MinAC: Minimal-Area Approximate Compressor Design Based on Exact Synthesis for Approximate Multipliers

Xuan Wang[1] and Weikang Qian[1,2]
[1]University of Michigan-Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai, China
[2]MoE Key Laboratory of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai, China
Emails: xuan.wang@sjtu.edu.cn, qianwk@sjtu.edu.cn

***Abstract*—Approximate multiplier is a fundamental arithmetic block for designing energy-efficient systems, which can be realized by approximate 4-2 compressors. However, the prior works all design approximate 4-2 compressors manually, so the area optimality of the circuits cannot be guaranteed. In this paper, given any input distribution and error bound, we propose MinAC, an exact synthesis-based method to automatically produce minimal-area approximate 4-2 compressors. To directly obtain an area-optimal circuit using an industrial gate library, gates with multiple outputs are taken into account during the exact synthesis. The experimental results show that compared with the existing methods, MinAC on average can produce approximate 4-2 compressors with 39.8%, 44.2%, and 7.9% reduction in area-delay-product, power-delay-product, and mean error distance, respectively. The code of MinAC is available at https://github.com/SJTU-ECTL/MinAC.**

***Index Terms*—approximate mulitplier, approximate 4-2 compressor, exact synthesis, satisfiability modulo theories (SMT).**

I. Introduction

Approximate computing, an emerging computing paradigm, introduces slight imprecision in computation to improve circuit area, speed, and power consumption [1]. It has been successfully applied in many error-tolerant and computation-intensive applications, such as image processing and deep learning [2]. As one of the fundamental arithmetic units of many systems, multiplier costs much area and power due to its complex logic blocks. Thus, more and more attention has been paid to the design of approximate multipliers.

The approximation can be introduced in the multipliers in various ways. In [3], [4], some partial products are omitted to construct truncated multipliers. The works [5], [6] use approximate 2×2 or 4×4 multipliers to recursively build approximate $n \times n$ multipliers. Since the partial product reduction consumes the most area in a multiplier, much effort has been put in the design of approximate compressors to further reduce the hardware cost of approximate multipliers.

The most commonly used compressor is the full adder, which acts as a 3-2 compressor. Compared to 3-2 compressors, 4-2 compressors have been widely studied due to its higher compression efficiency. Several methods have been proposed to design approximate 4-2 compressors [6]–[14]. The works [6], [10] propose new approximate 4-2 compressors to build power-efficient approximate multipliers, and error detection and correction modules are designed to improve circuit accuracy. In [7]–[9], [12], by modifying the truth table of the exact compressors, different kinds of approximate 4-2 compressors are proposed to build approximate Dadda multipliers. The work [11] proposes four dual-quality approximate 4-2 compressors, which can flexibly switch between the exact and the approximate operating modes. The work [13] proposes two new 4-2 compressors using the concept of the stacking circuit. In [14], a novel method is proposed to design a new class of approximate compressors, whose outputs are equally weighted. However, existing approximate 4-2 compressors are all designed manually. Thus, given a required error bound, the area optimality of approximate 4-2 compressors cannot be guaranteed.

This work is supported by the National Key R&D Program of China under Grant 2020YFB2205501. Corresponding author: Weikang Qian.

To reach the optimal solution, we apply exact synthesis to automatically produce approximate 4-2 compressors in this work. Given a Boolean function, exact synthesis approach can find an optimum Boolean network (e.g., in size or depth) [15]. For traditional logic circuits, several previous works apply exact synthesis to obtain the gate-number-optimal circuits [15]–[17]. In addition, the work [18] designs a novel exact synthesis method to obtain the area-optimal circuits for stochastic circuits, aiming to eliminate the gap between technology-independent synthesis and technology mapping. Despite the progress, there still lacks an exact synthesis method that can handle gates with multiple outputs existing in industrial gate libraries (e.g., 1-bit full adder and half adder). Thus, the obtained circuits by existing exact synthesis methods may not be area-optimal when an industrial gate library is used.

In this paper, to address the aforementioned challenges, given any input distribution and error bound, we propose MinAC, an exact synthesis-based method to obtain minimal-area approximate 4-2 compressors. To directly reach an area-optimal circuit in an industrial gate library, we consider gates with multiple outputs during the exact synthesis, which has never been explored before. The experimental results show that compared with the existing works, MinAC on average reduces the gate number, area, delay, power, area-delay-product (ADP), power-delay-product (PDP), and mean error distance (MED) of the approximate 4-2 compressors by 28.6%, 34.2%, 11.29%, 39.9%, 39.8%, 44.2%, and 7.9%, respectively. The code of MinAC is made open source at https://github.com/SJTU-ECTL/MinAC.

The rest of the paper is organized as follows. Section II provides the background. Section III presents the MinAC method. Section IV shows the experimental results. Section V concludes the paper.

II. Background

In this section, we introduce the background on the conventional exact 4-2 compressor and the exact synthesis.

A. Conventional Exact 4-2 Compressor

Fig. 1 shows a design of the conventional exact 4-2 compressor. It consists of two full-adders and has five primary inputs (PIs), x_1, x_2, x_3, x_4, and C_{in}, and three primary outputs (POs), *Sum*, *Carry*, and C_{out}, where C_{in} comes from the preceding less significant block, and C_{out} goes to the next more significant block. The outputs *Sum*, *Carry*, and C_{out} are calculated as [7]:

$$\begin{aligned}
&\mathit{Sum} = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus C_{in},\\
&\mathit{Carry} = (x_1 \oplus x_2 \oplus x_3 \oplus x_4) \wedge C_{in} + \overline{(x_1 \oplus x_2 \oplus x_3 \oplus x_4)} \wedge x_4,\\
&C_{out} = (x_1 \wedge (x_2 \vee x_3)) \vee (x_2 \wedge x_3).
\end{aligned}$$

B. Exact Synthesis

Given a Boolean function, exact synthesis approach can be applied to find an optimum logic network [15]. Suppose that K-input 1-output gates are used in the circuit, where K is an arbitrary but fixed value. To obtain the optimal circuit for the given function, an exact

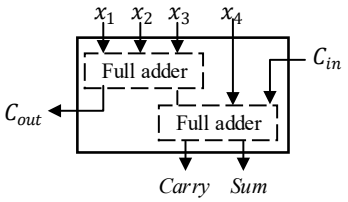

Fig. 1. A design of the conventional exact 4-2 compressor.

synthesis problem with proper encoding variables and constraints can be formulated and solved to determine the connections of the nodes, the gate functions, and the output nodes of the network.

III. Proposed Method

In this section, given an input distribution and an error bound, we present the proposed MinAC method to synthesize an area-optimal approximate 4-2 compressor.

A. Exact Synthesis Flow

The MinAC method is shown in Algorithm 1. It shares similarity with the exact synthesis flow in [18], but with the following main differences: 1) Gates with multiple outputs, which exist in an industrial gate library, are handled; 2) As we find that the circuit with the smallest gate number may not be area-optimal, the circuits with gate number larger than the smallest value are also considered to obtain the area-optimal circuit; 3) The specific requirement for approximate computing is considered.

Algorithm 1: The proposed MinAC method.

Input : the input distribution P, the truth table of the exact 4-2 compressor Y, and an error bound e_b.
Output: an area-optimal approximate 4-2 compressor C_a.
1 $S_g \leftarrow \emptyset$; $S_a \leftarrow \emptyset$; $r \leftarrow 0$;
2 $C_g \leftarrow$ *gateOptimalES*(P, Y, e_b, r);
3 $S_g \leftarrow S_g \cup \{C_g\}$; $r_{\min} \leftarrow C_g.r$;
4 **for** $r \leftarrow r_{\min} + 1$ *to* $r_{\min} + w$ **do**
5 $\quad C_g \leftarrow$ *gateOptimalES*(P, Y, e_b, r); $S_g \leftarrow S_g \cup \{C_g\}$;
6 **foreach** $C_g \in S_g$ **do**
7 $\quad C_a \leftarrow$ *areaOptimalES*(P, Y, e_b, C_g); $S_a \leftarrow S_a \cup \{C_a\}$;
8 **return** *the circuit* C_a^* *with the smallest area in* S_a;

The inputs of Algorithm 1 are the input distribution P, the truth table of the exact 4-2 compressor Y, and an error bound e_b. The error metrics can be any error measures, such as error rate (ER), MED, and mean relative error distance (MRED). The output is an area-optimal approximate 4-2 compressor. In Algorithm 1, set S_g stores the circuits with $r_{\min}, \ldots, r_{\min} + w$ gates, where $r_{\min}$ is the smallest gate number for realizing the given target and w is a non-negative integer parameter. Set S_a holds the area-optimal circuits with r gates for $r_{\min} \le r \le r_{\min} + w$. Line 2 calls the function *gateOptimalES*, which takes the gate number $r = 0$ as a starting point for search, to obtain a gate-number-optimal circuit C_g. Line 3 adds C_g to S_g and sets $r_{\min}$ as the gate number r of C_g. Then, for each $r_{\min} + 1 \le r \le r_{\min} + w$, Line 5 continues to call the function *gateOptimalES* with a new starting point for search as r to obtain a new circuit C_g, and adds it to S_g. For each C_g in the set S_g, Line 7 calls the function *areaOptimalES*, which takes C_g as a starting design for search, to obtain an area-optimal circuit C_a with the same gate number as C_g, and adds C_a to S_a. Line 8 chooses the circuit C_a^* with the smallest area from the set S_a and returns it.

The function *gateOptimalES* loops over the gate number R, which is initialized as the given starting value r. For each gate number R, we formulate a satisfiability modulo theories (SMT)-based exact synthesis problem of whether there exists a circuit C_g with R gates satisfying the error bound e_b. It is realized by adding the encoding variables and constraints, which will be described in Sections III-B2–III-B7. Once the SMT solver returns SAT, the loop terminates and a circuit C_g with the smallest gate number that is at least r is found. Otherwise, the gate number R is increased by 1 to continue the loop.

The function *areaOptimalES* loops over the area A, which is initialized as the area of the given starting design C_g. Besides, the gate number of the circuit is fixed as the gate number of C_g. Then, for each area A, we formulate an SMT-based problem of whether there exists a circuit C_a with area no more than A satisfying the error bound e_b. It is realized by adding the above encoding variables and constraints, and an additional area constraint to be described in Section III-B8. If the SMT solver returns SAT, we set the area A as a value smaller than the area of C_a to continue the loop. Otherwise, the loop terminates and an area-optimal circuit C_a is found.

B. SMT Problem Formulation

MinAC uses multiple selection variable encoding [15]. It requires all the gates in the circuit to have the same input size and output size. We set the input size of the gates in the circuit, K, as the maximum input number of the gates in a given library. To handle gates with multiple outputs in an industrial library, we also set the output size of the gates in the circuit, B, as the maximum output number of the gates in the library. In order to map a gate in the circuit to a gate in the library, we need to extend each gate in the library with the input number less than K or the output number less than B to one with K inputs and B outputs. We first propose a method to do so. Then, we will describe the details of encoding variables and constraints. We illustrate our method with $K = 3$ and $B = 2$.

1) Extending the Gates in the Library: For gates in the library with the input number less than 3 or the output number less than 2, we propose a method to extend them to 3-input 2-output gates. An example of a 2-input NAND gate, NAND2, is used to describe this method.

Fig. 2(a) shows the original truth table of NAND2 represented as a row vector.[1] First, we perform input extension, i.e., extending it to a gate with 3 inputs by introducing 1 additional fake input [18]. We need to select 2 from the 3 inputs as the real inputs. Thus, there are in total $\binom{3}{2} = 3$ NAND2s with 3 inputs. Their truth tables are shown in Fig. 2(b), where NAND2-ij denotes that its i-th and j-th inputs are the real inputs. Then, for each NAND2 gate with 3 inputs, we perform output extension, i.e., extending it to a gate with 2 outputs by introducing 1 redundant output. The truth table of this redundant output is copied from that of the real output. Fig. 2(c) shows the truth table of NAND2-12 with 3 inputs and 2 outputs. Generally, if we want to extend a gate with M outputs to N outputs, we need to introduce $(N - M)$ redundant outputs. We set the truth tables of the $(N - M)$ redundant outputs the same as that of the first real output.

	NAND2-12: 11111100	NAND2-12:
NAND2: 1110	NAND2-13: 11111010	11111100 (output 1)
	NAND2-23: 11101110	11111100 (output 2)
(a)	(b)	(c)

Fig. 2. Truth tables represented as row vectors of the NAND2 gate: (a) original; (b) with input extension; (c) with both input and output extension.

[1]A truth table with k inputs in the order of $x_1, x_2, \ldots, x_k$ is represented as a row vector of 2^k entries, where entry i $(0 \le i < 2^k)$ gives the output when $x_1, \ldots, x_k$ encode the binary value i, i.e., $(x_1 \ldots x_k)_2 = i$.

We apply the above method to each gate in the given library. Then, we obtain an *extended gate library* with each gate in it of K inputs and B outputs. We denote its size as L.

2) Encoding Variables: Suppose that there are n PIs and m POs in the circuit. In this work, we consider the design of the approximate 4-2 compressor. Thus, $n = 4$ and $m = 2$. Besides, assume that the gate number of the circuit is r. The PIs are denoted as $x_0, \ldots, x_{n-1}$ and the gates are denoted as $x_n, \ldots, x_{n+r-1}$. The maximum output number M_i of node x_i $(0 \leq i < n + r)$ is defined as:

$$M_i = \begin{cases} 1, & 0 \leq i < n \text{ (for PIs)}, \\ B, & n \leq i < n + r \text{ (for gates)}. \end{cases}$$

Then, we define the following encoding variables:

- $x_{i\alpha t}$: the t-th bit in the global truth table of the α-th output of node x_i, where $0 \leq i < n + r$, $0 \leq \alpha < M_i$, and $0 \leq t < 2^n$.
- $g_{hi\alpha}$: whether the h-th PO connects with the α-th output of node x_i, where $0 \leq h < m$, $0 \leq i < n + r$, and $0 \leq \alpha < M_i$.
- o_{ht}: the t-th bit of the h-th PO, where $0 \leq h < m$ and $0 \leq t < 2^n$.
- $s_{ij\alpha}$: whether gate x_i has an input from the α-th output of node x_j, where $n \leq i < n + r$, $0 \leq j < i$, and $0 \leq \alpha < M_j$.
- $f_{i\alpha pqu}$: the value of the α-th output of gate x_i under the local input combination (p, q, u), where $n \leq i < n+r$, $0 \leq \alpha < M_i$, and $p, q, u \in \{0, 1\}$.
- v_{il}: whether gate x_i is the l-th gate in the extended library, where $n \leq i < n + r$ and $0 \leq l < L$.

3) Connection Constraint: Different from the constraints for gates with single output in [15]–[18], we formulate the connection constraint for gates with multiple outputs as follows, which ensures the circuit outputs the correct results.

$$\begin{aligned} &\bar{s}_{ij\beta} \vee \bar{s}_{ik\gamma} \vee \bar{s}_{il\delta} \vee (x_{i\alpha t} \oplus a) \vee (x_{j\beta t} \oplus b) \vee (x_{k\gamma t} \oplus c) \\ &\vee (x_{l\delta t} \oplus d) \vee (f_{i\alpha bcd} \oplus \bar{a}) = 1. \end{aligned} \tag{1}$$

It must be satisfied for all $n \leq i < n + r$, $0 \leq j \leq k \leq l < i$, $0 \leq \alpha < M_i$, $0 \leq \beta < M_j$, $0 \leq \gamma < M_k$, $0 \leq \delta < M_l$, $0 \leq t < 2^n$, and $a, b, c, d \in \{0, 1\}$. Eq. (1) means that if the three inputs of gate x_i are the β-th output of x_j, the γ-th output of x_k, the δ-th output of x_l, their t-th bits of the global truth tables are b, c, d, respectively, and the t-th bit of the global truth table of the α-th output of gate x_i is a, then the α-th output of x_i under the local input combination (b, c, d) must be a. Note that two of the three inputs of gate x_i may come from the two outputs of another same gate g, so the conditions $j = k$ and $k = l$ can happen. However, if they occur, the two inputs of gate x_i must come from two *different* outputs of g. Thus, when $j = k$, we require $\beta \neq \gamma$, and when $k = l$, we require $\gamma \neq \delta$.

4) Input Number Constraint: Each gate should have three inputs. Thus, Eq. (2) must hold for all $n \leq i < n + r$.

$$\sum_{j=0}^{i-1} \sum_{\alpha=0}^{M_j-1} s_{ij\alpha} = 3. \tag{2}$$

5) Gate Index Constraint: Since each gate x_i should be mapped to exactly one gate in the extended library, Eq. (3) must hold for $n \leq i < n + r$.

$$\sum_{l=0}^{L-1} v_{il} = 1. \tag{3}$$

In addition, if gate x_i is mapped to the l-th gate in the extended library, the local truth table of x_i must equal the truth table of the l-th gate in the library. Denote $\psi_{l\alpha pqu}$ as the value of the α-th output of the l-th gate in the extended library under the input combination (p, q, u). Then, Eq. (4) must hold for all $n \leq i < n+r$, $0 \leq l < L$, $0 \leq \alpha < M_i$, and $p, q, u \in \{0, 1\}$.

$$\bar{v}_{il} \vee (\bar{f}_{i\alpha pqu} \oplus \psi_{l\alpha pqu}) = 1. \tag{4}$$

6) Output Constraint: As each PO is given by an output of a node in the circuit, Eq. (5) must hold for all $0 \leq h < m$.

$$\sum_{i=0}^{n+r-1} \sum_{\alpha=0}^{M_i-1} g_{hi\alpha} = 1. \tag{5}$$

Furthermore, if PO h connects to the α-th output of gate x_i, the global truth table of x_i must equal the truth table of PO h. Thus, Eq. (6) must hold for all $0 \leq h < m$, $0 \leq i < n + r$, $0 \leq \alpha < M_i$, and $0 \leq t < 2^n$.

$$\bar{g}_{hi\alpha} \vee (\bar{x}_{i\alpha t} \oplus o_{ht}) = 1. \tag{6}$$

7) Error Constraint: We take MED metric as an example. Assume that the probability of the t-th $(0 \leq t < 2^n)$ input combination is P_t. The MED of the approximate circuit cannot exceed the given error bound e_b, as shown in Eq. (7).

$$\sum_{t=0}^{2^n-1} \left| \sum_{h=0}^{m-1} 2^h o_{ht} - Y_t \right| P_t \leq e_b, \tag{7}$$

where $\sum_{h=0}^{m-1} 2^h o_{ht}$ and Y_t are the approximate output and the accurate output for the t-th input combination, respectively.

8) Area Constraint: This constraint is applied to the function *areaOptimalES* to check whether there exists a circuit with area no more than A. Suppose that the area of the l-th $(0 \leq l < L)$ gate in the extended library is a_l. To ensure that the area of the whole circuit is no more than A, Eq. (8) must hold.

$$\sum_{i=n}^{n+r-1} \sum_{l=0}^{L-1} a_l v_{il} \leq A. \tag{8}$$

IV. Experimental Results

In this section, we show the experimental results of MinAC. SMT solver z3 [19] is applied to solve the exact synthesis problem. We further synthesize the circuits using Synopsys Design Compiler [20] with Nangate 45nm technology [21] and measure their hardware costs. The inputs are set as uniformly distributed. The parameter w in Algorithm 1 is set as 2. In Nangate 45nm library, the maximum output number of gates is 2. To speed up the solving process, we only consider gates with no more than 4 inputs. However, our method can also be applied to the gates with more than 4 inputs.

A. Approximate 4-2 Compressors

In this section, we compare the approximate 4-2 compressors obtained by MinAC with 11 state-of-the-art existing ones, including Momeni [7], Venka [8], Yang1 [9], Yang2 [9], Yang3 [9], Lin [6], Ha [10], Akbar1 [11], Akbar2 [11], Ahma [12], and Strollo [13]. For each existing design, we calculate its MED e_b. Then, MinAC is applied to synthesize approximate 4-2 compressors with MED no more than e_b. For three groups of existing designs (i.e., 1st group with Yang1, Lin, and Strollo, 2nd group with Venka and Akbar2, and 3rd group with Yang3 and Ha), although the designs in each group have different MEDs, MinAC obtains the same approximate 4-2 compressors for them. Thus, there are 7 corresponding approximate 4-2 compressors obtained by MinAC in total.

Fig. 3 shows the area-MED and the power-MED plots for the approximate 4-2 compressors. The black dashed line represents the Pareto-optimal frontier, where the Pareto-optimal points are all given by MinAC. Thus, the MinAC designs outperform the existing ones. Table I further lists the average hardware cost and the average MED of the existing designs and the MinAC designs, together with the

reduction ratio of the MinAC designs over the existing ones in parentheses. As shown in Table I, MinAC designs have a significant reduction over the existing ones in the average gate number, area, power, ADP, and PDP. Note that the average MED of MinAC designs also reduces. The reason is that MinAC produces some designs with a smaller MED than their counterparts.

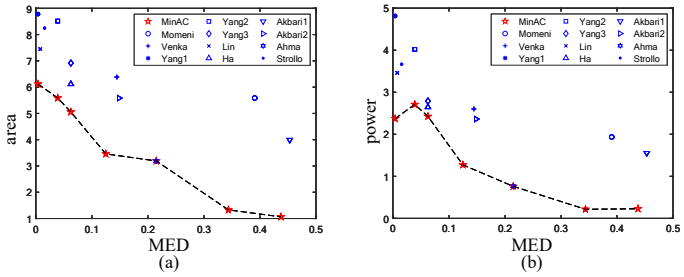

Fig. 3. The (a) area-MED and (b) power-MED plots for the approximate 4-2 compressors.

Table I. The average hardware cost and error comparison between the existing approximate 4-2 compressors and those synthesized by MinAC.

design	gate number	area (μm^2)	delay (ns)	power (μW)	ADP ($\mu m^2 \cdot ns$)	PDP ($\mu W \cdot ns$)	MED
existing	5.6	6.432	0.248	2.78	1.662	0.730	0.140
MinAC	4.0 (-28.6%)	4.232 (-34.2%)	0.220 (-11.29%)	1.67 (-39.9%)	1.001 (-39.8%)	0.407 (-44.2%)	0.129 (-7.9%)

B. Application: Approximate 8×8 Multipliers

In this section, we study the performance of approximate 8×8 multipliers. For each approximate 4-2 compressor in Section IV-A, we construct its corresponding approximate 8×8 multiplier. Thus, there are 11 and 7 approximate 8×8 multipliers realized by the existing and MinAC approximate 4-2 compressors, respectively.

To construct an approximate 8×8 multiplier, first, the two-input AND gates are used to generate the partial product array (PPA). Then, we perform PPA reduction. As shown in Fig. 4, the PPA is divided into two parts: the accurate part (the 7 most significant columns) and the approximate part (the 8 least significant columns). The approximate 4-2 compressor is only used in the approximate part. We determine the connection order of the approximate compressors using the method in [13]. Finally, a Kogge-Stone adder is applied to compute the final result. We perform exhaustive simulations for the approximate 8×8 multipliers to obtain their error metrics, including ER, MED, and MRED.

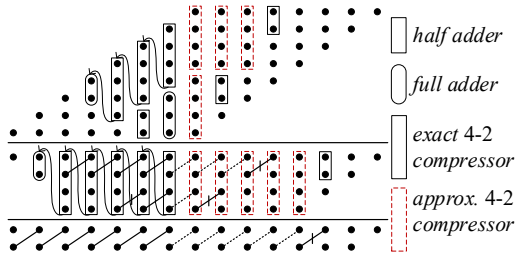

Fig. 4. A reduction scheme of an approximate 8×8 multiplier. The detailed meaning of the symbols and the lines can be found in [13].

Fig. 5 shows the area/power-vs.-ER/MED/MRED plots for the approximate 8×8 multipliers. The black dashed line represents the Pareto-optimal frontier. As shown in Fig. 5, for all the trade-offs between the hardware cost and the error performance, the Pareto-optimal points are all given by MinAC. This shows the advantage of building approximate multipliers with the approximate compressors synthesized by MinAC.

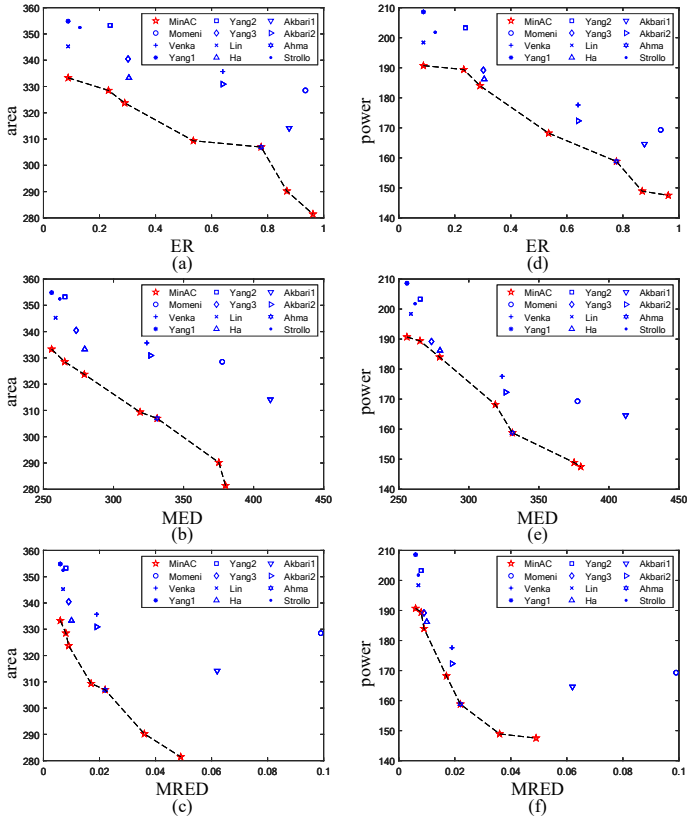

Fig. 5. The (a) area-ER, (b) area-MED, (c) area-MRED, (d) power-ER, (e) power-MED, and (f) power-MRED plots for the approximate 8×8 multipliers.

Table II lists the average hardware cost and error measures of the approximate multipliers, together with the reduction ratio of MinAC designs over the existing ones in parentheses. For all the metrics, MinAC designs are better than the existing ones. The MRED of MinAC is reduced by 33.3%, while the other metrics show less reduction. This is because we only use approximate 4-2 compressors in the 8 least significant columns of PPA, which only occupy a small portion of the approximate 8×8 multiplier.

Table II. The average hardware cost and error comparison between the approximate 8×8 multipliers with the existing approximate 4-2 compressors and those with the compressors synthesized by MinAC.

design	area (μm^2)	power (μW)	ADP ($\mu m^2 \cdot ns$)	PDP ($\mu W \cdot ns$)	ER (%)	MED	MRED
existing	336.0	184.6	329.3	180.9	45.6	305.9	0.024
MinAC	315.7 (-6.0%)	174.6 (-5.4%)	309.1 (-6.1%)	171.0 (-5.5%)	43.2 (-5.3%)	301.3 (-1.5%)	0.016 (-33.3%)

V. Conclusion

In this work, we propose MinAC, a novel exact synthesis-based method to obtain minimal-area approximate 4-2 compressors, which considers gates with multiple outputs in an industrial gate library. MinAC can obtain approximate 4-2 compressors with smaller ADP, PDP, and MED than the state-of-the-art designs, which further leads to approximate multipliers with smaller hardware cost and error. In the future work, we will explore how to apply the approximate 4-2 compressors with different hardware costs and errors to the PPA to further reduce the hardware cost of the approximate multipliers.

REFERENCES

[1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *European Test Symposium*, 2013, pp. 1–6.

[2] Q. Xu, T. Mytkowicz, and N. S. Kim, "Approximate computing: A survey," *IEEE Design & Test*, vol. 33, no. 1, pp. 8–22, 2015.

[3] N. Petra, D. D. Caro, V. Garofalo, E. Napoli, and A. G. M. Strollo, "Design of fixed-width multipliers with linear compensation function," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 5, pp. 947–960, 2010.

[4] D. D. Caro, N. Petra, A. G. M. Strollo, F. Tessitore, and E. Napoli, "Fixed-width multipliers and multipliers-accumulators with min-max approximation error," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 9, pp. 2375–2388, 2013.

[5] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power with an underdesigned multiplier architecture," in *Internatioal Conference on VLSI Design*, 2011, pp. 346–351.

[6] C.-H. Lin and I.-C. Lin, "High accuracy approximate multiplier with error correction," in *International Conference on Computer Design*, 2013, pp. 33–38.

[7] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 984–994, 2014.

[8] S. Venkatachalam and S.-B. Ko, "Design of power and area efficient approximate multipliers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 5, pp. 1782–1786, 2017.

[9] Z. Yang, J. Han, and F. Lombardi, "Approximate compressors for error-resilient multiplier design," in *International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, 2015, pp. 183–186.

[10] M. Ha and S. Lee, "Multipliers with approximate 4-2 compressors and error recovery modules," *IEEE Embedded Systems Letters*, vol. 10, no. 1, pp. 6–9, 2017.

[11] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 4, pp. 1352–1361, 2017.

[12] M. Ahmadinejad, M. H. Moaiyeri, and F. Sabetzadeh, "Energy and area efficient imprecise compressors for approximate multiplication at nanoscale," *AEU-International Journal of Electronics and Communications*, vol. 110, p. 152859, 2019.

[13] A. G. M. Strollo, E. Napoli, D. D. Caro, N. Petra, and G. D. Meo, "Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 9, pp. 3021–3034, 2020.

[14] D. Esposito, A. G. M. Strollo, D. D. C. E. Napoli, and N. Petra, "Approximate multipliers based on new approximate compressors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4169–4182, 2018.

[15] W. Haaswijk *et al.*, "SAT-based exact synthesis: encodings, topology families, and parallelism," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 4, pp. 871–884, 2020.

[16] M. Soeken *et al.*, "Practical exact synthesis," in *Design, Automation & Test in Europe Conference & Exhibition*, 2018, pp. 309–314.

[17] ——, "Exact synthesis of majority-inverter graphs and its applications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 11, pp. 1842–1855, 2017.

[18] X. Wang, Z. Chu, and W. Qian, "MinSC: An exact synthesis-based method for minimal-area stochastic circuits under relaxed error bound," in *International Conference on Computer-Aided Design*, 2021.

[19] L. D. Moura and N. Bjørner, "Z3: An efficient SMT solver," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2008, pp. 337–340.

[20] "Synopsys design compiler," 2012. [Online]. Available: http://www.synopsys.com

[21] "Nangate 45nm open cell library," 2008. [Online]. Available: http://www.nangate.com/