IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 December 2023, accepted 13 February 2024, date of publication 21 February 2024, date of current version 11 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3368512

RESEARCH ARTICLE

An Effective Fanout-Based Method for Improving Error Propagation Probability Estimation in Combinational Circuits

ESFANDIAR ESMAIELI[1], ALI PEIRAVI[1], AND YASSER SEDAGHAT[2]
[1]Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad 9177948974, Iran
[2]Dependable Distributed Embedded Systems (DDEmS) Laboratory, Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad 9177948974, Iran

Corresponding author: Yasser Sedaghat (y_sedaghat@um.ac.ir)

ABSTRACT The downsizing of nanoscale circuits imposes new challenges for circuit reliability, including hard defects, soft errors and unsaturated voltage/current. Many studies on the reliability of digital circuits have focused on achieving accurate reliability estimation and more efficiency for larger circuits. To achieve accurate reliability estimation, it is necessary to address the issue of error propagation and consider correlated signals from reconverging paths in reliability calculations. In this paper, an error propagation probability model for each gate, which takes into account the probability of an unreliable logic gate's input signal and relates it to the probability of the output signal is proposed. Additionally, we introduce an efficient approach that utilizes a new fanout matrix to tackle the reconvergent fanouts problem. Furthermore, to ensure an accurate estimation of combinational logic circuit reliability, the probabilities obtained for each fanout should be included in the calculations by defining a fanout probability matrix. To address this issue, a new method is proposed at each calculation stage, aiming to minimize computational complexity making it suitable for large circuits with a significant number of fanouts. We conducted various simulations to demonstrate the accuracy and scalability of the proposed method on the ISCAS 85 benchmark circuit and EPFL Benchmark. The results show less than 1% average relative error in reliability estimation and outperform state-of-the-art methods in reliability estimation and algorithm runtime.

INDEX TERMS Combinational circuit, convergent path, error masking, gate level reliability, reliability, error propagation.

I. INTRODUCTION

The continuous down-scaling of CMOS technology presents new challenges for digital circuit designers, particularly in terms of circuit reliability. Nanoscale fabrication imprecision can lead to many hard and soft errors due to environmental variations, which can affect device reliability [1], [2]. The problem is compounded by high integration density and unsaturated voltage/current. Device unreliability can have a negative impact on circuit performance at high levels, making reliability a significant concern for circuit designers as technology is scaled down to a few nanometers [3]. To ensure market competitiveness, there is a need for a quick and efficient evaluation method to measure circuit reliability at early stages of circuit design, enabling timely decisions and shortening product development cycles [4], [5].

The computational methods for estimating the reliability of digital circuits are usually divided into two categories: Statistical models and Probabilistic models [6]. Statistical methods, such as Monte Carlo (MC) logic simulation [7], [8], [9], Stochastic Computation Models (SCM) [10], [11], and Bayesian inference (BI) [12], are employed for reliability estimation. Their accuracy improves with more iterations; however, their long processing time makes them impractical for large integrated circuits. They are commonly employed to evaluate the accuracy of other methods. In these methods,

The associate editor coordinating the review of this manuscript and approving it for publication was Cristian Zambelli.

a fault pattern is randomly generated and injected into circuit wires (including primary inputs, primary outputs, and gate pins) to evaluate logic values and detect errors. Some approaches, like the stochastic method, attempt to reduce its high time complexity [6].

Probabilistic models employ gate behaviors, fault probability distributions, and probabilistic equations for reliability estimation. They offer the advantage of being iteration-free and faster than statistical methods, making them widely employed for reliability estimation. However, the presence of converging paths poses a significant challenge as it leads to repetitive states and inaccurate calculations. Various solutions have been proposed to address this issue, often by adding specific assumptions to improve accuracy while reducing computational costs. Exact methods like Probabilistic Transfer Matrices (PTMs) [13], [14], [15], Probabilistic Gate Models (PGMs) [16], Binary Decision Diagrams (BDDs) [17], Boolean Difference Calculus [18], [19], [20], Conditional Probability Matrix (CPM) [21], [22], and Bayesian Networks (BNs) [1], [23], calculate signal probability. However, their exponential time/space complexity limits their applicability to small circuits. Analytical methods inspired by von Neumann's study [6], employ gate reliability models such as PGM, PTM, Bayesian networks, Boolean difference calculus, etc., offering high efficiency, however, with accuracy concerns. Recent approaches introduced correlation coefficients to evaluate signal correlations. However, they either rely on rough estimations or exhibit relatively large errors [24], [25], [26].

Probabilistic methods for achieving accurate reliability in large-scale circuits face challenges of scalability and computational efficiency. Convergent paths and correlated signals from fanouts can introduce inaccuracies in computation results. PTM and PGM-based methods accurately estimate reliability by considering all possible signal states of fanout-generated signals. However, they become impractical as the number of fanouts increases exponentially. Signal probability and correlation coefficient-based methods consider four signal states (1 correct, 1 incorrect, 0 correct, 0 incorrect), However, they require larger correlation coefficient matrices with more dependent signals. Techniques like Monte Carlo simulations and SCM are employed to reduce computational burden, however they are time-consuming for large circuits. Error propagation probability-based methods, utilizing Boolean functions, reduce computational complexity with two signal states (correct, fail) and can be combined with other methods for a hybrid approach.

In our previous work [27], we evaluated the reliability of combinational circuits based on signal probability and effective fanouts. In this paper, an accurate and scalable reliability estimation method has been developed for large combinational circuits with linear computational complexity increasing with circuit fanouts. As in [18], we propose an error propagation model to estimate the failure probability in combinational circuits. The main objective is to introduce fundamental concepts, such as fault propagation matrix and input/output matrix of gates. The concept of effective fanouts and the fanout probability matrix has been modified to accommodate the reconvergent paths. The contributions of this paper can be summarized as follows:

1) Modeling the release probability matrix for each gate.
2) Determining the new fanout matrix for each fanout.
3) Identifying the effective fanouts for calculating the circuit's output probability using the proposed method.
4) Developing a new method to calculate the output probability matrix in the presence of converging paths and the fanout matrix defined for each fanout to eliminate duplicate calculations.
5) Presenting a new method to determine fanout probability matrix for each effective fanout and apply them to accurately estimate the circuit's reliability.

The paper is organized as follows: Section II introduces some preliminaries on signal probability and reliability (input probability matrix, fault propagation matrix and output probability matrix). Section III introduces the problem of convergent paths and their effect on calculations, and then a solution for this issue is presented. Section IV explains how to calculate the probability of the output signal of the circuit. Section V provides an example of how to calculate the reliability of digital circuits. Section VI explains the accuracy of the simulation and describes the reasons for calculation errors. Section VII shows our simulation results, and Section VIII concludes the paper.

II. PROBABILISTIC MODEL OF ERROR PROPAGATION

The signal probability concept has been used for fault-prone circuit reliability estimates. In this section, we describe some background of digital signal probability and concept fault propagation.

A. THE FAULT PROPAGATION MATRIX

A fault propagation matrix (FPM) is defined for each gate based on its behavior and input signal probability to calculate the digital circuit's reliability. The values in the matrix represent the probability of fault propagation for each input signal combination and thoroughly examine the logical fault masking. For example, FPM of a two-input AND gate with input signals a and b is shown in Equation 1.

$$\begin{matrix} a_{correct}b_{correct} \\ a_{correct}b_{fail} \\ a_{fail}b_{correct} \\ a_{fail}b_{fail} \end{matrix} \begin{bmatrix} \varepsilon \\ (1-\varepsilon)P_a+\varepsilon Q_a \\ (1-\varepsilon)P_b+\varepsilon Q_b \\ (1-\varepsilon)(P_a.P_b+Q_a.Q_b)+\varepsilon(Q_aP_b+Q_bP_a) \end{bmatrix} \tag{1}$$

ε is the gate's fault probability.

In this equation, $P_x (P_x = 1-Q_x)$ represents the probability of a circuit node being in the state '1' [18]. This probability is computed within an error-free circuit using the bit stream method, which bears resemblance to the approach described in [22] and [25].

For example, in Equation 1, if one input of the 2-input AND gate has a fault, then the fault will be propagated only if the other input has logic 1. Otherwise, if the other input is 0, the fault in the gate's output only results from the internal fault of the gate. Regarding the above discussions, we tried to include the fault logical masking problem in the reliability estimation using FPM and the gate fault propagation matrix based on the gate behavior is shown in Table 1.

TABLE 1. The gate fault propagation matrix.

Gate	Fault Propagation Matrix (FPM)
AND&NAND	$\begin{matrix} a_{correct}b_{correct} \\ a_{correct}b_{fail} \\ a_{fail}b_{correct} \\ a_{fail}b_{fail} \end{matrix}\begin{bmatrix} \varepsilon \\ (1-\varepsilon)P_a+\varepsilon Q_a \\ (1-\varepsilon)P_b+\varepsilon Q_b \\ (1-\varepsilon)(P_a.P_b+Q_a.Q_b)+\varepsilon(Q_aP_b+Q_bP_a) \end{bmatrix}$
OR& NOR	$\begin{matrix} a_{correct}b_{correct} \\ a_{correct}b_{fail} \\ a_{fail}b_{correct} \\ a_{fail}b_{fail} \end{matrix}\begin{bmatrix} \varepsilon \\ (1-\varepsilon)Q_a+\varepsilon P_a \\ (1-\varepsilon)Q_b+\varepsilon P_b \\ (1-\varepsilon)(P_a.P_b+Q_a.Q_b)+\varepsilon(Q_aP_b+Q_bP_a) \end{bmatrix}$
XOR	$\begin{matrix} a_{correct}b_{correct} \\ a_{correct}b_{fail} \\ a_{fail}b_{correct} \\ a_{fail}b_{fail} \end{matrix}\begin{bmatrix} \varepsilon \\ (1-\varepsilon) \\ (1-\varepsilon) \\ \varepsilon \end{bmatrix}$

B. THE OUTPUT SIGNAL PROBABILITY MATRIX

With appropriate use of the gate's input probability matrix in Equation 2 (represent the probabilities of the correct or faulty states of the inputs) and the FPM, the output signal probability matrix (SPM) can be calculated according to Equation 4:

$$Input_a = \begin{bmatrix} a_{carnet} & a_{fat} \end{bmatrix}, Input_b = \begin{bmatrix} b_{correct} & b_{fail} \end{bmatrix} \quad (2)$$

$$T = \begin{bmatrix} a_{carret}b_{correet} & a_{carrer}b_{fall} & a_{fall}b_{carrect} & a_{fall}b_{fall} \end{bmatrix}_{1\times 4} \quad (3)$$

$$P_{fail} = T \times FPM$$
$$SPM_{OUT} = \begin{bmatrix} 1-P_{fail} & P_{fail} \end{bmatrix} \quad (4)$$

For example, the calculation steps of the fault probability in the output of a 2-input AND gate are depicted in Figure 1. The probability of a fault in the A and B inputs are 0.1 and 0.2, respectively, the gate's fault probability is 0.05. Additionally, the probability of both input signals being in the state '1' is 0.5 ($P_a, P_b = 0.5$). According to Equation 4, the gate's output fault probability is obtained to be 0.176 based on the AND gate fault propagation matrix which is shown in Figure 1. The probability of the output signal of each gate is considered as the input matrix of the next gate. This process continues until the output of the circuit is reached.

III. THE CONVERGING PATHS

The converging paths are an essential issue that should be considered in the reliability calculations. The reason is that

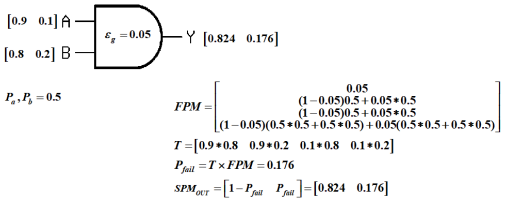

FIGURE 1. The fault probability in the output of AND gate.

these paths create similar (iterative) states and impossible state calculations in the reliability calculation. The impossible states refer to the condition where the signal cannot be 0 or 1 simultaneously. However, these states may occur in calculations and must be removed. Since fanouts are the origin of iterative and impossible states, one matrix is considered for each fanout that has to satisfy the two discussed assumptions during reliability calculations. For this purpose, the signal probability matrix for each fanout is defined as Equation 5. This matrix considers two separate states for the fanout. The first row shows that only the "fault signal" state is applied to the circuit as the fanout probability matrix. The second row determines the "correct signal" state, which is used in the circuit as the fanout signal probability matrix. Selecting 0 and 1 elements for each row will prevent creation of repetitive and impossible states of the fanout.

$$SPM_{fanout} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (5)$$

Now assume that the circuit has more than one fanout. For each fanout, a fanout matrix is assigned. Section IV provides a comprehensive explanation of how to calculate the probability of the output signal of the circuit, specifically addressing scenarios involving multiple fanouts.

A. EFFECTIVE FANOUT ALGORITHM

Since fanouts are the primary cause of repeated and impossible states in reliability calculations, it is crucial to consider their effect during calculations. The accuracy of estimates is influenced by fanouts when there is more than one path from a fanout to the output. Otherwise, if a fanout has only a single path, it is treated as a normal and independent signal, and there is no need to define a fanout matrix for it. However, if a fanout, initially intended for one output, travels multiple paths for other outputs, and we intend to assess the reliability of these outputs together, the fanout no longer functions as an independent signal.

This section presents a solution for determining the effective fanouts before assigning the fanout matrix to them. Effective fanouts are determined and numbered based on the primary circuit's output and a depth-first algorithm [28]. First, we define a circuit's node matrix to specify the fanouts and primary outputs of the circuit. For example, the node matrix of the circuit in Figure 2 is determined in Equation 6, which can be utilized for determining the fanout nodes and the

circuit's output. The nodes with a repetition count above 2 in matrix X are identified as the fanout nodes, and the nodes with a repetition count equal to 0 in matrix X are identified as the circuit's output node. Then, the fanouts with more than one path to the output are specified using the depth-first algorithm. Therefore, the conditions for selecting an effective fanout are as follows:

1. The circuit's primary input fanouts are not considered effective fanouts. This is due to our assumption of error-free primary input, as indicated in [12], [15], and [23], a probability input matrix of [1 0] is considered. Consequently, in terms of input fanout configurations, the binary nature of the input probability matrix eliminates the possibility of similar iterative states or impassable states in the computations.
2. Fanouts with a maximum of one path to the output are not considered to be effective fanouts.

After determining the effective fanouts, they are sequentially numbered from the input to the output, starting with 1, and the entire procedure is described in Algorithm 1. Since the time complexity of the depth-first algorithm is $O(N)$, where N is the number of gates, the time complexity of Algorithm 1 can also be achieved in $O(N)$ for each primary output.

As shown in Figure 2, the effective fanouts are dedicated to the signals O_2, O_3, O_4, O_5, O_7 for the output signal O_{11}. Then, all circuit signals are assigned numbers based on the largest fanout number associated with that signal. For example, Table 2 shows the assigned fanout number for each circuit node in Figure 2 according to the largest fanout number associated with that signal.

$$X = [O_1 \quad O_2 \quad O_2 \quad O_3 \quad O_3 \quad O_4 \quad O_4 \quad O_5 \quad O_5 \quad O_6 \quad O_7 \quad O_7 \quad O_8 \quad O_9 \quad O_{10}]$$

$$Y = [O_6 \quad O_4 \quad O_5 \quad O_5 \quad O_8 \quad O_6 \quad O_7 \quad O_7 \quad O_8 \quad O_9 \quad O_9 \quad O_{10} \quad O_{10} \quad O_{11} \quad O_{11}]$$

$$\xrightarrow{\text{The number of repetitions of elements of the matrix X}} \begin{matrix} O_2 \\ O_3 \\ O_4 \\ O_5 \\ O_7 \\ O_{11} \end{matrix} \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 0 \end{bmatrix}$$

$$\rightarrow \left| \begin{matrix} \mathrm{O_2,O_3,O_4,O_5,\ O_7{=}fanouts} \\ \\ \\ \mathrm{O_{11}{=}output} \end{matrix} \right.$$

$$\xrightarrow{Depth-first\ algorithm}$$

$$\mathrm{O_2,O_3,O_4,O_5,\ O_7\ are\ effective\ fanout\ for\ O_{11}} \tag{6}$$

IV. THE OUTPUT SIGNAL'S PROBABILITY ALGORITHM

Now, considering the matrix size of input signals and the assigned fanout number, the output signal's probability of the two input signals is calculated according to one of the following states:

1. If the fanout number and matrix size of input signals are equal, the output probability matrix is calculated by Equation 7:

$$in_1 = \begin{matrix} correct & fail \end{matrix} \begin{bmatrix} a_{11} & a_{12} \\ \vdots & \vdots \\ a_{n1} & a_{n2} \end{bmatrix},$$

$$in_2 = \begin{matrix} correct & fail \end{matrix} \begin{bmatrix} b_{11} & b_{12} \\ \vdots & \vdots \\ b_{n1} & b_{n2} \end{bmatrix}$$

$$T = \begin{bmatrix} a_{1c}b_{1c} & a_{1c}b_{1f} & a_{1f}b_{1c} & a_{1f}b_{1f} \\ \vdots & \vdots & \vdots & \vdots \\ a_{nc}b_{nc} & a_{nc}b_{nf} & a_{nf}b_{nc} & a_{nf}b_{nf} \end{bmatrix}$$

$$SPM_{out} = T \times EPM_{gare} \tag{7}$$

2. If the fanout numbers of the input signal are equal and the sizes of the input signal matrices are not equal, the input matrix with the smaller number of rows is replicated as needed to match the number of rows in the larger input matrix, following the procedure described in equation 8. Subsequently, similar to the first case,

Algorithm 1 Search Effective fanout

An expanded method based on depth-first search.
Input: *Circuit netlist*
Output: *Effective fanout for each output*
1. *Extract the circuit information from the netlist file*
1.1 *Create matrix node form circuits*
1.2 *Obtain the sum of the elements of each row*
1.3 *If the sum of elements row => 2, the nodes are determined as the fanout node*
1.4 *If the sum of elements row = 0, the nodes are determined as the circuit's output node.*
2. *Determine effective fanouts for each output node of the circuit based on a depth-first search algorithm*
2.1 *For each output node of the circuit*
2.1.1 *If fanout is a circuit's main input, it is removed for the effective fanouts list*
2.1.2 *If a fanout has at most one path to the output, it is not considered effective.*
2.1.3 *Create a fanouts list for output*
2.2 *End For*
3. *Fanouts are numbered from the input to the output starting from one.*

TABLE 2. The fanout number of all the circuit nodes in Figure 2.

Nodes	in_1	in_2	in_3	in_4	in_5	in_6	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}	Q_{11}
Fanout Number	0	0	0	0	0	0	0	1	2	3	4	3	5	4	5	5	5

the probability of the output signal is calculated.

$$in_1 = \begin{matrix} correct & fail \end{matrix} \begin{bmatrix} a_{11} & a_{12} \\ \vdots & \vdots \\ a_{n_1 1} & a_{n_1 2} \end{bmatrix}_{n_1 \times 2},$$

$$in_2 = \begin{matrix} correct & fail \end{matrix} \begin{bmatrix} b_{11} & b_{12} \\ \vdots & \vdots \\ b_{n_2 1} & b_{n_2 2} \end{bmatrix}_{n_2 \times 2}$$

$$k = \frac{n_1}{n_2}, n_1 > n_2 \rightarrow in_{2new} = \begin{bmatrix} in_2 \\ \vdots k \\ in_2 \end{bmatrix}_{n_1 \times 2} \quad (8)$$

3. If the fanout number of input signals are different, first, each row of the input matrix with the lower fanout number should be repeated equally to f in Equation 9 to ensure an equal fanout number of input signals. Then, similar to the second case, the probability of the output signal is calculated.

$$in_1 = \begin{matrix} correct & fail \end{matrix} \begin{bmatrix} a_{11} & a_{12} \\ \vdots & \vdots \\ a_{n_1 1} & a_{n_1 2} \end{bmatrix}_{n_1 \times 2},$$

$$in_2 = \begin{matrix} correct & fail \end{matrix} \begin{bmatrix} a_{11} & a_{12} \\ \vdots & \vdots \\ a_{n_1 1} & a_{n_1 2} \end{bmatrix}_{n_2 \times 2}$$

$$f_1 = \text{fanout number in } {}_1$$
$$f_2 = \text{fanout number in } {}_2$$
$$f_2 > f_1$$
$$f = \text{minimum}\left(2^{(f_2 - f_1)}, n_2\right) \quad (9)$$

$$in_{1nww} = \begin{bmatrix} a_{11} & a_{12} \\ \vdots f & \vdots f \\ a_{11} & a_{12} \\ \vdots f & \vdots f \\ a_{n_1 1} & a_{n_1 2} \end{bmatrix} \quad (10)$$

As stated in Algorithm 2, based on the input signal matrix and the assigned fanout numbers, the output probability of each gate can be calculated and utilized as the input signal probability matrix for the next gate. However, if the gate's output signal is intended as the circuit's fanout, the calculated signal probability is stored as the fanout probability matrix with the name $P_{fanout,i}$, and a new signal probability matrix is defined according to Equation 5.

In Algorithm 2, the worst-case time complexity of step 1 is approximately $O(N * Size_{out})$. In step 2, only a matrix assignment is performed, so its time-space complexity is $O(1)$.

Algorithm 2 The Output Signal's Probability Algorithm

All effective fanouts are numbered from the input to the output, starting from one
All circuit signals are numbered according to the last number of the effective fanout
Inputs:*The matrix size of input signals and the assigned fanout number*
Output:*Output signal's probability*
1-*Calculate output signal's probability*
1.1- *If the fanout number and the number of two input signals rows are equal, the output probability matrix is calculated by Equation 7*
1.2- *If the fanout numbers of the signal are equal while the sizes of the input signal matrices are not equal: according to Equation 8, the number of two input signals rows is equal. Then go to case 1.1*
1.3- *If the fanout number of input signals is different, according to Equation 10, the assigned fanout number of two input signals is equal, then go to case 1.2.*
2-*Signal probability*
2.1- *If the gate's output signal is not the circuit's fanout, the output probability of each gate is used as the next gate input signal probability matrix.*
2.2- *If the gate's output signal will be used as the circuit's effective fanout, the calculated signal's probability is saved as$P_{fanout,i}$ a new signal's probability matrix is defined as Equation 5.*

To summarize the above analysis, the time complexity of Algorithm 2 is approximately $O(N * Size_{out})$.

V. COMPUTATION OF CIRCUIT RELIABILITY

In Section III, a new matrix was defined for each effective fanout in order to increase the accuracy of calculations and account for convergent paths, following Equation 5. Subsequently, a fanout probability matrix was saved for each fanout to calculate the probability of circuit failure. To calculate the final error probability, the stored matrix for each fanout must be applied in the calculations to achieve accuracy. In order to incorporate the fanout probability matrix into the calculations, according to Algorithm 3, we calculate the sum of all the states that the fanout probability matrices share with each other. To calculate the probability of circuit failure, we first define the following parameters:

$$Size_{fanout(i)} = \text{The number of rows } P_{fanout(i)}$$
$$Size_{out} = \text{The number of rows } P_{out}$$
$$F_{out} = \text{Fanout number of signal out}$$
$$F_{fanout(i)} = \text{Fanout number of } i\text{th fanout}$$

To calculate the reliability of a circuit's output, it is necessary first to determine all the states in which the fanouts relate to each other. The resulting matrix is then multiplied by the probability matrix of the output, and the sum of all states equals the reliability of the circuit's output. However, all states in which the fanouts relate to each other amount to $2^{F_{out}}$ in practice, and if the circuit is large, obtaining this matrix is practically impossible. Algorithm 3 provides

an efficient method for calculating reliability that does not require a matrix of size $2^{F_{out}}$. In a simple example, the proposed method is explained.

$$P_{fanou1} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \quad P_{fanou2} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

$$P_{fanou3} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}, \quad SPM_{out} = \begin{bmatrix} o_1 \\ o_2 \\ o_3 \\ o_4 \end{bmatrix}$$

$$\times \begin{bmatrix} a_1b_1c_1 \\ a_1b_1c_2 \\ a_1b_2c_3 \\ a_1b_2c_4 \\ a_2b_1c_1 \\ a_2b_1c_2 \\ a_2b_2c_3 \\ a_2b_2c_4 \end{bmatrix} .* \begin{bmatrix} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_1 \\ o_2 \\ o_3 \\ o_4 \end{bmatrix} \rightarrow \text{Re}liability$$

$$= Sum \begin{bmatrix} a_1b_1c_1o_1 \\ a_1b_1c_2o_2 \\ a_1b_2c_3o_3 \\ a_1b_2c_4o_4 \\ a_2b_1c_1o_1 \\ a_2b_1c_2o_2 \\ a_2b_2c_3o_3 \\ a_2b_2c_4o_4 \end{bmatrix} \tag{11}$$

$$\text{Re}liability = Sum \begin{bmatrix} a_1\left(b_1\left(c_1o_1 + c_2o_2\right) + b_2\left(c_3o_3 + c_4o_4\right)\right) \\ a_2\left(b_1\left(c_1o_1 + c_2o_2\right) + b_2\left(c_3o_3 + c_4o_4\right)\right) \end{bmatrix} \tag{12}$$

In the above example, we have a circuit with 3 fanouts and an SPM matrix. Equation 11 represents the typical method for calculating reliability, which involves a matrix of size 2^3. However, Equation 12 introduces a factorization that reduces it to Equation 11, eliminating the need to compute a matrix of size 2^3. The approach presented in Algorithm 3 aims to reduce computation time from $2^{F_{out}}$ to F_{out}, similar to Equation 12, in order to efficiently determine the reliability of digital circuits with a large number of fanouts while minimizing computational complexity.

In the first step, following Equation 8, we ensure that both matrices, the output probability matrix and the fanout probability matrix for the last fanout, are the same size because both matrices have the same number of fanouts. Then, we perform an element-wise multiplication of these two matrices, following the instructions in equation 13.

$$SPM_{out} = \begin{bmatrix} o_1 \\ \vdots \\ \vdots \\ o_m \end{bmatrix}_{Size_{out}\times 1}, \quad P_{fanout(n)} = \begin{bmatrix} n_1 \\ \vdots \\ \vdots \\ n_k \end{bmatrix}_{Size_{fanout(n)}\times 1}$$

$$r = \frac{Size_{out}}{Size_{fanout(n)}} \rightarrow P^{new}{}_{fanout(n)}$$

$$= \begin{bmatrix} P_{fanout(n)} \\ \vdots r \\ P_{fanout(n)} \end{bmatrix}_{Size_{out}\times 1}$$

$$P^n = SPM_{out} .* P^{new}{}_{fanout(n)}$$

$$= \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_{m-1} \\ P_m \end{bmatrix} \tag{13}$$

In the next step, the matrix elements obtained from the previous step are summed pairwise to create a P^i_{Sum} matrix based on Equation 14. Similar to the preceding step, the new matrix is initially resized to match the size of the fanout probability matrix for fanout number *i-1*, and then matrix p^{i-1} is derived as described in Equation 13. This process is iterated for each fanout, following a sequence in descending order of fanout numbers until we reach fanout number 1.

$$P^i = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_{m-1} \\ P_m \end{bmatrix} \rightarrow P^i_{Sum} = \begin{bmatrix} P_1 + P_2 \\ \vdots \\ P_{m-1} + P_m \end{bmatrix}$$

$$P_{fanout(i-1)} = \begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix}$$

$$P^{i-1} = P_{fanout(i-1)} .* P^i_{Sum} \tag{14}$$

Finally, by obtaining the matrix P^1 from the fanout probability matrix for fanout number 1, the reliability of circuit's output is determined by summing the elements of the matrix P^1.

$$\text{Re}liability = Sum(P) \tag{15}$$

$$P_{failure(out)} = 1 - Sum(P) \tag{16}$$

Reliability of multiple outputs ($R_{multiple}$) is the probability that all outputs are error-free. Alternatively, the fault probability can be represented as $1 - R_{multiple}$, where $R_{multiple}$ is computed after connecting the primary outputs (POs) to the AND gate [12].

$$R_{multiple} = P(R_{out1} \wedge R_{out2} \wedge \ldots \wedge R_{outm}) \tag{17}$$

Here, $\wedge$ represents the AND operator.

For example, consider the circuit in Figure 2. The probability of the output signal for each gate is calculated based on the input matrices and FPM, and a new matrix is assigned to the fanout signal. Its probability matrix is saved as the fanout probability matrix ($P_{fanout,i}$) as described in Equation 18.

$$P_{fanout1} = \begin{bmatrix} 0.95 \\ 0.05 \end{bmatrix}, \quad P_{fanout2} = \begin{bmatrix} 0.95 \\ 0.05 \end{bmatrix},$$

$$P_{fanout3} = \begin{bmatrix} 0.95 \\ 0.05 \\ 0.5 \\ 0.5 \end{bmatrix}$$

$$P_{fanout4} = \begin{bmatrix} 0.95 \\ 0.05 \\ 0.387 \\ 0.613 \end{bmatrix}, \quad P_{fanout5} = \begin{bmatrix} 0.950 \\ 0.050 \\ 0.5144 \\ 0.4856 \end{bmatrix},$$

$$SPM_{out} = \begin{bmatrix} 0.882 \\ 0.588 \\ 0.700 \\ 0.574 \end{bmatrix} \tag{18}$$

By specifying the $P_{fanout,i}$ matrices and SPM_{out} the matrix P^1 is obtained.

$$P^5 = P_{fanout5}.^*SPM_{out} = \begin{bmatrix} 0.8378 \\ 0.0294 \\ 0.3600 \\ 0.2789 \end{bmatrix} \rightarrow P^5_{Sum} = \begin{bmatrix} 0.8672 \\ 0.6389 \end{bmatrix}$$

$$P^4 = P_{fanout4}.^*P^5_{Sum} = \begin{bmatrix} 0.95 \\ 0.05 \\ 0.387 \\ 0.613 \end{bmatrix}.^* \begin{bmatrix} 0.8672 \\ 0.6389 \\ 0.8672 \\ 0.6389 \end{bmatrix} = \begin{bmatrix} 0.8238 \\ 0.0319 \\ 0.3363 \\ 0.3911 \end{bmatrix} \rightarrow P^4_{Sum} = \begin{bmatrix} 0.8558 \\ 0.7274 \end{bmatrix}$$

$$P^3 = P_{fanout3}.^*P^4_{Sum} = \begin{bmatrix} 0.95 \\ 0.05 \\ 0.5 \\ 0.5 \end{bmatrix}.^* \begin{bmatrix} 0.8558 \\ 0.7274 \\ 0.8558 \\ 0.7274 \end{bmatrix} = \begin{bmatrix} 0.8130 \\ 0.0364 \\ 0.4279 \\ 0.3637 \end{bmatrix} \rightarrow P^3_{Sum} = \begin{bmatrix} 0.8494 \\ 0.7916 \end{bmatrix}$$

$$P^2 = P_{fanout2}.^*P^3_{Sum} = \begin{bmatrix} 0.95 \\ 0.05 \end{bmatrix}.^* \begin{bmatrix} 0.8494 \\ 0.7916 \end{bmatrix} = \begin{bmatrix} 0.8069 \\ 0.0396 \end{bmatrix} \rightarrow P^2_{Sum} = [0.8465]$$

$$P^1 = P_{fanout1}.^*P^2_{Sum} = \begin{bmatrix} 0.95 \\ 0.05 \end{bmatrix}.^* \begin{bmatrix} 0.8465 \\ 0.8465 \end{bmatrix} = \begin{bmatrix} 0.8042 \\ 0.0423 \end{bmatrix}$$

P^1 is calculated using the proposed solution, and the circuit's reliability is subsequently determined according to Equation 15.

$$R = Sum(P^1) = 0.846 \tag{19}$$

The time complexity of Algorithm 3 is approximately $O(Size_{out} * F_{out})$. Altogether, to estimate the reliability of the circuit's output based on the proposed method for a circuit with N gates and F fanouts, the worst-case time complexity is approximately $O(Size_{out}(N + F_{out}) \approx O(Size_{out} * N)$, and its time complexity increases linearly with the increase of the basic gates.

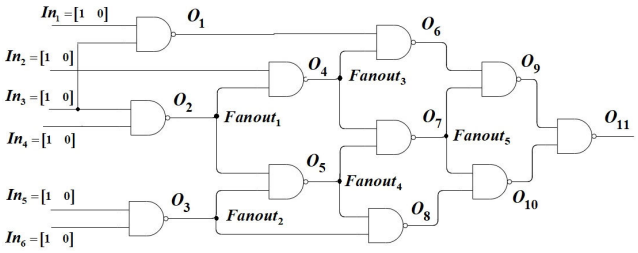

FIGURE 2. Example circuit with error gate 0.05.

Algorithm 3 The Fanout probability matrix algorithm

Input: *The fanout probability matrix is calculated*
Output: *Reliability*
P^n *matrix is calculated by Equation 13*
For $i = F_{out}$*-1 to 1*
P^i_{sum}*matrix is calculated by Equation 14*
P^{i-1} *matrix is calculated by Equations 13*
End
The reliability of circuit's output is calculated by Equations 15

VI. ACCURACY

The proposed method aimed to significantly reduce the computational complexity while considering all cases involving converging paths. However, limitations exist. For instance, when determining the probability of a signal being error-free, we utilized the bit-stream method. This method employs a bit sequence to ascertain the error-free probability of a signal. It inherently possesses two types of errors: quantization error and random permutation [10], [22].

Quantization error occurs due to imprecision in converting probabilities into representations of stochastic binary sequences. This error is reliant on the length of the sequence, specifically the number of bits utilized in a stochastic sequence. When dealing with a sequence of L bits, a real value is rounded to the nearest representation achievable within this sequence (in increments of 1/L). Employing a more significant value for L, mitigates this error. Consequently, the maximum quantization error (using an appropriate rounding method) is constrained to 1/2L [10], [22].

Random permutations constituting an inherent aspect of stochastic computation. Figure 3 illustrates a randomized permutation as an example: the logic operation in Figure 3 (a) produces the desired output value, while the operation in Figure 3 (b) generates an output considered erroneous. Longer sequences generally tend to exhibit better randomization; however, due to the inherent probabilistic nature of random permutations [10], [22].

VII. SIMULATION RESULTS WITH DISCUSSIONS

The simulation results of the proposed methods are presented in this section. The simulations were performed using the MATLAB software environment, assuming that all primary

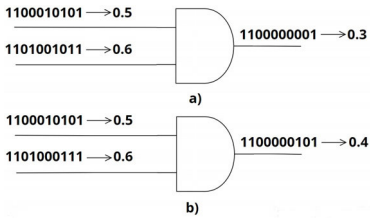

FIGURE 3. Random permutations in stochastic computation: (a) The intended permutation; (b) A permutation leading to an erroneous outcome.

inputs are reliable. The information topology and the list of complete gates in the circuits presented in the ISCAS'85 and EPFL benchmarks were added to MATLAB using the gate-level netlist prepared in [29] and [30]. As in all previous works, gates with fan-in greater than two were replaced by equivalent combinations of two-input gates. In the equivalent combination, all gates other than the gate driving the final output were assumed to be error-free. For example, in a four-input AND gate replaced by a combination of three two-input AND gates [22], only the final AND gate was assumed to be erroneous. All simulations have been run on a 1.8-GHz Intel i7-8550 with 12-GB memory. The length of bitstream sequences was set to L = 10000.

Similar to previous studies in this field [12], [16], our simulation results are compared with the Monte Carlo (MC) method to verify the accuracy of the presented method. For this paper, the circuit is examined using 100,000 input vectors. The percentage relative error is obtained using the following Equation 20 presented in [24]:

$$\text{Relative error} = \frac{Measured - MC}{MC} * 100 \qquad (20)$$

We presented a new fanout matrix to increase accuracy and eliminate the number of impossible and repetitive states caused by reconvergent paths. However, the issue is that the circuit may have too many fanouts, while all of them do not significantly impact the error probability of the circuit output. These fanouts should be identified and excluded from the calculations. To address this issue, we utilized the first-depth algorithm to identify effective fanouts for each output, and only these fanouts were included in the calculations.

First, to justify our previous assumption, we experimented using a c432 circuit with 216 gates. The aim was to demonstrate the difference in the output probability matrix obtained for two cases: considering all fanouts with Monte Carlo simulation and the effective fanouts obtained based on Algorithm 1 with the proposed method. The results are presented in Table 3 with gate error probabilities 0.001,0.05 and 0.1. Comparing the obtained results with the assumed results of considering effective fanouts, the average Relative error between the results is about 0.3567%. Therefore, these effective fanouts assist in reducing the computational workload.

Based on Section VI, we obtained and compared the reliability of ISCAS'85 benchmark circuits to explore the impact of bitstream method error on the accuracy of our method. We assumed an error probability of 0.001 for the gate and evaluated different bit sequences accordingly. Table 4 displays simulation results for various bit sequences. The average relative error at l=10000 is approximately 0.0324, indicating a suitable bit sequence length for estimating the probability of a signal being one in the presence of circuit errors. Additionally, the runtime increases linearly as the length of the bit sequence increases.

Figure 4 depicts the reliability of the C432 circuit with changes in the error probability of the gates, considering three cases: without the assumption of reconvergent paths,

TABLE 3. Comparison of monte carlo simulation results for all fanout with the proposed approach (gate error = 0.001,0.05,0.1).

Outputs	Number of Effective Fanout	Size Output	Relative error %		
			$\varepsilon = 0.001$	$\varepsilon = 0.05$	$\varepsilon = 0.1$
O223	0	1	0.024	0.033	0.021
O329	10	4	0.252	0.243	0.361
O370	38	16	0.565	0.453	0.671
O421	43	4	0.129	0.331	0.242
O430	44	4	0.225	0.143	0.239
O431	46	16	0.677	0.566	0.581
O432	45	8	0.392	0.359	0.449
All output	53	1024	0.401	0.699	0.506

TABLE 4. Simulation results of ISCAS'85 benchmarks for different value of bit sequences (gate error = 0.001).

Circuits	L=100		L=1000		L=10000		L=100000	
	Relative error %	Time (s)	Relative error %	Time (s)	Relative error %	Time (s)	Time (s)	Reliability
C432	0.5621	0.115	0.0625	0.123	0.0104	0.2480	2.35	0.9606
C499	0.0214	0.340	0.0107	0.376	0.0107	0.6130	2.88	0.9332
C880	0.0346	0.310	0.0231	0.345	0.0112	0.7460	3.534	0.8662
C1335	0.1114	0.367	0.0248	0.337	0.0124	0.8000	4.278	0.8078
C1908	0.1468	0.272	0.0440	0.308	0.0101	0.5990	1.789	0.6812
C2670	0.0922	0.556	0.0615	0.644	0.0154	0.9310	4.393	0.6505
C3540	0.2456	1.167	0.0982	1.209	0.0164	3.2300	6.357	0.6107
C5315	0.2502	1.68	0.0893	1.906	0.0179	3.5250	7.218	0.5596
C6288	1.0888	1.231	0.7538	1.392	0.1675	3.5230	8.567	0.1194
C7552	0.5872	2.302	0.2936	2.524	0.0734	4.7500	11.154	0.2725
Average	**0.3140**	**0.8340**	**0.1462**	**0.9164**	**0.0324**	**1.8965**	**5.2520**	

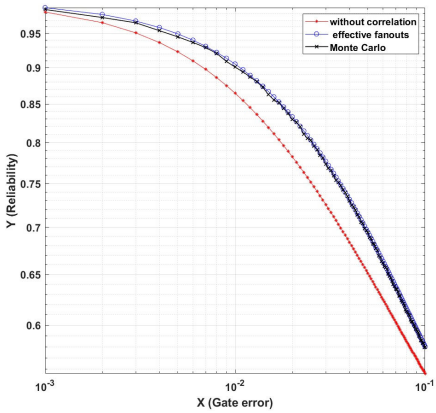

FIGURE 4. The reliability of the C432 circuit with the changes in the error probability of the gates.

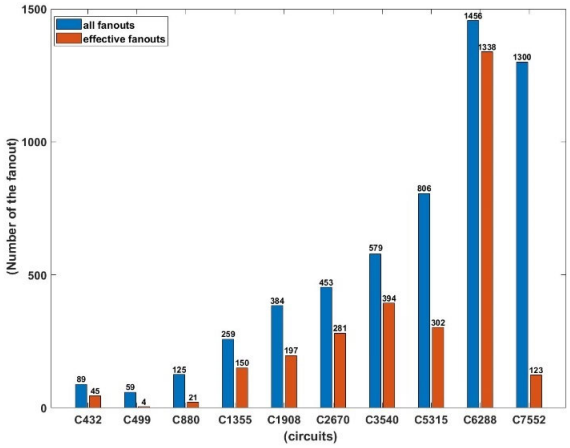

FIGURE 5. The total number of fanouts of a circuit and the number of effective fanouts.

all fanouts ending in the output (O432) with Monte Carlo simulation, and the effective fanouts of that output with the proposed method. Two graphs assuming all fanouts and effective fanouts are superimposed and have a slight difference, where the average difference of the results is about 0.0013.

Figure 5 compares the number of fanouts in two modes for the ISCAS'85 circuits. These modes include the total number of the circuit's fanouts and the number of fanouts for the circuit output that ends with the highest fanouts based on Algorithm 1. This comparison demonstrates that it's not necessary to consider all the circuit fanouts to calculate the reliability of an output. With Algorithm 1, you can eliminate some of these fanouts from the calculations without affecting the accuracy of the computations.

Figure 6 compares the results obtained for two circuits, C3540 and C6288, from ISCAS'85 benchmark. These results are provided for three error probabilities of gates: 0.01, 0.05, and 0.1. Figure 6 compares our method with the Monte Carlo Simulation for different outputs of these two circuits to verify the accuracy of the presented method. The purpose of this comparison is to demonstrate the efficiency of our method when dealing with circuits with a high number of fanouts. In addition to the reliability analysis, the runtime for computing the probability of each output is displayed, providing an indication of the efficiency of Algorithm 3 when confronted with digital circuits that have a significant number of fanouts.

As discussed in Section IV, when calculating the output signal's probability, there are duplicate entries that can be used only once due to the consideration of the fanouts' states. We proposed an algorithm to reduce the number of calculations and the output probability matrix size as much as possible by considering all fanouts states. Figure 7 shows the maximum size of the output probability matrix for the ISCAS'85 benchmark circuits. For example, in circuit C3540, the size of the output probability matrix, without applying the gate output probability calculation algorithm and considering all states, is equal to 2^{25}. By considering the algorithm, the size of this matrix is reduced to 4096 (2^{12}). In other words, the output probability matrix with size 2^{25} is repeated as $8192(2^{25-15})$. By avoiding the recalculation of these repeated values, the volume of calculations is significantly reduced.

The obtained simulation results of estimating the digital circuit's reliability in the presence of multiple errors are investigated in this section. Table 5 shows the evaluation outcomes of various ISCAS'85 combinational logic benchmarks, with the gate error rate set at 0.001 and 0.0001. In this context, the average relative error measures below 1%, and the proposed method adeptly manages signal correlations introduced by reconvergent fanouts. The proposed method achieved an impressively low average calculation error of 0.492, outperforming the other methods with average errors of 0.956 and 1.47, respectively. Additionally, it demonstrated superior computational efficiency, with an average processing time of 0.9483, while the other methods required more time, with averages of 4.23 and 4.207, respectively. Notably, the reliability estimation results from the proposed method slightly outperformed those of [12] and [25].

One of the crucial aspects of the reliability calculation method is its efficiency in handling large circuits, aiming to achieve both accurate results and reasonable computation

TABLE 5. Simulation results of ISCAS'85 benchmarks by the proposed method, [12] and [25].

Circuits	Characteristics				Reliability Circuit		Relative error % (error gate 0.05)			Time (s)			
	Gates	Inputs	Outputs	Fanouts	error gate 10^{-3}	error gate 10^{-4}	Proposed method	[25]	[12]	Proposed method: Effective fanout	Proposed method: Estimate probability	[25]	[12]
C432	216	36	7	89	0.9605	0.9957	0.35	0.26	4	0.106	0.142	0.39	0.9
C499	246	41	32	59	0.9333	0.9912	0.23	0.33	2	0.361	0.252	0.76	1.1
C880	435	60	26	125	0.8663	0.9781	0.41	0.61	0.5	0.333	0.413	1.18	2
C1355	590	41	32	259	0.8079	0.9770	0.12	0.88	1.2	0.359	0.441	1.55	1.3
C1908	1057	33	25	384	0.6813	0.9604	0.53	0.55	0.4	0.297	0.302	2.34	2
C2670	1400	157	64	453	0.6506	0.9578	0.57	1.46	0.1	0.383	0.548	3.99	2
C3540	1456	50	22	579	0.6106	0.9525	0.69	1.2	4	0.52	2.71	4.1	8
C5315	2973	178	123	806	0.5597	0.9451	0.86	1.05	1.2	0.985	2.54	8.13	8
C6288	2416	32	32	1456	0.1192	0.8056	0.73	1.79	0.3	1.02	2.503	6.41	10
C7552	4042	207	108	1300	0.2727	0.8690	0.43	1.43	1	1.302	3.448	13.22	7
Average							0.492	0.956	1.47	0.9483		4.207	4.23

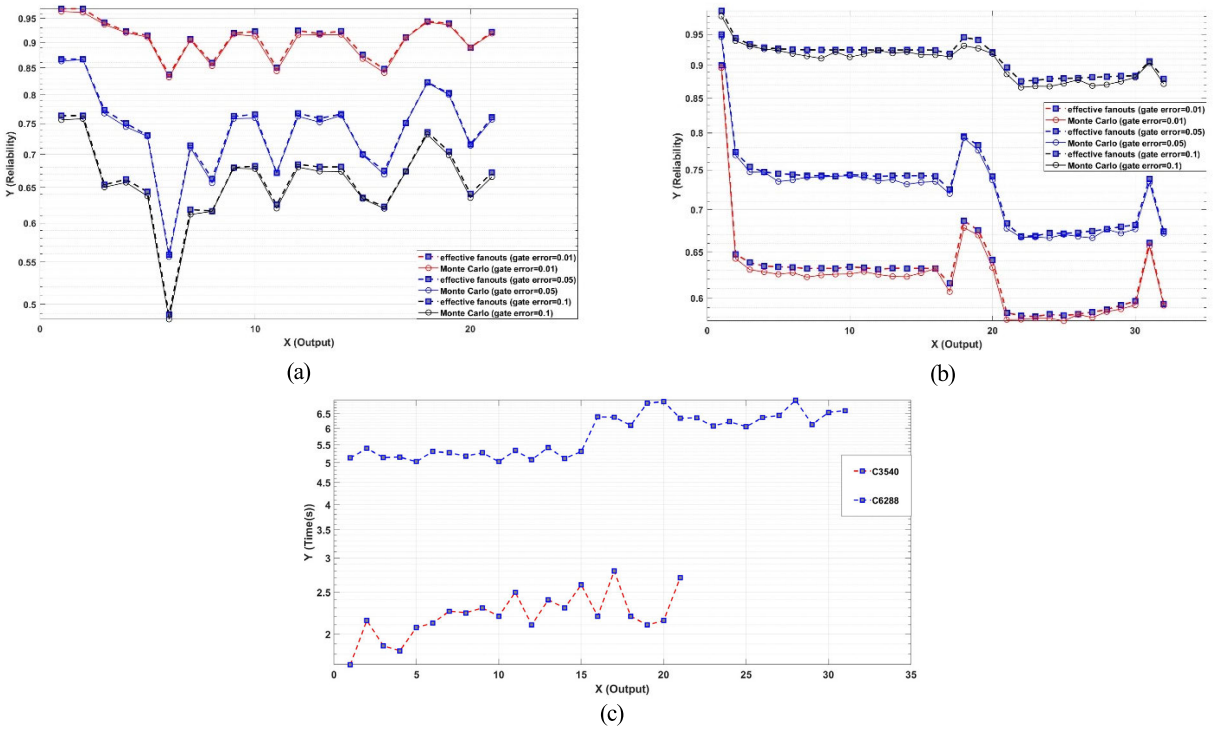

FIGURE 6. Comparative Analysis: Our Method vs. Monte Carlo Simulation for Various Outputs (Gate Errors: 0.01, 0.05, 0.1). (a) Circuit C3540, (b) Circuit C6288, (c) Probability Computation Runtime for Circuits C3540 and C6288.

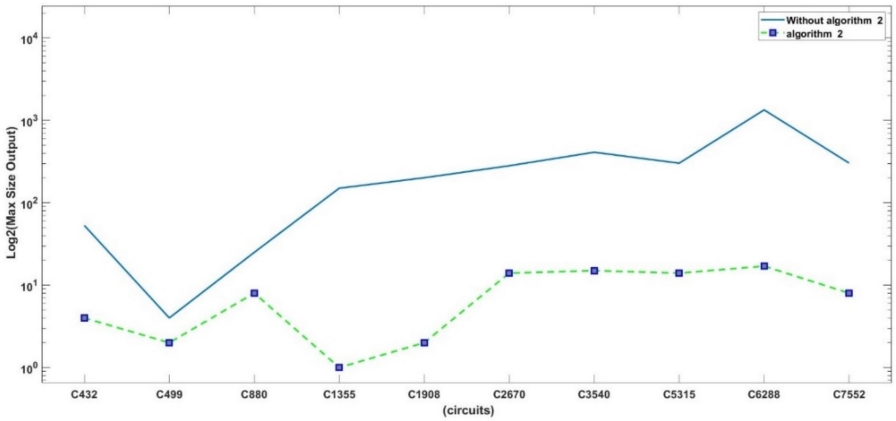

FIGURE 7. The maximum size of the output probability matrix.

time while minimizing the computational workload. Many existing approaches in this field lack efficiency when dealing with large circuits, efficiency when dealing with large circuits, making it impractical to calculate the circuits' reliability. To showcase the potential of the proposed method, the relative error of calculations based on Monte Carlo simulation is reported for the EPFL benchmark circuit in Table 6. The results in Table 6 demonstrate that the proposed method achieves good accuracy and acceptable computation time, even for circuits with a substantial number of gates. For instance, the Memory Controller circuit is approximately 10 times larger than circuit C7552, yet the processing time is only 30 times longer. The proposed method exhibits a speed-up factor of 258 for the Memory Controller circuit, significantly outperforming other circuits. These findings suggest that the proposed method is more scalable and advantageous for estimation of reliability of larger circuits.

In Table 7, we've compiled data on the average relative error, mean runtime, and runtime complexity of our proposed approach, [12] and [25]. The table shows that our proposed approach, when combined with mitigation techniques, exhibits a computational complexity of approximately $O(N^{1.04})$, with N representing the number of circuit gates. A comparison of the results in Table 7 highlights that

TABLE 6. Comparison simulation results of EPFL benchmarks (error gate 0.05) by the proposed method with Monte Carlo Simulations.

Circuits	Characteristics			Relative error %	Time	
	Gates	Inputs	Outputs		Prop.method	MC
Adder	1020	256	129	0.39	0.4s	5min
B- shifter	3336	135	128	0.23	3.73s	15min
Divisor	44762	128	128	1.54	73s	10h
Log2	32060	32	32	1.02	52s	7h
Max	2865	512	130	0.31	4.3s	10min
Multiplier	27062	128	128	0. 92	39s	5h
Sine	5416	24	25	0.76	4.91s	30min
S-root	24618	128	64	0.73	33s	6h
Square	18484	64	128	0.11	26s	3h
Memory controller	46836	1204	1231	1.038	97s	10h
Average				0.6048	33.33s	15120s

TABLE 7. Comparative analysis of accuracy, runtime, and runtime complexity for Ref [12], Ref [25], and the proposed method.

Approach	Mean Relative error %	Mean run time(s)	Runtime complexity
Proposed method	0.492	0.9483	$O(N^{1.04})$
Ref [25]	0.956	4.207	$O(N^{1.07})$
Ref [12]	1.47	4.23	$O(N^{1.53})$

TABLE 8. Simulation results of ISCAS'85 benchmarks by the proposed method with the probability of input error and gate error is 10^{-3}.

Circuits	Reliability	Time (s)	
	error gate 10^{-3}	Proposed method	
		Effective fanout	Estimate probability
C432	0.9621	0.16	0.542
C499	0.9038	0.61	0.6 52
C880	0.8516	0.53	0.413
C1355	0.7802	0.59	1.441
C1908	0.6807	0.97	1.302
C2670	0.6405	0.83	1.748
C3540	0.6089	0.52	2.71
C5315	0.5340	1.985	2.54
C6288	0.1182	1.32	2.503
C7552	0.2707	1.52	4.448

our proposed approach outperforms other methods in both speed and accuracy.

In Section III, the input fanouts have been treated as independent signals under the assumption of error-free inputs to the circuit. However, if there is a probability of errors in the circuit inputs, then these input fanouts are also considered effective. Consequently, Table 8 displays the simulation result of ISCAS'85 benchmarks, assuming erroneous the circuit's primary inputs. We have taken into account an error probability of 0.001 for both the primary inputs and gate. The runtime reflects a linear relationship between the computational complexity and the increase in the number of fanouts.

VIII. CONCLUSION

In this paper, a precise and scalable method for reliability estimation in combinational digital circuits is presented. The challenges of error propagation are addressed by proposing an error propagation probability model for each gate. Additionally, an efficient approach is introduced that utilizes a new fanout matrix to handle the issue of reconvergent fanouts and applies a first-depth algorithm to determine the effective fanouts for each output. To ensure accurate reliability estimation, probabilities obtained for each fanout are included by defining a fanout probability matrix for each effective fanout. Furthermore, a new method is proposed at each calculation stage, aiming to minimize computational complexity and adapt it for large circuits with a significant number of fanouts. Thorough simulations on combinational benchmark circuits, including ISCAS 85 and EPFL benchmarks, were conducted to validate the effectiveness and scalability of the approach. The results show less than 1% average relative error in reliability estimation, and the simulation results indicate that the proposed approach is 4.5× faster than approaches [12] and [25].

This paper focuses on estimating the reliability of digital combinational circuits. However, in reality, sequential circuits constitute a significant portion of digital circuits. Therefore, in future work, we aim to develop the presented method for estimating the reliability of sequential circuits. Also, estimating the probability of errors in the circuit output and considering the electrical and latching-windows masking issues in digital circuits are among the objectives of the proposed method.

REFERENCES

[1] T. Rejimon, K. Lingasubramanian, and S. Bhanja, "Probabilistic error modeling for nano-domain logic circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 1, pp. 55–65, Jan. 2009.
[2] Y. S. Dhillon, A. U. Diril, A. Chatterjee, and A. D. Singh, "Analysis and optimization of nanometer CMOS circuits for soft-error tolerance," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 5, pp. 514–524, May 2006.
[3] A. H. El-Maleh and K. A. K. Daud, "Simulation-based method for synthesizing soft error tolerant combinational circuits," *IEEE Trans. Rel.*, vol. 64, no. 3, pp. 935–948, Sep. 2015.
[4] M. Raji, M. A. Sabet, and B. Ghavami, "Soft error reliability improvement of digital circuits by exploiting a fast gate sizing scheme," *IEEE Access*, vol. 7, pp. 66485–66495, 2019.
[5] S. K. Saha, "Compact MOSFET modeling for process variability-aware VLSI circuit design," *IEEE Access*, vol. 2, pp. 104–115, 2014.
[6] M. Ferreira Pontes, C. Farias, R. Schvittz, P. Butzen, and L. Da Rosa Jr., "Survey on reliability estimation in digital circuits," *J. Integr. Circuits Syst.*, vol. 16, no. 3, pp. 1–11, Dec. 2021.
[7] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo Method*. Hoboken, NJ, USA: Wiley, 2016.
[8] B. Liu and L. Cai, "Monte Carlo reliability model for single-event transient on combinational circuits," *IEEE Trans. Nucl. Sci.*, vol. 64, no. 12, pp. 2933–2937, Dec. 2017.
[9] C. R. Farias, R. B. Schvittz, T. R. Balen, and P. F. Butzen, "Evaluating soft error reliability of combinational circuits using a Monte Carlo based method," in *Proc. IEEE 23rd Latin Amer. Test Symp. (LATS)*, Sep. 2022, pp. 1–6.
[10] J. Han, H. Chen, J. Liang, P. Zhu, Z. Yang, and F. Lombardi, "A stochastic computational approach for accurate and efficient reliability evaluation," *IEEE Trans. Comput.*, vol. 63, no. 6, pp. 1336–1350, Jun. 2014.
[11] J. Xiao, Q. Ji, Q. Shen, J. Jiang, Y. Huang, and J. Lou, "Accelerating stochastic-based reliability estimation for combinational circuits at RTL using GPU parallel computing," *Int. J. Intell. Syst.*, vol. 37, no. 11, pp. 8309–8326, Nov. 2022.
[12] S. Bathla and V. Vasudevan, "A framework for reliability analysis of combinational circuits using approximate Bayesian inference," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 31, no. 4, pp. 543–554, Apr. 2023.

[13] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Probabilistic transfer matrices in symbolic reliability analysis of logic circuits," *ACM Trans. Design Autom. Electron. Syst.*, vol. 13, no. 1, pp. 1–35, Jan. 2008.
[14] J. Xiao, W. Lee, J. Jiang, and X. Yang, "Circuit reliability estimation based on an iterative PTM model with hybrid coding," *Microelectron. J.*, vol. 52, pp. 117–123, Jun. 2016.
[15] J. Xiao, J. Jiang, X. Li, Y. Huang, X. Yang, Z. Shi, and J. Lou, "A novel trust evaluation method for logic circuits in IoT applications based on the E-PTM model," *IEEE Access*, vol. 6, pp. 35683–35696, 2018.
[16] J. Han, H. Chen, E. Boykin, and J. Fortes, "Reliability evaluation of logic circuits using probabilistic gate models," *Microelectron. Rel.*, vol. 51, no. 2, pp. 468–476, Feb. 2011.
[17] O. Keszocze, "BDD-based error metric analysis, computation and optimization," *IEEE Access*, vol. 10, pp. 14013–14028, 2022.
[18] N. Mohyuddin, E. Pakbaznia, and M. Pedram, "Probabilistic error propagation in a logic circuit using the Boolean difference calculus," in *Advanced Techniques in Logic Synthesis, Optimizations and Applications*, 2011, pp. 359–381.
[19] M. Kvassay, E. Zaitseva, V. Levashenko, and J. Kostolny, "Reliability analysis of multiple-outputs logic circuits based on structure function approach," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 3, pp. 398–411, Mar. 2017.
[20] V. V. Hamiyati and A. Peiravi, "Reliability evaluation of digital circuits using probabilistic signal flow graphs," *Tabriz J. Elect. Eng.*, vol. 50, no. 2, pp. 679–689, 2020.
[21] J. T. Flaquer, J. M. Daveau, L. Naviner, and P. Roche, "Fast reliability analysis of combinatorial logic circuits using conditional probabilities," *Microelectron. Rel.*, vol. 50, nos. 9–11, pp. 1215–1218, Sep. 2010.
[22] H. Jahanirad, "CC-SPRA: Correlation coefficients approach for signal probability-based reliability analysis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 4, pp. 927–939, Apr. 2019.
[23] W. Ibrahim and H. Ibrahim, "Multithreaded and reconvergent aware algorithms for accurate digital circuits reliability estimation," *IEEE Trans. Rel.*, vol. 68, no. 2, pp. 514–525, Jun. 2019.
[24] Z. Wang, G. Zhang, J. Ye, J. Jiang, F. Li, and Y. Wang, "Accurate reliability analysis methods for approximate computing circuits," *Tsinghua Sci. Technol.*, vol. 27, no. 4, pp. 729–740, Aug. 2022.
[25] S. Zhan and C. Chen, "A hybrid method for signal probability and reliability estimation with combinational circuits," *Integration*, vol. 87, pp. 275–283, Nov. 2022.
[26] S. Cai, B. He, S. Wu, J. Wang, W. Wang, and F. Yu, "An accurate estimation algorithm for failure probability of logic circuits using correlation separation," *J. Electron. Test.*, vol. 38, no. 2, pp. 165–180, Apr. 2022.
[27] E. Esmaieli Sartakhti, Y. Sedaghat, and A. Peiravi, "A model for probabilistic fault propagation with the approach of effective fanouts in the logic circuits," in *Proc. 31st Int. Conf. Electr. Eng. (ICEE)*, May 2023, pp. 1–6.
[28] M. T. Goodrich, R. Tamassia, and D. M. Mount, *Data Structures and Algorithms in C++*. Hoboken, NJ, USA: Wiley, 2011.
[29] [Online]. Available: https://pld.ttu.ee/~maksim/benchmarks/iscas85/bench/
[30] [Online]. Available: https://www.epfl.ch/labs/lsi/page-102566-en-html/benchmarks/

ESFANDIAR ESMAIELI received the B.S. degree in electrical engineering from the University of Sistan and Baluchestan, Zahedan, Iran, in 2013, and the M.S. degree in electrical engineering from the Shahid Bahonar University of Kerman, Kerman, Iran, in 2016. He is currently pursuing the Ph.D. degree with the Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran. His research interests include reliability of digital circuits, fault tolerance, and fault injection.

ALI PEIRAVI received the B.S. degree in electrical engineering from the University of Pittsburgh, Pittsburgh, PA, USA, in 1976, the M.S. degree in electrical engineering from the University of California, Berkeley, Berkeley, CA, USA, in 1978, and the Ph.D. degree in electrical engineering from the University of California, Irvine, Irvine, CA, USA, in 1984. In 1984, he joined the Department of Electrical Engineering, Ferdowsi University of Mashhad, where he is currently a Full Professor. His research interests include reliability, micromagnetic simulations, noise immunity, low power, and wireless sensor networks.

YASSER SEDAGHAT received the B.S. degree in computer engineering from the Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran, in 2004, and the M.S. and Ph.D. degrees in computer engineering from the Sharif University of Technology, Tehran, Iran, in 2006 and 2011, respectively. In 2011, he joined the Department of Computer Engineering, Ferdowsi University of Mashhad, where he is currently an Assistant Professor. His research interests include reliability, fault tolerance, fault injection, safety-critical systems, and distributed embedded systems.