^{-4}
Pro3,4	0.72	5.62×10^{-6}	3.51×10^{-2}	3.58	6.81×10^{-4}
Hybrid1	0.59	5.66×10^{-6}	5.53×10^{-2}	5.87	6.55×10^{-4}
Hybrid2	0.63	9.21×10^{-6}	5.57×10^{-2}	5.57	5.09×10^{-4}

same metrics as "Yang1" because of the same truth table, so "Yang1," "Pro1," and "Pro5" have the minimum ER and the largest NoEB. Other compressors we proposed with the maximum ER 0.20 (8 × 8) also perform well. The last two rows of Tables V and VI are metrics of hybrid multipliers we designed. "Hybrid1" and "Hybrid 2" with small ER, NMED, MRED, PRED, and large NoEB are designed by the C-NH scheme and C-NS scheme, respectively.

C. Experimental Results and Analysis

The circuits are implemented by Verilog and synthesized by Synopsys Design Compiler using the TSMC 28-nm CMOS cell library. The experimental results are summarized in Tables VII and VIII. Table VII shows that Sabetz and Ahma attained the minimum delay in all 8 × 8 multipliers due to their simple structures. They have about 20% improvement in speed, but their ER up to 98% made them less adoptable in applications with high-precision requirements. According to the error metrics, our design "Pro1–Pro5" exhibited high accuracy at least equal to "Yang2." The low ER compressors we proposed are effective in high-precision applications with modest 17.9% improvement in speed, as shown in Tables VII and VIII.

The experimental results show that our designs have delay, area, and power reduction. For instance, the compressor "Pro1" presents a certain reduction on area than "Strollor3" and "Yang1" with the same accuracy. The area difference of multipliers is related to the architecture difference of compressors. According to [51], the XOR gate consumes more area about 1.5 times than compound gate AOI22, and however, the full adder consumes more than two times area compared with XOR gate. Because of the large area consumption of XOR gate and full adder, the compressor "Strollo3" equipped with full adder consumes more area. Similarly, the architecture of "Yang1" equipped with two XOR gates consumes more area than "Pro1" without XOR gate or full adder. Another factor is synthesis tactics. For Synopsys Design Compiler,

TABLE VII
EXPERIMENTAL RESULTS OF IMPLEMENTED 8 × 8 MULTIPLIERS

	delay (ps)	delay (reduction) (exact)	area (μm^2)	power (mW)	ADP ($\mu m^2 \cdot ns$)	ADP reduction (exact)	ADP reduction (same accuracy)
Exact	201		845.37	1.0474	169.91		
Momeni [18]	162	-19.40%	518.78	0.5477	84.04	-50.53 %	
Venka [19]	164	-18.41%	517.61	0.5276	84.89	-50.03%	
Akbar1 [24]	163	-18.91%	471.24	0.4949	76.81	-54.79%	
Akbar2 [24]	163	-18.91%	474.09	0.4968	77.27	-54.52%	
Sabetz [25]	160	-20.39%	393.45	0.4483	62.95	-62.95%	
Ahma [26]	160	-20.39%	425.54	0.4369	68.08	-59.93%	
Yang1 [20]	172	-14.43%	893.08	1.1022	153.6	-9.60 %	
Yang2 [20]	171	-14.93%	874.77	1.0618	149.58	-11.97%	
Yang3 [20]	176	-12.44%	764.06	0.9803	134.47	-20.86%	
Strollo1 [23]	176	-12.44%	664.94	0.8341	117.03	-31.12%	
Strollo2 [23]	175	-12.94%	803.37	0.9884	140.58	-17.26%	
Strollo3 [23]	173	-13.93%	905.85	1.1365	156.71	-7.77%	
Ranjbar1 [35]	163	-18.90%	551.88	0.5993	89.95	-47.06 %	
Ranjbar2 [35]	163	-18.90%	492.07	0.5108	79.22	-53.37 %	
Ranjbar3 [35]	163	-18.90%	506.52	0.5007	82.56	-51.41 %	
Lin [21]	174	-13.43%	852.43	1.0645	148.32	-12.71%	
Ha [22]	174	-13.43%	706.27	0.9233	122.89	-27.67%	
pro1	164	-18.41%	582.46	0.6096	95.52	-43.78%	-37.81%
pro2	164	-18.41%	557.93	0.6163	91.50	-46.15%	-21.81%
pro3	164	-18.41%	574.06	0.5819	94.15	-44.59%	-55.43%
pro4	165	-17.91%	510.38	0.5431	84.72	-50.14%	-43.36%
pro5	164	-18.41%	498.12	0.5036	81.69	-51.92%	-46.82%
hybrid1	164	-18.41%	541.30	0.5442	88.77	-47.75%	
hybrid2	165	-17.91%	538.61	0.5513	88.87	-47.70%	

TABLE VIII
EXPERIMENTAL RESULTS OF IMPLEMENTED 16 × 16 MULTIPLIERS

	delay (ps)	delay (reduction) (exact)	area (μm^2)	power (mW)	ADP ($\mu m^2 \cdot ns$)	ADP reduction (exact)	ADP reduction (same accuracy)
Exact	264		2849.95	4.4845	752.39		
Momeni [18]	234	-11.36%	2161.65	2.6147	505.83	-32.77 %	
Venka [19]	237	-10.23%	2095.79	2.5261	496.70	-33.98%	
Akbar1 [24]	235	-10.98%	2023.72	2.6059	475.57	-36.79%	
Akbar2 [24]	235	-10.98%	2067.57	2.5781	485.88	-35.42%	
Sabetz [25]	232	-12.12%	1711.75	2.1897	397.13	-47.22%	
Ahma [26]	233	-11.74%	1826.50	2.2393	425.57	-43.44%	
Yang1 [20]	253	-4.17%	2445.24	3.5880	618.65	-17.77 %	
Yang2 [20]	248	-6.06%	2367.96	3.3305	587.25	-21.95%	
Yang3 [20]	251	-4.92%	2231.21	3.0409	560.03	-25.57%	
Strollo1 [23]	245	-7.20%	2279.42	3.6285	558.46	-25.78%	
Strollo2 [23]	243	-7.95%	2433.14	3.7920	591.25	-21.42%	
Strollo3 [23]	251	-5.18%	2407.94	3.7855	604.39	-19.67%	
Ranjbar1 [35]	235	-10.98%	2097.64	2.5486	492.95	-34.48%	
Ranjbar2 [35]	236	-10.61%	1989.45	2.4201	469.51	-37.60%	
Ranjbar3 [35]	235	-10.98%	1945.10	2.4380	459.04	-38.99%	
Lin [21]	249	-5.68%	2305.63	3.5189	574.10	-23.70%	
Ha [22]	247	-6.44%	2163.67	3.1064	534.43	-28.97%	
pro1	244	-7.58%	1893.86	2.2910	462.10	-38.58%	-25.31%
pro2	238	-9.85%	2200.97	2.7676	523.83	-30.38%	-8.76%
pro3	237	-10.23%	2242.63	2.7791	531.50	-29.36%	-9.49%
pro4	237	-10.23%	2345.28	2.8956	555.83	-26.12%	-5.35%
pro5	239	-9.47%	2196.39	2.7530	524.94	-30.23%	-15.15%
hybrid1	240	-9.09%	2256.41	2.7382	541.54	-28.02%	
hybrid2	240	-9.09%	2198.11	2.7699	527.55	-29.88%	

this synthesis tool selects different synthesis tactics at every process node, and it will select different synthesis tactics for designs with various critical paths, including the area optimization part in synthesis.

Compared to the exact multiplier, the delay and ADP reduction of our design can achieve up to 18% and 52%, respectively. Under the same accuracy, the ADP reduction achieved by our designs is about 21%–55%. For the most accurate multiplier "Yang1," the optimization of ADP can reach 55.43% by "Pro5" with the same accuracy. The ER versus ADP reduction (compared with exact multiplier) graph is plotted to illustrate the tradeoffs between electrical performance and accuracy. This performance–accuracy tradeoff graph (Fig. 17) shows that all designs we proposed have good tradeoff and "Pro5" has the best tradeoff among these high-accuracy approximate multipliers.

D. Image Processing

The image blending application as one of the error-resilient image processing applications is performed using "Pro1–Pro5" and variable existing designs to investigate the behavior of approximate multipliers.

Authorized licensed use limited to: VIT University. Downloaded on January 25,2023 at 08:37:47 UTC from IEEE Xplore. Restrictions apply.

Fig. 16. Image blending results computed by 8 × 8 multipliers.

TABLE IX
METRICS OF IMAGE BLENDING APPLICATIONS

Metrics	Momeni [18]		Venka [19]		Akbar1 [24]		Akbar2 [24]		Sabetz [25]		Ahma [26]		[A]		[B]		Yang3 [20]	
	8	16	8	16	8	16	8	16	8	16	8	16	8	16	8	16	8	16
PSNR	43.47	23.15	48.70	32.59	47.57	27.12	48.53	32.57	38.36	22.43	48.81	31.05	68.18	58.33	57.97	47.78	56.37	44.42
SSIM	0.99	0.65	1.00	0.95	1.00	0.75	1.00	0.95	0.98	0.63	1.00	0.86	1.00	1.00	1.00	1.00	1.00	0.99
	[C]		[D]		Strollo2 [23]		Ranjbar1 [35]		Ranjbar2 [35]		Ranjbar3 [35]		Hybrid1		Hybrid2			
PSNR	54.95	52.45	54.23	42.66	58.56	50.99	45.74	23.83	43.44	26.91	46.93	23.19	60.21	53.19	59.71	50.15		
SSIM	1.00	1.00	1.00	0.99	1.00	1.00	1.00	0.67	0.99	0.75	1.00	0.65	0.99	1.00	1.00	1.00		

$[A]$:Yang1,Strollo3 and Proposed1,5; $[B]$:Yang2,Proposed3,Proposed4; $[C]$:Lin and Proposed2; $[D]$:Ha and Strollo1;

As one application commonly used to investigate the behavior of approximate multiplier, image blending is processed by multiply two input pixel on the pixel–pixel basis and then scales back to 8 bit. We blend images of "Lena" and "Cameraman" by all previous approximate multipliers and our designs in schemes C-N ($N = 8$ and 16), C-NS, and C-NH. The image results of "Lena" and "Cameraman" are shown in Fig. 16, and our design with high accuracy performs well in this application both C-N scheme and Hybrid schemes. To assess the quality of blending application, the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are calculated in Table IX. The table shows that all of the proposed 4-2 compressors perform well with the PSNR larger than 47.78 dB and the SSIM equal to 1.00. "Pro1 and Pro5," "Yang1," and "Strollo3" exhibit the best values of metrics due to the same accuracy. Different with high ER compressors, the approximate 4-2 compressors with high accuracy present a relatively stable precision performance. Even if the approximate columns are enlarged to $N = 16$, they still have a high image processing quality.

E. Matrix Multiplication

As one of the most critical kernel of machine learning, convolution, and so on, matrix multiplication, its efficiency

Authorized licensed use limited to: VIT University. Downloaded on January 25,2023 at 08:37:47 UTC from IEEE Xplore. Restrictions apply.

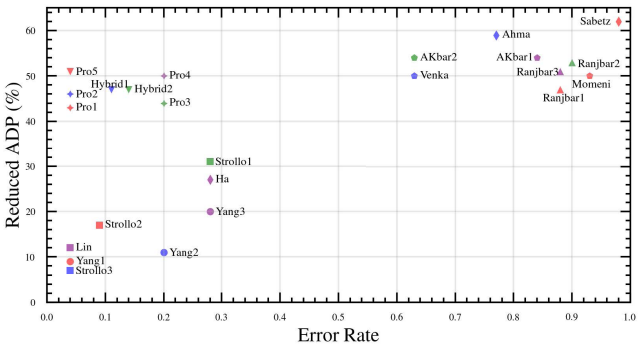

Fig. 17. ER versus ADP reduction of proposed approximate 8-bit multipliers and existing high-accuracy designs.

TABLE X
ERROR METRICS OF MATRIX MULTIPLICATION APPLICATIONS

Compressor	Error Metrics	
	NMED	MRED
Momeni [18]	2.05×10^{-2}	4.25×10^{-3}
Venka [19]	8.34×10^{-3}	1.81×10^{-3}
Akbar1 [24]	5.90×10^{-3}	1.26×10^{-3}
Akbar2 [24]	8.53×10^{-3}	1.86×10^{-3}
Sabetz [25]	3.94×10^{-2}	7.89×10^{-3}
Ahma [26]	5.24×10^{-3}	1.14×10^{-3}
Yang1 [20]	1.99×10^{-4}	4.71×10^{-5}
Yang2 [20]	1.26×10^{-3}	2.68×10^{-4}
Yang3 [20]	1.40×10^{-3}	3.02×10^{-4}
Strollo1 [23]	2.45×10^{-3}	5.42×10^{-4}
Strollo2 [23]	1.18×10^{-3}	2.59×10^{-4}
Strollo3 [23]	1.99×10^{-4}	4.71×10^{-5}
Lin [21]	3.85×10^{-4}	9.09×10^{-5}
Ha [22]	2.45×10^{-3}	5.42×10^{-4}
Ranjbar1 [35]	9.15×10^{-3}	1.82×10^{-3}
Ranjbar2 [35]	8.26×10^{-3}	1.64×10^{-3}
Ranjbar3 [35]	9.09×10^{-3}	1.83×10^{-3}
Pro1,5	1.99×10^{-4}	4.71×10^{-5}
Pro2	3.85×10^{-4}	9.09×10^{-5}
Pro3,4	1.26×10^{-3}	2.68×10^{-4}
Hybrid1	8.15×10^{-4}	1.78×10^{-4}
Hybrid2	9.09×10^{-4}	1.96×10^{-4}

affects the application system in some degree. For instance, for an MNIST classifier, the accuracy of matrix multiplication determines whether the system can output the correct results. With increasing layers in classifier, the noise introduced by operation will amplify layer-by-layer. Therefore, the classifier system is more likely to output a correct result with a high-accuracy matrix multiplication in each layer.

In this section, we implemented 5×5 matrix multiplication [10] by all of the mentioned 8-bit multipliers. All the matrix multipliers have been simulated with 10 million random vectors under the Gaussian distribution. The error metrics of matrix multiplications are presented in Table X. As shown in the table, the high-accuracy multipliers perform better than the multipliers constructed with low-accuracy compressors. The compressors "Pro1 and Pro5, Yang1, and Strollo3" give the best error metrics, and their NMED and MRED are at least 10 times and 100 times lower than other low-accuracy compressors in matrix multiplication, respectively. Further research can be conducted to explore the approximate compressors in DSP/CNN and floating-point applications.

V. CONCLUSION

In this article, various high-accuracy approximate 4-2 compressors with better performance have been proposed. Four of these compressors (Pro1–Pro4) are designed by decomposition and recombination methods to reduce the critical path delay and area and power consumption of the circuits. The fifth design "Pro5" is designed by optimizing the sorting technique, and it has the best ADP–accuracy tradeoff. All the previous state-of-the-art compressors, "Pro1-5" compressors, and two hybrid schemes "Hybrid1-2" have been implemented in the TSMC 28-nm CMOS process to investigate the electrical performance. The experimental results indicate that our designs have about 18% delay reduction, 43%–52% ADP reduction compared with exact multiplier, and 21%–55% optimization in ADP compared to the compressors with the same accuracy. This article further verifies the efficacy of the proposed compressors through image blending and matrix multiplication applications.

REFERENCES

[1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *Proc. 18th IEEE Eur. TEST Symp. (ETS)*, May 2013, pp. 1–6.
[2] S. Venkataramani, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Computing approximately, and efficiently," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2015, pp. 748–751.
[3] S. Venkataramani, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Approximate computing and the quest for computing efficiency," in *Proc. 52nd Annu. Design Autom. Conf.*, Jun. 2015, pp. 1–6.
[4] 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)*, 2013, pp. 1–9.
[5] W. Liu, J. Xu, D. Wang, C. Wang, P. Montuschi, and F. Lombardi, "Design and evaluation of approximate logarithmic multipliers for low power error-tolerant applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 9, pp. 2856–2868, Sep. 2018.
[6] X. Cui, W. Liu, X. Chen, E. E. Swartzlander, and F. Lombardi, "A modified partial product generator for redundant binary multipliers," *IEEE Trans. Comput.*, vol. 65, no. 4, pp. 1165–1171, Apr. 2016.
[7] W. Liu *et al.*, "Design and analysis of approximate redundant binary multipliers," *IEEE Trans. Comput.*, vol. 68, no. 6, pp. 804–819, Jun. 2019.
[8] M. Imani, D. Peroni, and T. Rosing, "CFPU: Configurable floating point multiplier for energy-efficient computing," in *Proc. 54th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2017, pp. 1–6.
[9] K. M. Reddy, M. H. Vasantha, Y. B. N. Kumar, and D. Dwivedi, "Design of approximate booth squarer for error-tolerant computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 5, pp. 1230–1241, May 2020.
[10] S. Venkatachalam, E. Adams, H. J. Lee, and S.-B. Ko, "Design and analysis of area and power efficient approximate booth multipliers," *IEEE Trans. Comput.*, vol. 68, no. 11, pp. 1697–1703, Nov. 2019.
[11] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, "Design of approximate radix-4 booth multipliers for error-tolerant computing," *IEEE Trans. Comput.*, vol. 66, no. 8, pp. 1435–1441, Aug. 2017.
[12] 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.
[13] A. D. Booth, "A signed binary multiplication technique," *Quart. J. Mech. Appl. Math.*, vol. 4, pp. 236–240, 1951.
[14] O. L. Macsorley, "High-speed arithmetic in binary computers," *Proc. IRE*, vol. 49, no. 1, pp. 67–91, 1961.
[15] S. Jou, M.-H. Tsai, and Y.-L. Tsao, "Low-error reduced-width booth multipliers for DSP applications," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 50, no. 11, pp. 1470–1474, Nov. 2003.
[16] 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.
[17] 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.
[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.

Authorized licensed use limited to: VIT University. Downloaded on January 25,2023 at 08:37:47 UTC from IEEE Xplore. Restrictions apply.

[19] S. Venkatachalam and S.-B. Ko, "Design of power and area efficient approximate multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 5, pp. 1782–1786, May 2017.
[20] Z. Yang, J. Han, and F. Lombardi, "Approximate compressors for error-resilient multiplier design," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFTS)*, Oct. 2015, pp. 183–186.
[21] C.-H. Lin and I.-C. Lin, "High accuracy approximate multiplier with error correction," in *Proc. IEEE 31st Int. Conf. Comput. Design (ICCD)*, Oct. 2013, pp. 33–38.
[22] M. Ha and S. Lee, "Multipliers with approximate 4C2 compressors and error recovery modules," *IEEE Embedded Syst. Lett.*, vol. 10, no. 1, pp. 6–9, Mar. 2018.
[23] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo, "Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 9, pp. 3021–3034, Sep. 2020.
[24] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 4, pp. 1352–1361, Apr. 2017.
[25] F. Sabetzadeh, M. H. Moaiyeri, and M. Ahmadinejad, "A majority-based imprecise multiplier for ultra-efficient approximate image multiplication," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 11, pp. 4200–4208, Nov. 2019.
[26] M. Ahmadinejad, M. H. Moaiyeri, and F. Sabetzadeh, "Energy and area efficient imprecise compressors for approximate multiplication at nanoscale," *Int. J. Electron. Commun.*, vol. 110, Oct. 2019, Art. no. 152859.
[27] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 1, pp. 124–137, Jan. 2013.
[28] W. Xu, S. S. Sapatnekar, and J. Hu, "A simple yet efficient accuracy-configurable adder design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 6, pp. 1112–1125, Jun. 2018.
[29] F. Ebrahimi-Azandaryani, O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Block-based carry speculative approximate adder for energy-efficient applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 1, pp. 137–141, Jan. 2020.
[30] H. Jiang, C. Liu, N. Maheshwari, F. Lombardi, and J. Han, "A comparative evaluation of approximate multipliers," in *Proc. IEEE/ACM Int. Symp. Nanoscale Archit. (NANOARCH)*, Jul. 2016, pp. 191–196.
[31] D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, and N. Petra, "Approximate multipliers based on new approximate compressors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 12, pp. 4169–4182, Dec. 2018.
[32] V. Leon, G. Zervakis, S. Xydis, D. Soudris, and K. Pekmestzi, "Walking through the energy-error Pareto frontier of approximate multipliers," *IEEE Micro*, vol. 38, no. 4, pp. 40–49, Jul. 2018.
[33] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Trans. Comput.*, vol. 62, no. 9, pp. 1760–1771, Sep. 2013.
[34] G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, "Design-efficient approximate multiplication circuits through partial product perforation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 10, pp. 3105–3117, Oct. 2016.
[35] F. Ranjbar, Y. Forghani, and D. Bahrepour, "High performance 8-bit approximate multiplier using novel 4:2 approximate compressors for fast image processing," *Int. J. Integr. Eng.*, vol. 10, no. 1, pp. 1–20, 2018.
[36] L. O. Chua and T. Roska, "The CNN paradigm," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 40, no. 3, pp. 147–156, Mar. 1993.
[37] G. Lentaris, G. Chatzitsompanis, V. Leon, K. Pekmestzi, and D. Soudris, "Combining arithmetic approximation techniques for improved CNN circuit design," in *Proc. 27th IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Nov. 2020, pp. 1–4.
[38] X. Lian, Z. Liu, Z. Song, J. Dai, W. Zhou, and X. Ji, "High-performance FPGA-based CNN accelerator with block-floating-point arithmetic," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 8, pp. 1874–1885, Aug. 2019.
[39] M. Nourazar, V. Rashtchi, A. Azarpeyvand, and F. Merrikh-Bayat, "Code acceleration using memristor-based approximate matrix multiplier: Application to convolutional neural networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 12, pp. 2684–2695, Dec. 2018.
[40] M. S. Kim, A. A. D. Barrio, L. T. Oliveira, R. Hermida, and N. Bagherzadeh, "Efficient Mitchell's approximate log multipliers for convolutional neural networks," *IEEE Trans. Comput.*, vol. 68, no. 5, pp. 660–675, May 2019.
[41] M. S. Ansari, V. Mrazek, B. F. Cockburn, L. Sekanina, Z. Vasicek, and J. Han, "Improving the accuracy and hardware efficiency of neural networks using approximate multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 2, pp. 317–328, Oct. 2020.
[42] C. Chen, S. Yang, W. Qian, M. Imani, X. Yin, and C. Zhuo, "Optimally approximated and unbiased floating-point multiplier with runtime configurability," in *Proc. 39th Int. Conf. Computer-Aided Design*. New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 1–9.
[43] D. Peroni, M. Imani, and T. S. Rosing, "Runtime efficiency-accuracy tradeoff using configurable floating point multiplier," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 2, pp. 346–358, Feb. 2020.
[44] Y. Wang, J. Deng, Y. Fang, H. Li, and X. Li, "Resilience-aware frequency tuning for neural-network-based approximate computing chips," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 10, pp. 2736–2748, Oct. 2017.
[45] S. K. Bose *et al.*, "ADEPOS: A novel approximate computing framework for anomaly detection systems and its implementation in 65-nm CMOS," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 3, pp. 913–926, Mar. 2020.
[46] Z. G. Tasoulas, G. Zervakis, I. Anagnostopoulos, H. Amrouch, and J. Henkel, "Weight-oriented approximation for energy-efficient neural network inference accelerators," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 4670–4683, Dec. 2020.
[47] B. Kar, P. K. Gopalakrishnan, S. K. Bose, M. Roy, and A. Basu, "ADIC: Anomaly detection integrated circuit in 65-nm CMOS utilizing approximate computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 12, pp. 2518–2529, Dec. 2020.
[48] V. Leon, K. Asimakopoulos, S. Xydis, D. Soudris, and K. Pekmestzi, "Cooperative arithmetic-aware approximation techniques for energy-efficient multipliers," in *Proc. 56th Annu. Design Autom. Conf.* New York, NY, USA: Association for Computing Machinery, Jun. 2019, pp. 1–6.
[49] M. Masadeh, O. Hasan, and S. Tahar, "Machine-learning-based self-tunable design of approximate computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 4, pp. 800–813, Apr. 2021.
[50] W. Shi *et al.*, "A 0.4 V 298 nJ/op neural signal spectral feature extraction module with novel approximate MACs and custom compressors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 10, pp. 1733–1737, Oct. 2019.
[51] *TCB-N28HPCPLUSBWP12T3 0P140LVT TSMC N28HPC Standard Cell Library*, Taiwan Semicond. Manuf. Company, 28nm Datasheet, Mar. 2017.

Tianqi Kong received the bachelor's degree in microelectronics science and engineering from Jilin University, Changchun, China, in 2018. She is currently working toward the Ph.D. degree at the Institute of Microelectronics, Tsinghua University, Beijing, China.

Her research interests include fully homomorphic cryptography and VLSI architectures for cryptography.

Shuguo Li (Member, IEEE) received the bachelor's degree in computer science from Xidian University, Xi'an, China, in 1986, the M.S. degree in computer science from Shandong University, Shandong, China, in 1993, and the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, in 1999.

He is currently a Professor with the Institute of Microelectronics, Tsinghua University, Beijing, China. His current research interest includes the algorithm, design, and VLSI implementation for cryptography.

Authorized licensed use limited to: VIT University. Downloaded on January 25,2023 at 08:37:47 UTC from IEEE Xplore. Restrictions apply.