TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 07/10 pp729–740
DOI: 10.26599/TST.2020.9010032
Volume 27, Number 4, August 2022

Accurate Reliability Analysis Methods for Approximate Computing Circuits

Zhen Wang, Guofa Zhang, Jing Ye*, Jianhui Jiang*, Fengyong Li, and Yong Wang

Abstract: In recent years, Approximate Computing Circuits (ACCs) have been widely used in applications with intrinsic tolerance to errors. With the increased availability of approximate computing circuit approaches, reliability analysis methods for assessing their fault vulnerability have become highly necessary. In this study, two accurate reliability evaluation methods for approximate computing circuits are proposed. The reliability of approximate computing circuits is calculated on the basis of the iterative Probabilistic Transfer Matrix (PTM) model. During the calculation, the correlation coefficients are derived and combined to deal with the correlation problem caused by fanout reconvergence. The accuracy and scalability of the two methods are verified using three sets of approximate computing circuit instances and more circuits in EvoApprox8b, which is an approximate computing circuit open source library. Experimental results show that relative to the Monte Carlo simulation, the two methods achieve average error rates of 0.46% and 1.29% and time overheads of 0.002% and 0.1%. Different from the existing approaches to reliability estimation for approximate computing circuits based on the original PTM model, the proposed methods reduce the space overheads by nearly 50% and achieve time overheads of 1.78% and 2.19%.

Key words: Approximate Computing Circuit (ACC); correlation coefficient; iterative Probabilistic Transfer Matrix (PTM); reliability

1 Introduction

With the continuous advancement of integrated circuit design technology, the existing feature size has reached the physical limit. Dennard's scaling law and Moore's law have also become difficult to maintain. Meanwhile, reducing power consumption to continuously meet the requirements of high-performance computing has emerged as a meaningful task. In many traditional and emerging applications, such as multimedia processing, digital signal processing, wireless communication[1–3], image rendering, augmented reality, data mining, computer vision, voice identification, deep neural network, mobile computing, and sensor data analysis[4–9], users can accept output results with certain errors. With the growing popularity of mobile and embedded computing systems, these applications are becoming increasingly widespread. Developed under this context is the Approximate Computing Circuit (ACC)[6, 10]. ACCs take advantage of the inherent error tolerance of a number of applications to balance the accuracy of the output results with the performance of the applications, relaxing 100% accuracy ultimately to reduce power consumption and area overhead. As ACCs meet application requirements while improving performance and energy

• Zhen Wang, Guofa Zhang, Fengyong Li, and Yong Wang are with the School of the Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200090, China. E-mail: wangzhenqq@hotmail.com; 546319599@qq.com; fyli@shiep.edu.cn; wy616@126.com.

• Jing Ye is with Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China. E-mail: yejing@ict.ac.cn.

• Jianhui Jiang is with the School of Software Engineering, Tongji University, Shanghai 200092, China. E-mail: jhjiang@tongji.edu.cn.

∗ To whom correspondence should be addressed.

Manuscript received: 2020-07-29; accepted: 2020-09-03

efficiency, they have increasingly attracted the scientific community. With the physical realization of ACCs, the reliability issues emerging during their design require close attention. One key aspect of today's nanoscale technologies is their vulnerability to process deviations, various defects, and failures. In order to ensure the availability of ACCs, it is necessary to analyze the reliability problem caused by manufacturing defects and failures.

At present, the research related to the reliability analysis of ACCs is immature. Existing studies mainly focus on the analysis of the error characteristics of ACCs to measure the approximation degree. In terms of research objects, most existing methods are oriented toward approximate adders, and a few are oriented toward approximate multipliers. Hence, approaches that do not distinguish the functions of ACCs are limited. As for the attributes of methods, they can be divided into three categories: methods based on Monte Carlo (MC) simulation analysis, methods establishing error characteristic metrics, and methods analyzing error probabilities. In the above works, errors refer to the different functional outputs of ACCs relative to those of corresponding conventional accurate circuits. Herein, reliability is not evaluated assuming that ACCs are faulty.

Some of the functional outputs of ACCs that differ from those of conventional circuits are sometimes acceptable for certain applications. However, little research has been conducted to estimate the probability of obtaining acceptable outputs. At present, only a few studies have explored ACC testing, fault-tolerant design, and failure rate. Herein, ACC reliability is the probability of obtaining an acceptable circuit output under the influence of production defects and operating faults. Calculating reliability can help designers to improve the ACC structure during the design stage[11].

In our previous work[12], we evaluated the reliability of small-scale approximate adders on the basis of the original Probabilistic Transfer Matrix (PTM) method[13]. In this work, we present relatively complete and accurate ACC reliability evaluation methods that are based on the iterative PTM model. Correlation vectors are derived for various basic topologies in a circuit and are then incorporated into the iterative PTM model to calculate ACC reliability. The derivation of the correlation vectors alleviates the correlation problem caused by fanout reconvergence, and the iterative PTM model increases the applicable circuit scale.

The rest of the paper is organized as follows. The existing studies in the field of ACC reliability analysis are introduced in Section 2. Preliminaries are detailed in Section 3. The proposed ACC reliability evaluation methods are discussed in Section 4. The analysis of the comparative verification experiment results and the test results of the proposed methods based on an open source library are provided in Section 5. Relevant conclusions are summarized in Section 6.

2 Related Work

ACC reliability analysis is still in its infancy. Most existing works focus on the analysis of ACC error characteristics. For example, the method for the modeling and analysis of ACCs simulates and analyzes the behavior of ACCs. Through Boolean analysis technology, the MC simulation method is used to obtain the worst error situation, calculate the error probability, and infer the error distribution[14]. This method can be applied to ACCs with different functions and is relatively accurate but time consuming. In this method, the descriptions of error statistics and error distribution are numerical data, which cannot be parameterized according to the circuit design. References [15–17] proposed a series of error characteristic metrics, including error rate, error distance, mean error distance, normalized mean error distance, relative error distance, mean relative error distance, etc. These metrics are widely used to analyze and compare existing ACCs or evaluate newly proposed ACCs, including various approximate adders[17, 18], approximate multipliers[19], and approximate dividers[20]. The methods for calculating error characteristics are mainly oriented toward arithmetic units and are based on statistical measurements; hence, they are difficult to extend to complex circuits, and they do not always reflect the critical degree of errors[21]. Existing error analysis methods are aimed toward evaluating the approximation degree of ACC design, taking the accurate output of conventional circuits as the benchmark, and calculating the deviation range and the rate of the approximate output. However, they cannot directly evaluate the reliability of ACCs. Studies focusing on the reliability of ACC are few. A number of scholars[22, 23] have emphasized the necessity of reliability analysis for ACCs, but their research is not extensive. Reference [24] selected an error-tolerant application, i.e., H.264/AVC decoder, to evaluate fault criticality in different modules and in the same modules. This work is dedicated to specialized applications and 1/0 (stuck-at) hard faults.

Our previous work[25] aimed at soft errors and involved the calculation of the failure probability of ACCs based on the error propagation probability model. This work can apply to different types of ACCs, but it cannot evaluate the overall reliability of ACCs. Reference [12] performed a preliminary exploration of ACC reliability prediction, defined an acceptable output set by an accurate output and an approximate fault-free output, and evaluated ACC reliability accurately on the basis of the original PTM model. However, due to the exponential space-time complexity of the original PTM model, the method can only be used for small-scale ACCs.

In this work, the complexity and correlation issues during ACC reliability evaluation are further studied. The iterative PTM model is used in the ACC reliability calculation, and time-space complexity is linear, which can help to expand the scale of the applied circuit. During the calculation, the correlation coefficients are expanded and derived to solve the correlation problem.

3 Preliminary

In this section, we briefly introduce the preliminary knowledge about the iterative PTM method and the basic theory of the correlation coefficients required by the evaluation methods analyzed in this work.

3.1 Iterative PTM

In conventional circuit reliability evaluation, the PTM[26] is a complete and accurate classic analysis model. In this method, the PTM is established for each primary gate. For example, assume that the probability of NAND2 producing an incorrect output is p; the corresponding PTM is shown in Fig. 1. When $p = 0$, the PTM becomes an Ideal Transfer Matrix (ITM). The PTM and ITM of the whole circuit are obtained according to the gate connection structure. Then, the reliability of the circuit is calculated as

$$R(c) = \sum_{\forall i,j \mid I(i,j)=1} p(i \mid j)p(i) \tag{1}$$

where $p(i)$ is the probability of the input i, $p(i \mid j)$ is the probability of the output j for input i in PTM, and $I(i,j)$ is the probability of the output j for input i in ITM.

$$\begin{array}{c|c} AB & \begin{array}{cc} C=0 & C=1 \end{array} \\ \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} & \begin{bmatrix} p & 1-p \\ p & 1-p \\ p & 1-p \\ 1-p & p \end{bmatrix} \end{array}$$

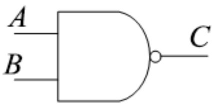


Fig. 1 Probabilistic transfer matrix of NAND2.

Xiao et al.[27] extended the PTM method and proposed an iterative PTM method by using hybrid coding. They divided a circuit into multiple modules according to the logical connection. For every basic circuit module, the input probability matrix, input reliability matrix, probabilistic transfer matrix, output probability matrix, output reliability, etc., are defined. The input probability matrix of a basic module with multiple inputs is equal to the tensor product of its different input probability matrices. Each input probability is equal to the output probability of the last connected gate. Similarly, the input reliability matrix of a multi-input module is the tensor product of the different input reliability matrices. The PTM of the basic module is the probability representation of the corresponding state of the output signals driven by the input signals. Finally, the Hadamard product of a module's PTM and ITM is multiplied by its input probability matrix, and its output probability matrix can be calculated accordingly.

3.2 Correlation coefficient

Assume: event A means that an input signal is equal to 1 in a circuit; event B means that another input signal is equal to 1 in this circuit; their probabilities are denoted as $P(A)$ and $P(B)$, respectively. Then, the probability of event $AB = A \cap B$, indicating the two-input signals equal to 1, can be written as

$$P(AB) = P(A)P(B)C_{A,B} \tag{2}$$

where $C_{A,B}$ is the correlation coefficient defined in Ref. [28]. $C_{A,B} = 1$ means that events A and B are independent; otherwise, $C_{A,B} \neq 1$. $C_{A,B}$ is calculated as

$$C_{A,B} = \frac{P(AB)}{P(A)P(B)} = \frac{P(A|B)}{P(A)} = \frac{P(B|A)}{P(B)} \tag{3}$$

Reference [28] discussed several basic topological structures in a circuit only considering the probability that a signal is equal to 1, and provided fundamental calculation methods. In this work, we expand the correlation coefficient theory and introduce it to the reliability evaluation of ACCs to improve the accuracy of results.

4 ACC Reliability Evaluation

4.1 Basic definitions

For ACC, we have the following definitions:

(1) Acceptable value: The ACC output value that meets the requirements;

(2) Approximate value (denoted as AV): The output value of the fault-free ACC;

(3) Accurate value (denoted as V): The output value of the corresponding fault-free conventional circuit;

(4) Maximum deviated value (denoted as MV): The acceptable value having maximum deviation from AV under every input combination in the ACC, with the direction of deviation approaching the accurate value;

(5) Threshold range: The range of acceptable values of ACC.

The definitions are shown in Fig. 2. In this study, we set MV $= V$, i.e., $[\mathrm{Min}(\mathrm{AV}, V), \mathrm{Max}(\mathrm{AV}, V)]$ as the threshold range. As for the threshold range, the interval of $[V - |V - \mathrm{AV}|, V + |V - \mathrm{AV}|]$ is suitable for some cases, but it cannot be used in nonlinear applications.

For example, if the output of an ACC controls a high pass filter to allow the passing of signal components that are higher than a certain frequency, then the output ranges $[V - |V - \mathrm{AV}|, V]$ and $[V, V + |V - \mathrm{AV}|]$ exert different effects. Therefore, we seek to improve the accuracy by applying the interval $[\mathrm{Min}(\mathrm{AV}, V), \mathrm{Max}(\mathrm{AV}, V)]$ as the threshold range of an ACC.

4.2 Correlation coefficient derivation

To resolve the correlation problem caused by fanout reconvergence, we initially derive the correlation coefficient.

We define four correlation coefficients, namely, $C_{i=0,j=0}$, $C_{i=0,j=1}$, $C_{i=1,j=0}$, and $C_{i=1,j=1}$ for every pair of nodes (i, j) in the circuit. For a two-input primary gate, the correlation coefficient of the outputs can be derived from the input signal probabilities and correlation coefficients. Taking the XOR gate as an example, we present the derivation process. As shown in Fig. 3, i, j, and h are the input signals, l and m are the output signals, we defines three events: $F = \{l = 0\}, D = \{i = 0, j = 0\}$, and $E = \{i = 1, j = 1\}$. Events D and E are mutually exclusive, hence, the equation is $P(F) = P(D \cup E) = P(D) + P(E)$, i.e.,

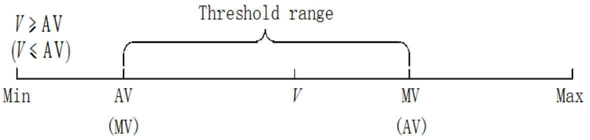


Fig. 2 Threshold range of an approximate circuit.

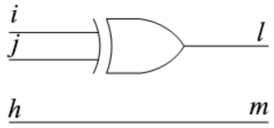


Fig. 3 XOR gate connection structure.

$$P_{l=0} = P_{i=0,j=0} + P_{i=1,j=1} \tag{4}$$

where $P_{l=0}$ is the probability of $l = 0$ and $P_{i=0,j=0}$ is the probability that inputs i and j are 0. According to Eq. (3), Eq. (4) can be written as

$$P_{l=0} = P_{i=0}P_{j=0}C_{i=0,j=0} + P_{i=1}P_{j=1}C_{i=1,j=1} \tag{5}$$

By combining Fig. 3 and Eqs. (4) and (5), we have

$$\begin{aligned} P_{l=0|m=0} = P_{(i=0,j=0)|h=0} + P_{(i=1,j=1)|h=0} = \\ P_{i=0|h=0}P_{i=0|h=0}C_{i=0,j=0} + \\ P_{i=1|h=0}P_{j=1|h=0}C_{i=1,j=1} \end{aligned} \tag{6}$$

From Eq. (3), we can rewrite Eq. (6) as

$$\begin{aligned} P_{l=0}C_{l=0,m=0} = P_{i=0}C_{i=0,h=0}P_{j=0}C_{j=0,h=0}C_{i=0,j=0} + \\ P_{i=1}C_{i=1,h=0}P_{j=1}C_{j=1,h=0}C_{i=1,j=1} \end{aligned} \tag{7}$$

By substituting $P_{l=0}$ in Eq. (7) with Eq.. (5), we have

$$\begin{aligned} C_{l=0,m=0} = (P_{i=0}C_{i=0,h=0}P_{j=0}C_{j=0,h=0}C_{i=0,j=0} + \\ P_{i=1}C_{i=1,h=0}P_{j=1}C_{j=1,h=0}C_{i=1,j=1}) / \\ P_{i=0}P_{j=0}C_{i=0,j=0} + P_{i=1}P_{j=1}C_{i=1,j=1} \end{aligned} \tag{8}$$

For other three correlation coefficients of (l, m), the derivations are similar. Then we can obtain the following:

$$\begin{aligned} C_{l=0,m=1} = (P_{i=0}C_{i=0,h=1}P_{j=0}C_{j=0,h=1}C_{i=0,j=0} + \\ P_{i=1}C_{i=1,h=1}P_{j=1}C_{j=1,h=1}C_{i=1,j=1}) / \\ P_{i=0}P_{j=0}C_{i=0,j=0} + P_{i=1}P_{j=1}C_{i=1,j=1} \end{aligned} \tag{9}$$

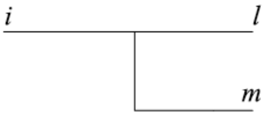
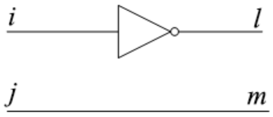
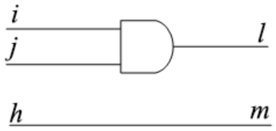
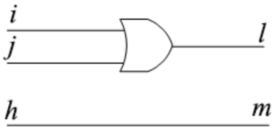
$$\begin{aligned} C_{l=1,m=0} = (P_{i=0}C_{i=0,h=0}P_{j=1}C_{j=1,h=0}C_{i=0,j=1} + \\ P_{i=1}C_{i=1,h=0}P_{j=0}C_{j=0,h=0}C_{i=1,j=0}) / \\ P_{i=0}P_{j=1}C_{i=0,j=1} + P_{i=1}P_{j=0}C_{i=1,j=0} \end{aligned} \tag{10}$$

$$\begin{aligned} C_{l=1,m=1} = (P_{i=0}C_{i=0,h=1}P_{j=1}C_{j=1,h=1}C_{i=0,j=1} + \\ P_{i=1}C_{i=1,h=1}P_{j=0}C_{j=0,h=1}C_{i=1,j=0}) / \\ P_{i=0}P_{j=1}C_{i=0,j=1} + P_{i=1}P_{j=0}C_{i=1,j=0} \end{aligned} \tag{11}$$

The correlation coefficients of the other primary gates are presented in Table 1.

Current discussions about correlation coefficients mainly focus on two-input primary gates. A multi-input gate can be converted into a cascade of two-input gates. Figure 4 shows an example of a four-input AND gate case. In this case, we add a virtual buffer gate at the end of the cascade module and let the fault probability of the

Table 1 Correlation coefficients of different structures with primary gates.

Primary gate	Correlation coefficient
	$C_{l=0,m=0}=\frac{1}{P_{i=0}}$; $C_{l=0,m=1}=0$; $C_{l=1,m=0}=0$; $C_{l=1,m=1}=\frac{1}{P_{i=1}}$
	$C_{l=0,m=0}=\frac{1-P_{i=0}C_{i=0,j=0}}{1-P_{i=0}}$; $C_{l=0,m=1}=\frac{1-P_{i=0}C_{i=0,j=1}}{1-P_{i=0}}$; $C_{l=1,m=0}=\frac{1-P_{i=1}C_{i=1,j=0}}{1-P_{i=1}}$; $C_{l=1,m=1}=\frac{1-P_{i=1}C_{i=1,j=1}}{1-P_{i=1}}$
	$C_{l=0,m=0}=\frac{P_{i=0}C_{i=0,h=0}+P_{j=0}C_{j=0,h=0}-P_{i=0}P_{j=0}C_{i=0,h=0}C_{j=0,h=0}C_{i=0,j=0}}{P_{i=0}+P_{j=0}-P_{i=0}P_{j=0}C_{i=0,j=0}}$; $C_{l=0,m=1}=\frac{P_{i=0}C_{i=0,h=1}+P_{j=0}C_{j=0,h=1}-P_{i=0}P_{j=0}C_{i=0,h=1}C_{j=0,h=1}C_{i=0,j=0}}{P_{i=0}+P_{j=0}-P_{i=0}P_{j=0}C_{i=0,j=0}}$; $C_{l=1,m=0}=C_{i=1,h=0}C_{j=1,h=0}$; $C_{l=1,m=1}=C_{i=1,h=1}C_{j=1,h=1}$
	$C_{l=0,m=0}=C_{i=0,h=0}C_{j=0,h=0}$; $C_{l=0,m=1}=C_{i=0,h=1}C_{j=0,h=1}$; $C_{l=1,m=0}=\frac{P_{i=1}C_{i=1,h=0}+P_{j=1}C_{j=1,h=0}-P_{i=1}P_{j=1}C_{i=1,h=0}C_{j=1,h=0}C_{i=1,j=1}}{P_{i=1}+P_{j=1}-P_{i=1}P_{j=1}C_{i=1,j=1}}$; $C_{l=1,m=1}=\frac{P_{i=1}C_{i=1,h=1}+P_{j=1}C_{j=1,h=1}-P_{i=1}P_{j=1}C_{i=1,h=1}C_{j=1,h=1}C_{i=1,j=1}}{P_{i=1}+P_{j=1}-P_{i=1}P_{j=1}C_{i=1,j=1}}$

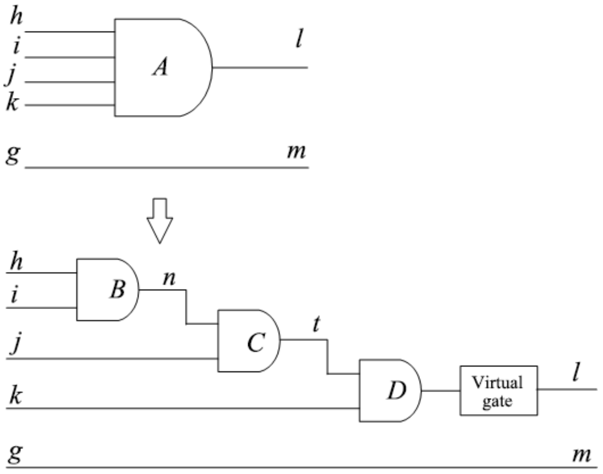


Fig. 4 Conversion of a four-input gate into a cascade module.

buffer gate to be equal to that of the original one. All gates except the last one are free from faults. Then, the correlation coefficients of multi-input gate connection structures can be obtained according to Table 1.

4.3 Evaluation methods considering correlation coefficients based on iterative PTM

For an ACC with n inputs and m outputs, let us assume that each gate has an error probability. The circuit reliability can be expressed as

$$\mathrm{Rel}=\sum_{i=0}^{2^n-1}p_{\mathrm{out}}(i)p_{\mathrm{in}}(i) \tag{12}$$

where $p_{\mathrm{in}}(i)$ is the probability of the input vector whose decimal value equals i and $p_{\mathrm{out}}(i)$ is the probability of the acceptable values under input i. For $p_{\mathrm{out}}(i)$, two calculation methods are given as follows:

Method 1:

$$p_{\text{out}}(i)=\sum p(i,j) \tag{13}$$

Method 2:

$$p_{\text{out}}(i)=\begin{cases} p(i,j), & \text{AV}(i)=V(i);\\ \dfrac{\sum |j-V(i)|p(i,j)}{|\text{AV}(i)-V(i)|}, & \text{AV}(i)\neq V(i)\end{cases} \tag{14}$$

In Methods 1 and 2, AV(i) and $V(i)$ are the approximate value and accurate value under input i, respectively; $p(i,j)$ denotes the probability of the ACC outputting acceptable value j under input i, where j belongs to the threshold range, i.e., $j\in[\text{Min}(\text{AV}(i),V(i)),\text{Max}(\text{AV}(i),V(i))]$.

In Method 1, Eq. (13) is plugged into Eq. (12), and then the ACC reliability is evaluated by the probability of all acceptable values in the threshold range wherein the acceptable values are not treated differently. In Method 2, Eq. (14) is plugged into Eq. (12), with the acceptable value nearest to $V(i)$ having the largest weight. That is to say, Method 1 focuses on the reliability of ACCs under faults, and Method 2 emphasizes the vulnerability of ACCs to faults.

In Eqs. (13) and (14), AV(i) can be obtained according to the truth table of the ACC, and $V(i)$ can be obtained by the arithmetic function of the corresponding conventional circuit. In order to obtain $p(i,j)$, the basic idea of the iterative PTM is applied. We take every primary gate as a basic module. Beginning from every Primary Input (PI) of the ACC, we calculate the reliability matrix of every gate output. According to the connection structure and by taking the matrix as the input reliability matrix, we can then calculate the reliability matrix of the next connected gate output. For such an iteration, the reliability matrices of every Primary Output (PO) can be obtained, and then $p(i,j)$ can be calculated. The correlation coefficients in the calculation are combined.

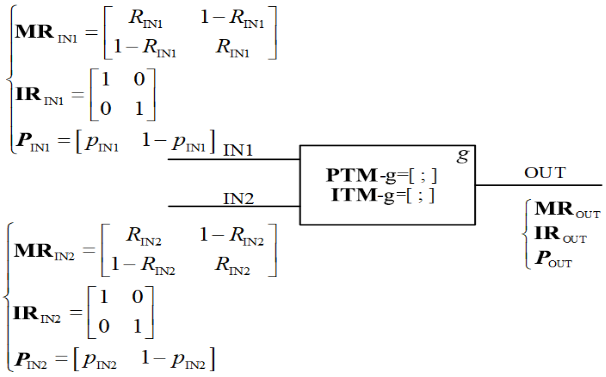

$\mathbf{MR}_{\text{IN1}}=\begin{bmatrix}R_{\text{IN1}} & 1-R_{\text{IN1}}\\ 1-R_{\text{IN1}} & R_{\text{IN1}}\end{bmatrix}$
$\mathbf{IR}_{\text{IN1}}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$
$\boldsymbol{P}_{\text{IN1}}=[p_{\text{IN1}}\quad 1-p_{\text{IN1}}]$
IN1
IN2
g
PTM-g=[;]
ITM-g=[;]
OUT
$\mathbf{MR}_{\text{OUT}}$
$\mathbf{IR}_{\text{OUT}}$
$\boldsymbol{P}_{\text{OUT}}$
$\mathbf{MR}_{\text{IN2}}=\begin{bmatrix}R_{\text{IN2}} & 1-R_{\text{IN2}}\\ 1-R_{\text{IN2}} & R_{\text{IN2}}\end{bmatrix}$
$\mathbf{IR}_{\text{IN2}}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$
$\boldsymbol{P}_{\text{IN2}}=[p_{\text{IN2}}\quad 1-p_{\text{IN2}}]$

Fig. 5 Relevant parameters of a two-input primary gate.

As shown in Fig. 5, IN1 and IN2 are the input nodes of gate g; OUT is the output node of gate g; $\mathbf{MR}_{\text{IN1}}$ and $\mathbf{MR}_{\text{IN2}}$ are the input reliability matrices of IN1 and IN2, respectively, where R_{IN1} and R_{IN2} denote the reliabilities of IN1$=$ 0 and IN2$=$ 0, respectively; $\mathbf{IR}_{\text{IN1}}$ and $\mathbf{IR}_{\text{IN2}}$ are the ideal input reliability matrices of IN1 and IN2, respectively; $\boldsymbol{P}_{\text{IN1}}$ and $\boldsymbol{P}_{\text{IN2}}$ are the input probability matrices, where p_{IN1} and p_{IN2} denote the probabilities of IN1 $=0$ and IN2$=$ 0; **PTM**-g and **ITM**-g are the PTM and ITM of gate g, respectively; and $\boldsymbol{C}_{\text{IN1,IN2}}$ denotes the correlation vectors of IN1 and IN2 and is defined as

$$\boldsymbol{C}_{\text{IN1,IN2}}=[C_{\text{IN1=0,IN2=0}},C_{\text{IN1=0,IN2=1}},\\ C_{\text{IN1=1,IN2=0}},C_{\text{IN1=1,IN2=1}}] \tag{15}$$

Then, the iterative calculation process for this example is given as follows:

(1) Initialize $\boldsymbol{P}_{\text{IN1}},\boldsymbol{P}_{\text{IN2}},\mathbf{MR}_{\text{IN1}}$, and $\mathbf{MR}_{\text{IN2}}$;

(2) Calculate the input probability matrix $\boldsymbol{P}_{\text{IN}}$ of g by the Hadamard product of $\boldsymbol{C}_{\text{IN1,IN2}}$ and the tensor product of $\boldsymbol{P}_{\text{IN1}}$ and $\boldsymbol{P}_{\text{IN2}}$,

$$\boldsymbol{P}_{\text{IN}}=\boldsymbol{P}_{\text{IN1}}\otimes\boldsymbol{P}_{\text{IN2}}\circ\boldsymbol{C}_{\text{IN1,IN2}} \tag{16}$$

(3) Calculate the input reliability matrix $\mathbf{MR}_{\text{IN}}$ of g by the tensor product of $\mathbf{MR}_{\text{IN1}}$ and $\mathbf{MR}_{\text{IN2}}$,

$$\mathbf{MR}_{\text{IN}}=\mathbf{MR}_{\text{IN1}}\otimes\mathbf{MR}_{\text{IN2}} \tag{17}$$

(4) Calculate the ideal input reliability matrix $\mathbf{IR}_{\text{IN}}$ of g by the tensor product of $\mathbf{IR}_{\text{IN1}}$ and $\mathbf{IR}_{\text{IN2}}$,

$$\mathbf{IR}_{\text{IN}}=\mathbf{IR}_{\text{IN1}}\otimes\mathbf{IR}_{\text{IN2}} \tag{18}$$

(5) Calculate the output reliability of g,

$$\boldsymbol{R}_{\text{OUT}}=\boldsymbol{P}_{\text{IN}}\times[(\mathbf{MR}_{\text{IN}}\times\textbf{PTM-g})\circ(\mathbf{IR}_{\text{IN}}\times\textbf{ITM-g})] \tag{19}$$

(6) Initialize the output reliability matrix of g,

$$\mathbf{MR}_{\text{OUT}}=\begin{bmatrix}R_{\text{OUT}} & 1-R_{\text{OUT}}\\ 1-R_{\text{OUT}} & R_{\text{OUT}}\end{bmatrix} \tag{20}$$

(7) Initialize the output probability matrix of g,

$$\boldsymbol{P}_{\text{OUT}}=\boldsymbol{P}_{\text{IN}}\times\textbf{ITM-g} \tag{21}$$

In Step (2), the values of $\boldsymbol{C}_{\text{IN1,IN2}}$ are combined to obtain the accurate signal probabilities, e.g., $\text{Pro}_{(\text{IN1=0,IN2=0})}=\text{Pro}_{(\text{IN1=0})}\times\text{Pro}_{(\text{IN2=0})}\times C_{\text{IN1=0,IN2=0}}$.

The probabilities of g getting outputs 0 and 1 are the 1st and 2nd elements in $\boldsymbol{R}_{\text{OUT}}$, respectively; they can be calculated with Step (5). Steps (6) and (7) are prepared for the next iteration. That is to say, after assigning value i and taking the iterative calculation, we can calculate $p(i,j)$ in Eqs. (13) and (14) according to $\boldsymbol{R}_{\text{OUT}}$ of the primary outputs.

4.4 Algorithm implementation

In this section, we describe the algorithms for calculating ACC reliability based on the iterative PTM with consideration of signal correlation. The complete procedure is shown in Algorithm 1.

We explain Algorithm 1 as follows. At the beginning, the ACC's information is derived by parsing the circuit netlist (see Line 1). For Line 3, calculating the correlation vectors for all the gates is not necessary, and thus, the procedure for determining the correlated gates is indicated in Algorithm 2. The **ITM**-g of every type of primary gate is prepared beforehand. Then, the corresponding **PTM**-g can be obtained by the given fault probability p and **ITM**-g. This procedure explains Line 14. In Line 21, NPI is the number of PIs of the ACC. In this step, AV(i) can be derived according to the ACC structure, and $V(i)$ can be calculated by the accurate function of the conventional circuit corresponding to the ACC. Take the adder circuit as an example, for AV(i) and $V(i)$, the rules are designed for inputting the order of PI and PO tags, i.e., inputting PO tags as $C_n S_n S_{n-1} \cdots S_0$ and PI tags as $X_n Y_n X_{n-1} Y_{n-1} \cdots X_0 Y_0 C_0$. According to the order of the PI tags, we convert the PI vectors into decimals X and Y. Then, we calculate the value of $V(i)$ by adding X and Y. According to the order of the PO tags, the AV(i) vector derived from the circuit structure can be converted into a decimal reasonably.

Algorithm 1 ACC reliability estimation

```
Input: ITM-g and fault probability $p$ of every primary gate $g$,
       ACC gate-level netlist
Output: ACC reliability Rel
1: Parse netlist and divide the multi-input gates into cascade modules;
2: Separate ACC levels;
3: Search correlated gates, see detail in Algorithm 2;
4: for every $i$-th PI vector do
5:     Initialize $\boldsymbol{P}_{\mathrm{IN1}}$, $\mathbf{MR}_{\mathrm{IN1}}$, $\mathbf{IR}_{\mathrm{IN1}}$, $\boldsymbol{P}_{\mathrm{IN2}}$, $\mathbf{MR}_{\mathrm{IN2}}$, and $\mathbf{IR}_{\mathrm{IN2}}$
         for every PI node;
6:     for every $j$-th level of the ACC do
7:         Use $\boldsymbol{P}_{\mathrm{OUT}}$ in Level $j$–1 to set value for $\boldsymbol{P}_{\mathrm{IN1}}$ and $\boldsymbol{P}_{\mathrm{IN2}}$,
           use $\mathbf{MR}_{\mathrm{OUT}}$ in Level $j$–1 to set value for $\mathbf{MR}_{\mathrm{IN1}}$ and
           $\mathbf{MR}_{\mathrm{IN2}}$ for gates in Level $j$;
8:         if there exist correlated gates in the $j$-th level then
9:             calculate correlation vectors for the related
                 connection parts, see detail in Algorithm 3;
10:            Calculate $\boldsymbol{P}_{\mathrm{IN}}$ considering correlation vector;
11:        else
12:            Calculate $\boldsymbol{P}_{\mathrm{IN}}$ without correlation vector;
13:        end if
14:        Overload ITM-g, and then obtain PTM-g by $p$ for
             every gate $g$ in the $j$-th level;
15:        Calculate $\mathbf{MR}_{\mathrm{IN}}$, $\mathbf{IR}_{\mathrm{IN}}$, $R_{\mathrm{OUT}}$, $\mathbf{MR}_{\mathrm{OUT}}$, and $\boldsymbol{P}_{\mathrm{OUT}}$;
16:    end for
17:    Obtain the signals "0" and "1" probabilities of POs by
         $R_{\mathrm{out}}$, then calculate $p(i, j)$;
18:    Convert the PI vector's decimal value $i$ to a binary value,
         and then calculate AV($i$) and $V(i)$ for the ACC;
19:    Calculate $p_{\mathrm{out}}(i)$ in Methods 1 and 2;
20: end for
21: Rel $= \frac{1}{2^{\mathrm{NPI}}} \sum_{i=0}^{2^{\mathrm{NPI}}-1} p_{\mathrm{out}}(i)$.
```

Algorithm 2 Correlated gate searching

```
1: for every $j$-th level of the ACC do
2:     for every line node $l$ in level $j$ do
3:         Mark $l$ with a number set of its fanout sources
             FromFan($l$) and initialize it by FromFan($l$) $= \varnothing$;
4:         if $l$ is one of the fanout branches then
5:             Put the number of this fanout source into
                 FromFan($l$);
6:         else
7:             FromFan($l$) = FromFan(IN1)$\cup$ FromFan(IN2)
8:         end if
9:     end for
10: end for
```

Algorithm 3 Correlation coefficient calculation

```
1: if for gate $g$'s inputs IN1 and IN2, FromFan(IN1) $\cup$
     FromFan(IN2) $\neq \varnothing$ then
2:     Calculate correlation vector $\boldsymbol{C}_{\mathrm{IN1,IN2}}$ of IN1 and IN2;
3:     Back trace to calculate the precursor correlation vector
         recursively until reaching the fanout source;
4:     if for two line nodes $m$ and $n$, FromFan($m$)$\cup$
         FromFan($n$) $= \varnothing$ then
5:         Set $\boldsymbol{C}_{m,n} = \{1, 1, 1, 1\}$;
6:     end if
7: end if
```

Signal correlation originates from fanout. Thus, Algorithm 2 describes the steps in constructing the paths for all branches of the fanouts. Lines 1–5 show that the fanout sources are searched level by level; for a fanout source f, a path can be constructed by the set FromFan(l), i.e., if $f \in$ FromFan(l), line node l is on the path originating from f. Every path starts from one fanout source and ends at the next fanout source. In Line 7, IN1 and IN2 are the input nodes driving l. For an inverter, IN2 is ignored. If one of FromFan(IN1) and FromFan(IN2) is empty, the input node does not come from a fanout. The line nodes with intersecting FromFan sets are correlated. Then, the correlation vector needs to be calculated for correlated gates. Algorithm 3 shows the calculation process of the correlation vector.

Take Fig. 6 as an example. The circuit is a part of

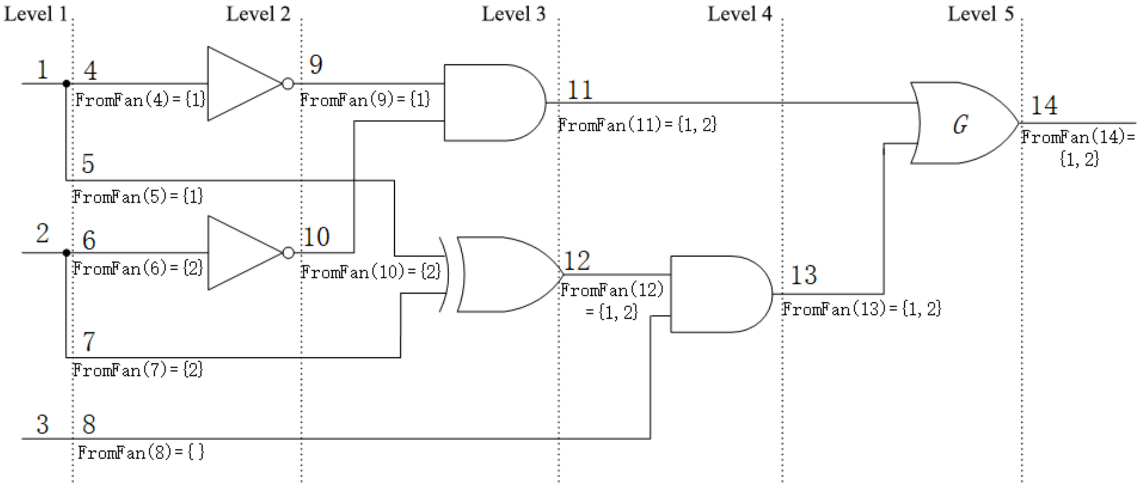


Fig. 6 Correlated gate searching.

approximate adder AXA1, and the sets from FromFan(1) to FromFan(14) are derived according to Algorithm 2. Then, we can know that line nodes 4, 5, 6, 7, 9, 10, 11, 12, 13, and 14 are on reconvergent paths. As shown in Fig. 6, from levels 1 to 5, only gate G has two correlated input nodes 11 and 13 (Algorithm 3); that is,

(1) $\boldsymbol{C}_{11,13}$ = function($\boldsymbol{C}_{11,12}, \boldsymbol{C}_{11,8}$);

(2) $\boldsymbol{C}_{11,12}$ = function($\boldsymbol{C}_{11,5}, \boldsymbol{C}_{11,7}$) = function($\boldsymbol{C}_{9,5}, \boldsymbol{C}_{10,5}, \boldsymbol{C}_{9,7}, \boldsymbol{C}_{10,7}$), and $\boldsymbol{C}_{11,8} = \{1,1,1,1\}$;

(3) $\boldsymbol{C}_{9,5}$ = function($\boldsymbol{C}_{4,5}$), $\boldsymbol{C}_{10,5} = \{1,1,1,1\}$, $\boldsymbol{C}_{9,7} = \{1,1,1,1\}$, and $\boldsymbol{C}_{10,7}$ = function($\boldsymbol{C}_{6,7}$);

(4) $\boldsymbol{C}_{4,5}$ and $\boldsymbol{C}_{6,7}$ can be calculated directly.

The calculations for $\boldsymbol{C}_{4,5}, \boldsymbol{C}_{6,7}$ and other functions are provided in Table 1. The correlation vector $\boldsymbol{C}_{11,13}$ can be calculated by a recursive function. For the correlation vectors of the nodes on the fanout branches, we calculate and store them in a table in advance and look them up from the table during recursion. By applying Algorithms 2 and 3 to Algorithm 1, we can calculate the reliability of the ACC.

At this point, we analyze the complexity of Algorithm 1. In Methods 1 and 2, based on the iterative PTM, every primary gate is regarded as a basic circuit component. Hence, the space complexity is *O(N)*, where N is the gate number of the ACC. The time complexity is mainly determined by two parts, namely, Algorithm 2 and the loop of Lines 4 to 20 of Algorithm 1. If we define the number of levels in the ACC as L and the average number of line nodes in every level as N_L, the time complexity of Algorithm 2 is $O(L \times N_L)$. The runtime of Algorithm 3 is also related to L, N_L, and the internal structure of the network. The calculation time is generally less than $\sum_L \frac{N_L(N_L-1)}{2}$. Thus, the time of Lines 4–20 in Algorithm 1 is less than $2^{\text{NPI}} \sum_L \frac{N_L(N_L-1)}{2} \times$ NPI. Despite the theoretical time complexity being exponential, the ACC reliability analysis approaches that are based on the iterative PTM and consider correlations have high accuracy. Furthermore, as the large approximate adder or multiplier is usually composed of small basic ones, NPI is not large in most cases.

5 Experimental Analysis

5.1 Experimental setup

The experimental circuit includes three series of typical approximate adders and a number of ACCs in the EvoApprox8b library. EvoApprox8b[29] is an open source library that provides many different approximate adders and approximate multipliers. We implement the proposed methods and MC simulation by using C++. All the experiments are conducted on a 2.2 GHz laptop with 8 GB memory.

Herein, the MC method and two other methods (called A1 and A2) proposed in Ref. [2] are selected for comparison. A1 and A2 are the methods of ACC reliability evaluation based on the original PTM; they feature a large space complexity, and they do not have any correlation coefficients.

5.2 Experimental result analysis

Method 1(A3) and Method 2(A4) in this work are compared with A1, A2, and MC methods. The results obtained under the assumption that the primary gate fault probability p is 0.01 are shown in Table 2. As the MC

Table 2 Accuracy verification and cost comparison for A3 and A4 with A1, A2, and MC.

ACC	Nodes	Rel. diff. (%)		Runtime (ms)					Memory (MB)				
		A3	A4	A3	A1	A4	A2	MC	A3	A1	A4	A2	MC
AXA1	31	1.4107	3.5034	3.1	77.8	3.5	99.3	119980	14.5	24.98	14.49	25.00	5.46
AXA3	41	0.6551	1.0816	1.6	365.4	1.4	276.7	101 770	14.48	25.00	14.52	24.98	5.47
AMA1	27	0.1901	1.4125	1.8	28.7	1.3	24.8	119 980	14.48	24.99	14.51	24.98	5.46
AMA2	23	0.0544	0.5048	1.3	76.8	2.1	84.0	89 250	14.48	24.99	14.48	25.00	5.46
InXA1	13	0.0005	0.5970	1.4	18.7	1.7	17.9	39 770	14.49	24.98	14.48	24.97	5.45
InXA3	31	0.4781	0.6486	2.1	74.0	2.5	71.7	87 850	14.51	25.01	14.51	24.98	5.45
Average	–	0.4648	1.2913	1.9	106.9	2.1	95.7	93 100	14.49	24.99	14.5	24.99	5.46

method is time consuming and A1 and A2 have a large memory overhead, only small-scale ACCs are used. All of the six approximate adders have 5 PIs and 3 POs. In Table 2, "Nodes" shows the number of line nodes in every ACC, that can affect the runtime and memory of the analytical methods.

The accuracy of A3 and A4 is proved by comparing them with MC simulation in Columns 3 and 4. "Rel. diff. (%)" denotes the percentage difference of reliability, i.e., $\frac{|\mathrm{A}i-\mathrm{MC}|}{|\mathrm{MC}|}$, where A$i$ means A3 in Column 3 and A4 in Column 4. Among the six ACCs, the maximum difference value of A3 is 1.4107%, and the average value is only 0.4648%. As for A4, the values are slightly large because different weights are combined to approximate values. By contrast, the MC method treats approximate values equally. The maximum difference value of A4 is 3.5034%, and the average value is 1.2913%. The difference value illustrates the accuracy of our methods. About the temporal and spatial overhead, we make comparison with MC, A1 and A2. For an example, according to the average time costs of A3 (i.e., 1.9) and MC (93100), we can calculate the ratio 0.002%. Hence, the time costs of A3 and A4 are 0.002% and 0.1% of those of the MC method, respectively. Relative to A1 and A2, A3 and A4 have a space cost of 50% and time costs of 1.78% and 2.19%, respectively. Relative to A1, A3 has a relatively stable runtime because the time complexity of A3 mainly depends on the PI number of the ACC and that of A1 depends on the line node number of the ACC. The six experimental ACCs have the same PI number, and thus, the runtimes of A3 on these circuits are similar. Meanwhile, AXA3 has the most line nodes; thus, the runtime of A1 on AXA3 is the longest. The runtime characteristics for A4 and A2 are similar. Columns 10–13 illustrate that the memory overhead of A3 and A4 is almost half of that of A1 and A2, respectively. The space complexity of A1 and A2 is almost $O(2^n)$, and n is the width of the ACC, i.e., the maximum line node number in all levels. By contrast, A3 and A4 have a nearly linear space complexity. As shown in Table 2, a portion of the preprocessing in the program causes the trend of the space complexity to be obscure.

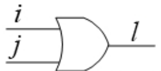
Table 3 shows the results of A3 and A4 methods running on 10 ACCs that are randomly selected from the EvoApprox8b library. All the ACCs have 16 input nodes. The adders have 9 output nodes, and the multipliers have 16 output nodes. In Table 3, the second column shows the total node number of each circuit. Columns 3 and 4 and Columns 5 and 6 show the reliabilities calculated by

Table 3 Reliability calculated by A3 and A4 for ACCs in EvoApprox8b.

ACC	Nodes	Reliability (p = 0.001)		Reliability (p = 0.0001)		p = 0.001(A3)	
		A3	A4	A3	A4	Runtime (s)	Memory (MB)
Add8_002	91	0.957078	0.956 988	0.995 562	0.995 556	14.66	22.71
Add8_034	112	0.978 099	0.977 740	0.997 657	0.997 568	14.96	22.70
Add8_039	111	0.980 825	0.980 721	0.998 218	0.998 194	16.26	22.71
Add8_057	102	0.949 126	0.949 088	0.994 819	0.994 818	14.17	22.67
Add8_070	91	0.986 841	0.986 753	0.999 231	0.999 167	13.87	22.71
Mul8_004	480	0.894 757	0.893 559	0.927 421	0.926 815	31.67	22.66
Mul8_024	453	0.846 698	0.844 198	0.876 521	0.874 408	30.73	22.67
Mul8_025	483	0.846 099	0.843 371	0.876 682	0.874 361	30.55	22.66
Mul8_044	472	0.879 174	0.876 475	0.920 183	0.918 427	29.74	22.75
Mul8_113	510	0.843 173	0.840 551	0.876 032	0.874 032	31.93	22.66

A3 and A4 given the fault probabilities of primary gate p of 0.001 and 0.0001, respectively.

For A3, the reliability trend under p = 0.001 is consistent with that under p = 0.0001. With regard to the general trend, the reliabilities of the approximate multipliers are smaller than those of the approximate adders because of the circuit scale. For the ACCs with similar node numbers, e.g., the five approximate adders, the reliability values are affected by the gate number and correlated coefficient calculation. If the correlation condition is the same, the reliability decreases with the increasing gate number. However, the consideration of correlation coefficients does not always lead to a simple reliability trend. Take the OR gate in Fig. 7 as an example. The range of $C_{i=1,j=1}$ of the OR gate is calculated. We can find that the correlated coefficients are not always smaller than 1. For the reliabilities calculated by A4, we observe similar phenomena and trends. Under a fixed value of p, the reliabilities calculated by A4 are always smaller than those calculated by A3. The reason is that in A4, every acceptable output value is multiplied by a weight; the weight is smaller than 1, except for that of AV. The last two columns show the runtime and memory overhead. As the overheads between A3 and A4 for each circuit are similar, only the values of A3 are listed.



$$p_{l=1} = p_{i=1} + p_{j=1} - p_{i=1}p_{j=1}C_{i=1,j=1}$$

$$0 \leqslant p_{l=1} \leqslant 1$$

$$\frac{p_{i=1} + p_{j=1} - 1}{p_{i=1}p_{j=1}} \leqslant C_{i=1,j=1} \leqslant \frac{p_{i=1} + p_{j=1}}{p_{i=1}p_{j=1}}$$

Fig. 7 Reliability calculated by A3 and A4 for ACCs in EvoApprox8b.

6 Conclusion

This study proposes two relatively complete and accurate methods for ACC reliability evaluation that are based on the iterative PTM with correlation vectors. The correlation coefficients of four signal combinations under different basic gate connection structures are derived to improve the accuracy of the reliability evaluation of ACCs. The proposed methods are less time consuming than the MC method, but their accuracies are similar. Moreover, the proposed methods have lower time and space costs than the methods for ACC reliability evaluation based on the original PTM. The accuracy and feasibility of the proposed methods are verified by experiments.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (Nos. 61432017 and 61772327), the Natural Science Foundation of Shanghai (Nos. 20ZR1455900 and 20ZR1421600), and the Qi'anxin National Engineering Laboratory for Big Data Collaborative Security Technology Open Project (No. QAX-201803), and State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences (No. CARCHA202005).

References

[1] M. A. Breuer, Intelligible test techniques to support error-tolerance, in *Proc. of Asia Test Symposium*, Kenting, China, 2004, pp. 386–393.

[2] Y. Song and W. Liang, Experimental study of discriminative adaptive training and MLLR for automatic pronunciation evaluation, *Tsinghua Science and Technology*, vol. 16, no. 2, pp. 189—193, 2011.

[3] S. K. Min, A. D. Barrio, R. Hermida, and N. Bagherzadeh, Low-power implementation of Mitchell's approximate logarithmic multiplication for convolutional neural networks, in *Proc. of Asia & South Pacific Design Automation Conference*, Jeju Island, Korea, 2018, pp. 617–622.

[4] G. Liu and Z. R. Zhang, Statistically certified approximate logic synthesis, in *Proc. of IEEE International Conference on Computer-Aided Design*, Irvine, CA, USA, 2017, pp. 344–351.

[5] C. Y. Chen, J. Choi, K. Gopalakrishnan, V. Srinivasan, and S. Venkataramani, Exploiting approximate computing for deep learning acceleration, in *Proc. of Design, Automation & Test in Europe Conference & Exhibition*, Dresden, Germany, 2018, pp. 821–826.

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

[7] A. Yazdanbakhsh, J. Park, H. Sharma, P. Lotfi-Kamran, and H. Esmaeilzadeh, Neural acceleration for GPU throughput processors, in *Proc. of IEEE International Symposium on Microarchitecture*, Waikiki, HI, USA, 2015, pp. 482–493.

[8] N. Jing, W. Bi, and Q. Yue, Attack simulation model and channel statistics in underwater acoustic sensor networks, *Tsinghua Science and Technology*, vol. 16, no. 6, pp. 611–621, 2011.

[9] S. Sidiroglou-Douskos, S. Misailovic, H. Hoffmann, and M. Rinard, Managing performance vs. accuracy trade-offs with loop perforation, in *Proc. ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, Szeged, Hungary, 2011, pp. 124–134.

[10] M. Shafique, R. Hafiz, S. Rehman, W. El-Harouni, and J. Henkel, Invited: Cross-layer approximate computing: From logic to architectures, in *Proc. of 53nd ACM/EDAC/IEEE Design Automation Conference*, Austin, TX, USA, 2016, pp. 99:1–99:6.

[11] A. Yan, C. Lai, Y. Zhang, J. Cui, Z. Huang, J. Song, J. Guo, and X. Wen, Novel low cost, double-and-triple-node-upset-tolerant latch designs for nano-scale CMOS, *IEEE Transactions on Emerging Topics in Computing*, doi: 10.1109/TETC.2018.2871861.

[12] J. Jiang, G. Lu, and Z. Wang, Methods for approximate adders reliability estimation based on PTM model, in *Proc. of IEEE 23th Pacific Rim International Symposium on Dependable Computing*, Taipei, China, 2018, pp. 221–222.

[13] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, Accurate reliability evaluation and enhancement via probabilistic transfer matrices, in *Proc. of the Design, Automation and Test in Europe Conference and Exhibition*, Munich, Germany, 2005, pp. 282–287.

[14] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, MACACO: Modeling and analysis of circuits for approximate computing, in *Proc. of IEEE International Conference on Computer-Aided Design*, San Jose, CA, USA, 2011, pp. 667–673.

[15] J. H. Liang, J. Han, and F. Lombardi, On the reliable performance of sequential adders in soft computing, in *Proc. of IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, Vancouver, Canada, 2011, pp. 3–10.

[16] J. H. Liang, J. Han, and F. Lombardi, New metrics for the reliability of approximate and probabilistic adders, *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1760–1771, 2013.

[17] C. Liu, J. Han, and F. Lombardi, An analytical framework for evaluating the error characteristics of approximate adders, *IEEE Transactions on Computers*, vol. 64, no. 5, pp. 1268–1281, 2015.

[18] Y. Wu, Y. Li, X. Ge, Y. Gao, and W. Qian, An efficient method for calculating the error statistics of block-based approximate adders, *IEEE Transactions on Computers*, vol. 68, no. 1, pp. 21–38, 2019.

[19] H. Jiang and G. Liu, A review, classification, and comparative evaluation of approximate arithmetic circuits, *ACM Journal on Emerging Technologies in Computing Systems*, vol. 13, no. 4, pp. 1–34, 2017.

[20] L. Chen, J. Han, W. Liu, P. Monstuschi, and F. Lombardi, Design, evaluation and application of approximate high-radix dividers, *IEEE Transactions on MultiScale Computing Systems*, vol. 4, no. 3, pp. 299–312, 2018.

[21] M. Bruestel and A. Kumar, Accounting for systematic errors in approximate computing, in *Proc. of IEEE Design, Automation & Test in Europe Conference & Exhibition*, Lausanne, Switzerland, 2017, pp. 298–301.

[22] Ilia Polian, Test and reliability challenges for approximate circuitry, *IEEE Embedded Systems Letters*, vol. 10, no. 1, pp. 26–29, 2018.

[23] L. Anghel, M. Benabdenbi, and A. Bosio, Test and reliability in approximate computing, in *Proc. of International Mixed Signals Testing Workshop*, Thessaloniki, Greece, 2017, pp. 375–387.

[24] Y. Fang, H. Li, and X. Li, A fault criticality evaluation framework of digital systems for error tolerant video applications, in *Proc. of Asian Test Symposium*, New Delhi, India, 2011, pp. 329–334.

[25] Z. Wang, J. Jiang, and T. Wang, Failure probability analysis and critical node determination for approximate circuits integration, *the VLSI Journal*, vol. 68, no. 5, pp. 122–128, 2019.

[26] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, Probabilistic transfer matrices in symbolic reliability analysis of logic circuits, *ACM Trans. on Design Automation of Electronic Systems*, vol. 13, no. 1, pp. 1–35, 2008.

[27] J. Xiao, W. Lee, J. H. Jiang, and X. H. Yang, Circuit reliability estimation based on an iterative PTM model with hybrid coding, *Microelectronics Journal*, vol. 52, pp. 117–123, 2016.

[28] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, Estimate of signal probability in combinational logic networks, in *Proc. of 1st European Test Conference*, Paris, France, 1989, pp. 132–138.

[29] V. Mrazek, R. Hrbacek, Z. Vasicek, and L. Sekanina, EvoApprox8b: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods, in *Proc. of Design, Automation & Test in Europe Conference & Exhibition*, Lausanne, Switzerland, 2017, pp. 258–261.

Zhen Wang received the PhD degree in computer science and technology from Tongji University in 2008. She ever worked as a senior engineer in Synopsys from 2008 to 2013. She is now working at Shanghai University of Electric Power. Her main research interests include fault tolerant computing, reliability evaluation of high-level circuits, and approximate computing. She is a member of China Computer Federation.

Guofa Zhang received the BS degree in electronic science and technology, from Jiliang University, China in 2019. He is a master student in the School of Computer Science and Technology, Shanghai University of Electric Power. His main research interest is approximate computing.

Jing Ye received the PhD degree from Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS) in 2014, and the BS degree in electronics engineering and computer science from Peking University in 2008. He is currently an associate professor at the State Key Laboratory of Computer Architecture, ICT, CAS, and the secretary-general of Technical Committee on Fault Tolerant Computing, China Computer Federation (CCF). His current research interests include VLSI test and diagnosis, hardware security (post quantum cryptography and physical unclonable function), and AI software/hardware security.

Jianhui Jiang received the PhD degree in traffic information engineering and control from Shanghai Tiedao University (in April 2000, it was merged to Tongji University) in 1999. Since 2011, he has been an associate dean of the School of Software Engineering, Tongji University. He is a professor and PhD supervisor at Tongji University. He is a senior member of China Computer Federation. His main research interests include dependable systems and networks, software reliability engineering, and VLSI test and fault tolerance.

Fengyong Li received the PhD degree from Shanghai University in 2014. He is currently an associate professor in Shanghai University of Electric Power. He worked as a visiting scholar at University of Victoria, Canada from 2018 to 2019. His research interests include multimedia security, information hiding, and machine learning.

Yong Wang received the PhD degree in system analysis and integration from East China Normal University in 2007, finished postdoctoral research from Shanghai Jiaotong University. He had been a visiting scholar at Texas A&M University. He is a professor in Shanghai University of Electric Power. His research interests are power system network security and false data injection attacks detection.