

## Review Article

# Gate-Level Circuit Reliability Analysis: A Survey

**Ran Xiao and Chunhong Chen**

*Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada N9B 3P4*

Correspondence should be addressed to Chunhong Chen; [cchen@uwindsor.ca](mailto:cchen@uwindsor.ca)

Received 25 April 2014; Accepted 7 June 2014; Published 10 July 2014

Academic Editor: Yu-Cheng Fan

Copyright © 2014 R. Xiao and C. Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Circuit reliability has become a growing concern in today's nanoelectronics, which motivates strong research interest over the years in reliability analysis and reliability-oriented circuit design. While quite a few approaches for circuit reliability analysis have been reported, there is a lack of comparative studies on their pros and cons in terms of both accuracy and efficiency. This paper provides an overview of some typical methods for reliability analysis with focus on gate-level circuits, large or small, with or without reconvergent fanouts. It is intended to help the readers gain an insight into the reliability issues, and their complexity as well as optional solutions. Understanding the reliability analysis is also a first step towards advanced circuit designs for improved reliability in the future research.

## 1. Introduction

As CMOS technology keeps scaling down to their fundamental physical limits, electronic circuits have become less reliable than ever before [1]. The reason is manifold. First of all, the higher integration density and lower voltage/current thresholds have increased the likelihood of soft errors [2, 3]. Secondly, process variations due to random dopant fluctuation or manufacturing defects have negative impacts on circuit performance and may cause circuits to malfunction [1]. These physical-level defects would statistically lead to probabilistic device characteristics. Also, some emerging nanoscale electronic components (such as single electron devices) have demonstrated their nondeterministic characteristics due to uncertainty inherent in their operation under high temperature and external random noise [4, 5]. This may further degrade the reliability of future nanoelectronic circuits. Thus, circuit reliability has been a growing concern in today's micro- and nanoelectronics, leading to the increasing research interest in reliability analysis and reliability-oriented circuit design.

For any reliability-aware architecture design, it is indispensable to estimate the reliability of application circuits both accurately and efficiently. However, analyzing the reliability (or the error propagation) for logic circuits could be computationally expensive in general (see Section 1.3 for

details). Some approaches have been reported in literature, which tackle the problem either analytically or numerically (by simulation). The contribution of this paper is to provide an extensive overview and comparative study on typical reliability estimation methods with our simulation results and/or results reported in literature.

We first review the key concepts in reliability analysis and its role in circuit design and then describe and evaluate several existing mainstream approaches for reliability analysis by looking at their accuracy, efficiency, and flexibility. Examples and simulation results are also given in order to show their advantages and disadvantages. Finally, we provide some useful suggestions on how to choose an appropriate reliability analysis method under different circumstances, along with some remarks on possible future work.

**1.1. Signal Probability and Reliability.** The probability of a logic signal  $s$  is by default defined as the probability of the signal being logic "1" and is expressed as  $P_s = \Pr\{s = "1"\}$ . The reliability of the probabilistic signal  $s$  is defined as the probability that its value is correct (i.e., it is equal to its error-free value) and is expressed as  $r_s = \Pr\{s = \text{its error-free value}\}$ . In gate-level design, the output signal of a gate may become unreliable due to its unreliable inputs and/or errors of gate itself. If we use the classical *von*

*Neumann* model [6] for gate errors, any gate can be associated independently with an error probability  $\varepsilon_i$ . In other words, the gate is modeled as a binary symmetric channel that generates a bit flip (from  $0 \rightarrow 1$  or  $1 \rightarrow 0$ ) by mistake at its output (known as *von Neumann* error [6]) symmetrically with the same probability. Thus, each gate  $i$  in the circuit has an independent gate reliability  $r_i = 1 - \varepsilon_i$ , which is assumed to be localized and statistically stable. Also, it is reasonable to assume that the error probability for any gate falls within  $[0, 0.5]$  (or  $r_i \in [0.5, 1]$ ).

The reliability for a combinational logic circuit (denoted by  $R_C$ ) is defined as the probability of the correct functioning at its outputs (i.e., the joint signal reliability of all primary outputs). This reliability can be generally expressed as a function of gate reliabilities in the circuit (denoted by  $\mathbf{r} = \{r_1, r_2, \dots, r_{N_g}\}$ , where  $N_g$  is the number of gates), as well as signal probabilities of all primary inputs (denoted by  $\mathbf{P}_{in} = \{P_{in1}, P_{in2}, \dots, P_{inN_{in}}\}$ , where  $N_{in}$  is the number of primary inputs), that is,

$$R_C = \Pr \{\text{all outputs are correct}\} = f(\mathbf{r}, \mathbf{P}_{in}), \quad (1)$$

where the function  $f$  depends on the topology of the circuit under consideration. Note that the primary inputs are assumed to be fully reliable ( $r_s = 1$  if  $s$  is a primary input). Under a particular case where all primary input probabilities are a constant (say 0.5),  $R_C$  turns out to be a function of  $\mathbf{r}$  only.

It is worth noting that gate errors may come from either external noises (thermal noise, crosstalk, or radiation) [3] or inherent device stochastic behaviors [4]. In literature, the term “soft error” is used to emphasize the temporariness of the errors due to random external noises (e.g., glitches). In this paper, however, a more general term of *von Neumann* gate error model is used instead, as the probabilistic feature of gates is expected to exist widely and independently throughout the circuit. This differs from single-event upsets due to soft errors, where external noises are usually correlated temporally and spatially. In other words, our focus is the error propagation in combinational networks, where the gate-level logic masking is considered. For instance, some logic errors may not affect (or propagate to) final outputs if they occur in a nonsensitized portion of the circuit. Identifying these nonsensitized gates would be critical for reliability estimation and improvement.

**1.2. Role of Reliability Analysis.** In order to guide the IC design for reliable logic operations, it is required to develop tools that can accurately and efficiently evaluate circuit reliability, which is also a first step towards reliability improvement. However, reliability analysis is a nontrivial task due to the large size of IC circuits as well as the complexity of signal correlation and probability/reliability propagation within the circuit (as will become clear later in this paper). On the other hand, circuit reliability can be generally improved by increasing the gate reliabilities. This can be done by using redundant components. Classic redundancy techniques such as TMR [5] or NAND-multiplexing [7] achieve this by systematically replicating logic gates (other than sizing up the transistors) at the cost of increased area and power

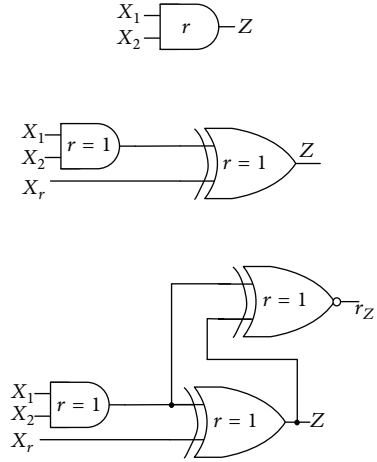


FIGURE 1: An AND gate and its equivalent circuit.

dissipation. One of the key issues in this context is to select the most critical (in terms of reliability and cost) components (or logic gates) in the circuit and improve the circuit reliability by increasing the robustness of only a few gates. In order to detect these critical gates, multiple cycles of reliability analysis are usually conducted for the whole circuit. In a more general term, accurate and efficient reliability analysis can provide a guideline for future reliability-oriented architecture design.

**1.3. Complexity of Gate-Level Reliability Analysis.** It is understood that the problem of determining whether the signal probability at a given node is nonzero is equivalent to the Boolean satisfiability (SAT) problem [8], a problem of determining whether there exists an interpretation that satisfies a given Boolean formula. A Boolean formula is called satisfiable if the variables of this given formula can be assigned in such a way as to make the formula evaluate to TRUE (3). The SAT has been proved to be an NP-complete problem (see [9]). The problem of computing all signal probabilities in a circuit can be formulated as a random satisfiability problem, which is to determine the probability that a random assignment of variables will satisfy a given Boolean formula [9]. The random satisfiability problem lies in a class of problems, called #P-complete, which is conjectured to be even harder than NP-complete. In the following, we show that the reliability evaluation problem is equivalent to the signal probability calculation problem and thus prove that it is also a #P-complete problem.

Let us consider a two-input AND gate ( $Z = X_1 X_2$ ) which has the gate reliability  $r$ , as shown in Figure 1. We first add an extra XOR gate at the output, as well as an extra input  $X_r$ , with an assumption that both the XOR gate and original AND gate are error-free. The signal probability of this extra input is equal to the original gate error rate  $\varepsilon$  (i.e.,  $P_{X_r} = \Pr\{X_r = "1"\} = 1 - r$ ). This ensures that the output  $Z$  of this extra XOR gate is equivalent to the original output of the AND gate.

For a combinational logic circuit, we first duplicate the whole circuit. In the original circuit, we make each gate error-free in order to compute the correct value at primary

outputs. For the duplicated one, we extract the reliability of each gate using the aforementioned method (as a result, all gates are also error-free in the duplicated circuit and the gates' number is doubled). Then, we add 2-input XNOR gates for each pair of corresponding primary outputs in the original and duplicated circuits. Thus, the output reliability can be expressed as the signal probability at the output of the XNOR gates. By doing so (i.e., duplicating the circuit and extracting gate reliabilities), we see that the reliability estimation of original circuit is equivalent to the problem of computing the signal probabilities of the transformed circuit.

For a combinational logic circuit with  $N_{in}$  primary inputs,  $N_{out}$  primary outputs, and  $N_g$  logic gates, the problem of evaluating the signal reliability of all primary outputs and their joint reliability (i.e., the overall circuit reliability  $R_C$ ) can be solved by exhaustively calculating all  $2^{(N_{in}+N_g)}$  scenarios. In each scenario, the expected (correct) output and actual output values need to be calculated with the complexity of  $O(N_g)$ . The total complexity is then  $O(N_g \cdot 2^{N_{in}+N_g})$ . As circuits become very large, it would be difficult or even impossible to perform the exact analysis of the reliability due to the exponential complexity. Usually, some tradeoff has to be made between the accuracy and efficiency for reliability analysis.

In order to tackle this issue, a number of different approaches have been reported in literature, including probabilistic transfer matrix (PTM) method [10–12], Bayesian networks (BN) [13–15], Markov random field (MRF) [16–20], Monte Carlo (MC) simulation, testing-based method [3], stochastic computation model (SCM) [2, 21], probabilistic gate model (PGM) [22–25], observability-based analysis [26], Boolean difference-based error calculator (BDEC), and correlation coefficient method- (CCM-) based approaches [8, 26–28]. In the following, we overview some of these approaches and analyze their pros and cons in terms of accuracy, efficiency, and flexibility with simulation results.

## 2. Probabilistic Transfer Matrix (PTM) Method

An accurate analytical model for reliability analysis problem is based on the probabilistic transfer matrices (PTMs), which compute the circuit output reliability for all input patterns [10, 11]. This computational framework begins with the definition of a probability matrix which is used to represent the probability of a logic gate's output for each input pattern. For instance, the probability matrix representation for a two-input NAND logic gate is shown in Figure 2, where each column of the matrix  $\mathbf{M}_g$  represents the probability of the gate output  $Z$  being "0" or "1" for all different input patterns (i.e.,  $X_1X_2 = "00", "01", "10", \text{ and } "11"$ ). For example, the element  $\mathbf{M}_{11} = \Pr\{Z = 0 \mid X_1X_2 = 00\} = 1 - r$ , where  $r$  is the gate reliability. In general, the probability matrix for an  $n$ -input 1-output gate is a  $2^n \times 2$  matrix.

For a circuit, all gate probability matrices shall be combined together to construct the PTM of the whole circuit. More specifically, the serial and parallel connections of gates correspond to a matrix product and tensor product [10],

respectively. The fanout behavior is represented by explicit fanout gates, where a 1-input  $m$ -output fanout gate is simply mimicked by a 1-input  $m$ -output buffer gate. A fault-free circuit has an ideal transfer matrix (ITM), where the correct value of the output occurs with the probability of 1. This means that, in each row of the PTM, there is single "1" for the correct output value and there are "0"s for other output combinations. The circuit reliability (i.e., the probability of outputs being correct) is evaluated by comparing its PTM and ITM.

The process of combining gate probability matrices implicitly takes into account the signal dependency between gates by considering the underlying joint and conditional probabilities within the circuit. As a result, the calculation of the circuit PTM is exact. However, the limited scalability is often a price that has to be paid for this computational framework to capture complex circuit behaviors. Consider a combinational logic circuit with  $N_{in}$  primary inputs,  $N_{out}$  primary outputs, and  $N_g$  logic gates. The circuit PTM is a matrix with  $2^{N_{in}}$  rows and  $2^{N_{out}}$  columns (i.e.,  $2^{N_{in}} \times 2^{N_{out}}$ ), which contains the transition probability from all input combinations toward all output combinations. In other words, its space complexity is  $O(2^{N_{in}+N_{out}})$ . This exponential space requirement is the main bottleneck of PTM approach. Particularly, for a computer with 2 GB memory, the maximum size of the circuit that can be handled is limited to 16 input/output signals. By utilizing some advanced computation methods (such as algebraic decision diagrams (ADDs) and encoding [10, 11]), the signal width may be extended up to ~50, where the signal width is defined as the largest number of signals at any level in the circuit. Unfortunately, this limit is still computationally unacceptable in the real world for large-scale benchmark circuits (e.g., C2670 which has 157 inputs and 64 outputs). Nonetheless, for small circuits, the PTM is a very good analytical method, as it provides exact results within a reasonable runtime and shows the probabilistic behavior of unreliable logic gates.

Also, this approach can serve as the foundation of many other heuristic approaches by providing other important information such as signal probabilities and observability, with the capability of analyzing the effect of electrical masking on error mitigation as well. For instance, in [10], the observability of a gate  $g$  is defined as the ratio of the error probability of the whole circuit and the error probability  $\varepsilon_i$  of this gate, that is,  $(1 - R_C(\varepsilon_i))/\varepsilon_i$ , where  $R_C(\varepsilon_i)$  is the circuit reliability when the only unreliable gate is  $i$ th gate (with all other gates being error-free). Clearly, the gate with highest observability can be regarded as the most susceptible, meaning that it will impact (or decrease) the circuit reliability the most. It should be noted that this only represents the simplest case where only single gate failure is considered. In most real cases, however, the gate observabilities may not be independent, and thus the joint observabilities usually need to be considered instead.

The detailed algorithm with the PTM is summarized as follows.

*Step 1.* Levelize the circuit; compute PTMs of each logic component in each level denoted by  $\mathbf{M}_{L_{V_i}}^j$ .

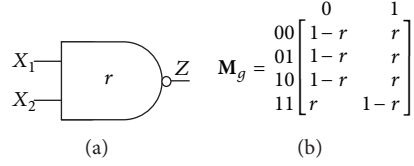


FIGURE 2: (a) A 2-input NAND gate and (b) its probability matrix  $\mathbf{M}_g$  (according to [10]).

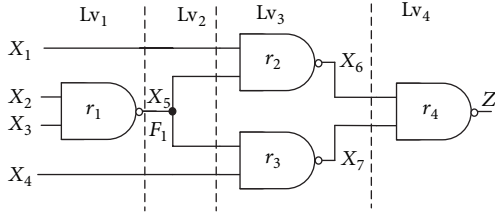


FIGURE 3: The example circuit schematic (a portion of C17 benchmark circuit).

*Step 2.* Within one level, the PTMs of each logic components (gates, wires, and fanout nodes) are tensored together to form the PTM of the current level; that is,  $\mathbf{M}_{Lv_i} = \mathbf{M}_{Lv_i}^1 \otimes \mathbf{M}_{Lv_i}^2 \cdots$ ;

*Step 3.* The PTMs of all levels are then multiplied together to get the circuit PTM; that is,  $\mathbf{M} = \prod_i \mathbf{M}_{Lv_i}$ .

*Step 4.* Calculate the ideal transfer matrix  $\mathbf{J}$  using the truth table of the logic function (error-free signal probabilities  $p(i)$  for input patterns are evaluated with the computation complexity of  $O(N_g \cdot 2^{N_{in}})$ ).

*Step 5.* The circuit reliability is given by [11]:

$$R_C = \sum_{i,j}^{J(i,j)=1} M(i, j) p(i). \quad (2)$$

We take a simple circuit as an example to illustrate the analysis process of PTM approach. The circuit schematic is shown in Figure 3, where the circuit has 4 levels, and the fanout  $F_1$  reconverges at gate number 4, generating the dependency between signal  $X_6$  and  $X_7$ . Since there are four inputs ( $X_1, X_2, X_3$ , and  $X_4$ ) and one single output  $Z$ , the circuit PTM would be a  $16 \times 2$  matrix  $\mathbf{M}$  which stores the probability of occurrence of all input-output vector pairs. The  $\mathbf{M}$  is constructed by combining PTMs of all levels (using matrix product due to serial connection in this case), while the PTM of each level is calculated by combining PTMs of each logic components within the current level (using tensor

product due to their parallel connection). More specifically, we have (based on [10])

$$\begin{aligned} \mathbf{M} &= (\mathbf{I} \otimes \text{NAND}_1 \otimes \mathbf{I})_{16 \times 8} \cdot (\mathbf{I} \otimes F \otimes \mathbf{I})_{8 \times 16} \\ &\quad \cdot (\text{NAND}_2 \otimes \text{NAND}_3)_{16 \times 4} \cdot (\text{NAND}_4)_{4 \times 2} \\ \mathbf{I} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (3)$$

$$\text{NAND}_i = \begin{bmatrix} 1-r_i & r_i \\ 1-r_i & r_i \\ 1-r_i & r_i \\ r_i & 1-r_i \end{bmatrix},$$

where the matrix  $\mathbf{I}$  refers to a  $2 \times 2$  identity PTM, and each parenthesized term in (3) corresponds to a specific circuit level. Assuming the gate reliabilities are  $r_1 = r_2 = r_3 = r_4 = 0.95$  and the probability of all input signals is equally 0.5, the circuit PTM and ideal transfer matrix are found using the above algorithm as follows:

$$\mathbf{M} = \begin{bmatrix} 0.8622 & 0.1378 \\ 0.1312 & 0.8688 \\ 0.8622 & 0.1378 \\ 0.1312 & 0.8688 \\ 0.8622 & 0.1378 \\ 0.1312 & 0.8688 \\ 0.8622 & 0.1378 \\ 0.8238 & 0.1762 \\ 0.1312 & 0.8688 \\ 0.0928 & 0.9073 \\ 0.1312 & 0.8688 \\ 0.0928 & 0.9073 \\ 0.1312 & 0.8688 \\ 0.0928 & 0.9073 \\ 0.8238 & 0.1762 \\ 0.8217 & 0.1783 \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}. \quad (4)$$

It can be seen from  $\mathbf{M}$  that the output reliability depends on input patterns. The lowest and highest values for the output reliability are 0.8217 and 0.9073, which occur when the input vector  $(X_1 X_2 X_3 X_4) = (1111)$  and  $(1001, 1011, \text{ and } 1101)$ , respectively. The circuit reliability is found to be  $R_C = 0.8658$  with the runtime of 0.2798 s.

The PTM algorithm has been implemented on some small circuits. The simulation results show that its performance is fairly good for circuits with less than 20 gates. If the circuit



size increases to  $\sim 40$ , both runtime and memory cost will grow dramatically, making the PTM method computationally expensive. In order to handle large-scale circuits, a variant PTM method was proposed in [11], where the input vector sampling is used. The simulation results show that this does improve efficiency with reduced memory cost, while the accuracy remains to be seen.

In summary, the PTM method has two major limitations. First, the signal width of the circuit that can be analyzed is very limited. This is due to the fact that its space complexity grows exponentially with the number of inputs and outputs, leading to prohibitively massive matrix storage and manipulation overhead for large-scale circuits. Secondly, the circuit structure needs to be preprocessed (such as circuit levelization and identification of the fanout nodes and wire pairs) prior to the algorithm implementation. Also, the PTM assumes all signals are correlated, which makes the method less efficient for circuits with no or a few reconvergent fanouts.

### 3. Monte Carlo (MC) Simulation

MC is a widely known simulation-based approach, where experimental data are collected to characterize the behavior of a circuit by randomly sampling its activity [2]. It is usually used when an analytical approach is unavailable or difficult to implement. The obvious drawbacks of this approach lie in the fact that numerous pseudorandom numbers need to be generated, and a large number of simulation runs must be executed to reach a stable result. This makes the reliability analysis for large circuits a very time-consuming process. As a stochastic computation framework, the MC method makes the result gradually converge to its exact value as more simulation runs are performed. In the process of achieving relatively stable results, certain statistical parameters (such as standard deviation  $\sigma$  and/or coefficient of variance (CV) which is defined as the ratio of the standard deviation and the mean, i.e.,  $\sigma/\mu$ ) are usually used as the stopping criteria. In [2],  $CV = 0.001$  is used to represent an acceptable level of accuracy, and the number of simulation runs required is given by

$$N_{MC} = \frac{1 - R_C}{R_C} \cdot \frac{1}{CV^2} \approx 10^6 \cdot \left( \frac{1}{R_C} - 1 \right), \quad (5)$$

where  $R_C$  is again the circuit reliability. Since the circuit reliability usually decreases with the circuit size ( $N_g$ ), the  $N_{MC}$  will increase with the circuit size for a given accuracy (measured by CV). Assuming that the  $R_C$  ranges from 0.1 to 0.9, the number of MC runs will vary around  $10^5 \sim 10^7$ . It should be mentioned that (5) only gives an approximated range of  $N_{MC}$ , and its actual value is usually determined experimentally for real circuits. Let us take the circuit of Figure 3 again as an example. From (5), the required  $N_{MC}$  is  $\sim 1.55 \times 10^5$  if  $R_C = 0.8658$ . Figure 4 shows the relative error at  $R_C$  against  $N_{MC}$ . It can be seen from the figure that after  $\sim 10^4$  runs, the result becomes relatively stable around its final value. However, a small random fluctuation is inevitable. Even after  $\sim 10^5$  simulation runs, the relative error of

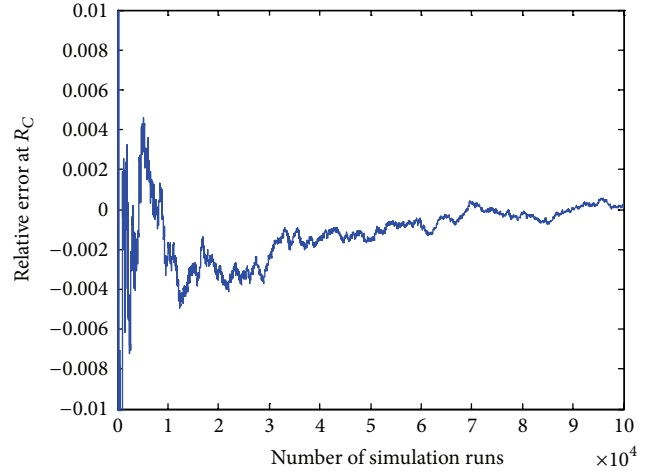


FIGURE 4: The relative error of circuit reliability  $R_C$  of Figure 3 versus the number of MC simulation runs  $N_{MC}$ .

the MC result is a nonzero value ( $5.636e - 04$ ), indicating a low convergent rate with the MC. This is a common feature for stochastic computations.

### 4. Stochastic Computation Model (SCM)

Unlike the MC method which uses Bernoulli sequences for simulation, the SCM approach takes non-Bernoulli sequences [2, 21]. In a non-Bernoulli sequence, for a given probability  $p$  and a sequence length  $N$ , the number of “1”s to be generated is fixed and given by  $N \cdot p$ , and only the positions of the “1”s are determined by a random permutation of binary bits. Therefore, in SCM approach, less pseudorandom numbers are generated for the same length of simulation, compared to MC simulation where pseudorandom numbers are independently generated for each gate or input to mimic the behavior of probabilistic circuits [2].

Consider a circuit with  $N_{in}$ ,  $N_{out}$ ,  $N_g$ ,  $P_{in}$ , and  $\varepsilon$  (refer to the previous sections for definitions of these variables). If we use a sequence length of  $N$ , the total required number of random numbers is given by  $(N_{in} + N_g) \cdot N$  in MC simulation. In contrast, for the SCM approach with the same sequence length, only  $N\varepsilon$  pseudorandom numbers need to be generated (for the positions of “1”s) for a gate with error rate  $\varepsilon$ . Therefore, the total number of random numbers is reduced to  $(N_{in} \cdot p_{in} + N_g \cdot \varepsilon) \cdot N$ . Since the gate error rate  $\varepsilon$  is usually a small value which can be viewed as a scale factor, the total required random number is significantly reduced. In other words, for a specific level of accuracy, the non-Bernoulli sequence requires a smaller sequence length than the Bernoulli sequence does. However, how to efficiently determine the required minimum sequence length for the SCM is still an open question. In [2], an empirical function (rather than an analytical expression) was used for this purpose.

Again, we took the example circuit of Figure 3 and used the same sequence length with MC (i.e.,  $N_{SCM} = N_{MC} = 10^5$ ) with gate error rate  $\varepsilon = 0.001$ . The SCM and MC

TABLE 1: Runtime comparison of MC and SCM on benchmark circuits ( $\varepsilon = 0.01$ ).

Circuit	Size	MC ( $10^6$ runs)	SCM ( $10^6$ runs)	
		Runtime (s)	Runtime (s)	
			$\varepsilon = 0.01$	$\varepsilon = 0.1$
c432	160	183	31	38
c499	202	203	37	45
c880	383	373	63	77
c1355	546	472	92	111
c1908	880	842	183	215
c2670	1193	1151	265	311
c3540	1669	1616	409	505
c5315	2406	2548	786	961
c7552	3512	3732	1325	1495

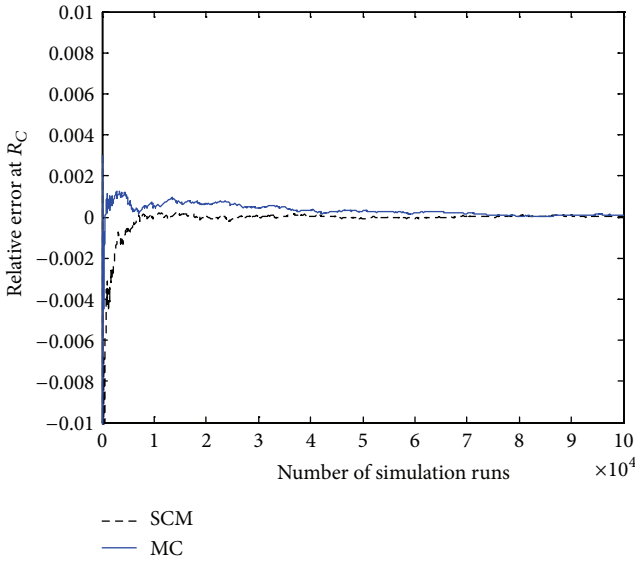


FIGURE 5: The relative error of circuit reliability  $R_C$  versus the number of simulation runs  $N_{MC}/N_{SCM}$ .

simulation results are compared in Figure 5, where both have a similar convergence rate. However, the runtimes with SCM and MC are  $T_{SCM} = 0.0745$  s and  $T_{MC} = 1.2528$  s, respectively, indicating that the SCM method is more efficient than the MC. This efficiency improvement is mainly due to less random numbers that are generated in the SCM simulation.

We also implemented both SCM and MC approaches in *Matlab* with the same sequence length of  $10^6$  (gate error rate  $\varepsilon = 0.01$ ) and tested their performance on ISCAS'85 benchmark circuits. The results are shown in Table 1, where the runtime with the SCM is around 1/6~1/3 of that with the MC. One of the disadvantages of SCM is the difficulty in determining its simulation sequence length  $N_{SCM}$ . Also, its runtime is proportional to gate error rate  $\varepsilon$  as well as input probabilities. If  $\varepsilon$  is relatively large (say 0.2), the runtime improvement of SCM over MC would be marginal (only scaled by a constant).

## 5. Probabilistic Gate Model (PGM)

The PGM is another reliability analysis method which is based on the probabilistic models of unreliable logic gates [22–25]. In the simple version of PGM, the input signals of each gate in the circuit are assumed to be independent. Under this assumption, the output probability of each gate can be easily calculated using the information of input signal probabilities and gate error rate. For instance, consider a 2-input NAND gate with input probabilities of  $X_1$  and  $X_2$  and gate error rate of  $\varepsilon$ . Its output signal probability can be expressed as (after [24])

$$\begin{aligned} Z &= \Pr("1" \mid \text{gate faulty}) \cdot \Pr(\text{gate faulty}) \\ &\quad + \Pr("1" \mid \text{gate not faulty}) \cdot \Pr(\text{gate not faulty}) \quad (6) \\ &= (1 - \varepsilon) + (2\varepsilon - 1) X_1 X_2. \end{aligned}$$

This output probability  $Z$  can be used recursively as the input information at next level of gates. One of the main features with PGM is that the circuit reliability is analyzed by exhaustively evaluating each input combination and output. For any given input combination, the error-free output value  $Z_{ef}$  is calculated, and then the output signal probability  $P_O$  is evaluated using the PGM of all gates in the circuit. Depending on the error-free output value, the output reliability  $R$  for this specific input combination is given by [24]

$$R = \begin{cases} P_O, & Z_{ef} = 1, \\ 1 - P_O, & Z_{ef} = 0. \end{cases} \quad (7)$$

Finally, the overall output reliability is the weighted sum of all conditional output reliabilities over all possible input combinations, where the weight is the probability of a specific input combination.

Intuitively, the operation process of PGM is similar to PTM in the sense that both of them consider all input combinations in a forward topological order. An obvious disadvantage with the PGM approach is that it is almost impossible to exhaustively enumerate all input combinations when the number of inputs increases (say to 30 and above). Therefore, a certain sampling technique is often necessary for large circuits. The input patterns sampling becomes another source of errors, in addition to the inaccuracy caused by signal independence assumption in constructing gate PGMs (it should be pointed out that while signal correlations due to fanouts originating from the primary inputs are eliminated by assigning the deterministic values (either "0" or "1") to all primary inputs, those caused by other reconvergent fanout nodes are not).

In order to eliminate all signal correlations, an accurate PGM algorithm was proposed in [24] where deterministic values are assigned explicitly to all reconvergent fanout nodes within the circuit. More specifically, for each fanout, the original circuit is transformed to two auxiliary circuits [24], one with the fanout node being set to logic value "0" and the other to "1." In each of these two circuits, the output probability is computed by using conditional probabilities for the given value at the fanout. This procedure is executed iteratively until

TABLE 2: Simulation results for simple PGM in comparison with MC.

Circuit	Size	Simple PGM approach ( $10^3$ samples)			Monte Carlo ( $10^6$ runs)
		Average error (%)	Max error (%)	Runtime (s)	Runtime (s)
C432	160	0.54	1.59	4.66	183
C499	202	0.1	0.31	4.92	203
C880	383	0.61	2.83	9.17	373
C1355	546	1.26	1.66	12.21	472
C1908	880	0.39	0.85	18.69	842
C2670	1193	2.43	16.61	25.72	1151
C3540	1669	0.077	2.27	39.46	1616
C5315	2307	10.88	43.16	61.42	2548
C7552	3512	2.68	13.68	75.19	3732
Average	—	<b>2.18</b>	<b>9.22</b>	—	—

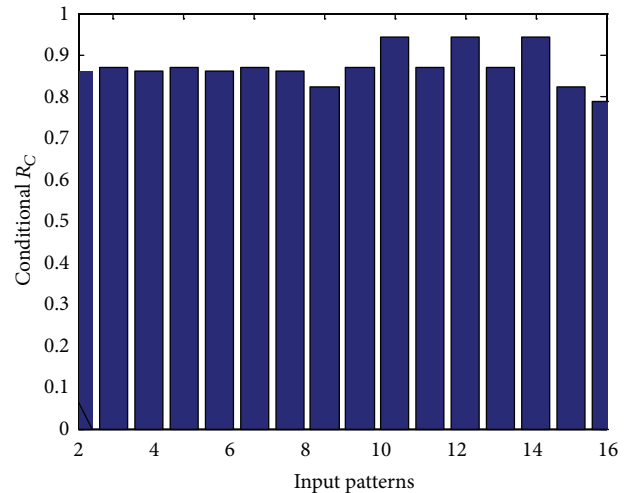
TABLE 3: Comparison of simple PGM and accurate PGM.

Circuit	Size	Simple PGM approach ( $10^3$ samples)		Accurate PGM approach ( $10^3$ samples) [24]
		Average error (%)	Runtime (s)	Runtime (s)
Cu	43	1.37	0.0277	0.10
z4ml	45	0.94	0.0039	0.05
x2	38	0.52	0.0275	0.22
Mux	50	0.52	0.282	0.10

all fanouts have been processed. If all input combinations are simulated, this procedure will lead to exact results for any circuits. However, for a circuit with  $N_f$  reconvergent fanouts, a total of  $2^{N_f}$  auxiliary circuits are required and analyzed. Therefore, the computation complexity becomes  $O(N_g \cdot 2^{N_{in}+N_f})$  [24]. However, in many real circuits, the number of reconvergent fanouts  $N_f$  is comparable to the number of gates ( $N_g$ ). Thus, the complexity of the above accurate PGM algorithm is still an exponential function of the circuit size, making it infeasible in general for large circuits.

In an effort to improve the efficiency of the accurate PGM method, a modular PGM approach was also introduced in [24]. It is based on the observation that many large circuits contain a limited number of simple logic components that are used repeatedly. With this in mind, circuits can be decomposed into several modules whose reliabilities are calculated using the accurate PGM method. The circuit output reliability is then evaluated by combining these modules along the path from primary inputs. Unfortunately, the input sampling is still needed in this case for large-scale circuits.

For the example circuit of Figure 3 with 4 input signals, a total of 16 input combinations need to be considered. We plot the conditional output reliability for each input combination in Figure 6, which shows that the output reliability varies within a relatively small range (no more than  $\pm 10\%$ ) for different input combinations. In other words, the input vector sampling can be implemented effectively with small errors. The overall output reliability is given by a weighted sum over all input combinations and is found to be  $R_C = 0.8701$  (with the runtime of  $T_{PGM} = 0.0093$  s), compared to the accurate value of 0.8658 given by PTM (i.e., the relative error is as low as  $\sim 0.5\%$ ).

FIGURE 6: The conditional output reliability of Figure 3 for different input combinations ( $x$ -axis labels 1~16 indicate 16 input patterns from 0000~1111).

In order to see the performance of different PGM algorithms on large circuits, we implemented the simple PGM algorithm in *Matlab* and tested it on ISCAS'85 benchmarks. The results are shown in Table 2. We also compare the simple PGM with both accurate and modular PGM methods in Tables 3 and 4, where the simulation results for both accurate and modular PGM methods are taken from [24].

It can be seen from these tables that the simple PGM algorithm can provide highly accurate results if the circuits (such as C432 and C1355) have no or few reconvergent fanouts and/or if the fanouts originate from the primary inputs.

TABLE 4: Simulation results for modular PGM in comparison with MC.

Circuit	Size	Modular PGM approach ( $10^3$ samples) [24]		Monte-Carlo ( $10^6$ runs)
		Average error (%)	Runtime (s)	Runtime (s)
C432	160	9.21	0.25	183
C499	202	0.11	2.15	203
C1355	546	4.2	2.98	472
C2670	1193	0.43	4.26	1151

For those circuits with significant fanouts (such as C2670 and C5315), the average (or maximum) errors for the simple PGM can increase significantly (in particular, the maximum error is up to 43% for C5315, as shown in Table 2). From Table 3, the accurate PGM need longer runtimes than the simple PGM for small circuits. Results in Table 4 confirm that the modular PGM is very efficient while the accuracy may not always be good enough for some circuits (with an average error of 9% for C432).

In summary, for all the above three different versions of PGM, the input sampling is inevitable for improved efficiency if the number of primary inputs  $N_{in}$  is large ( $\sim 30$ ). This is mainly where the analysis errors come in. Thus, it can be concluded that they represent a good model only for circuits with a small number of primary inputs, where no input sampling is required. For the circuits without reconvergent fanouts, the input sampling in the PGM approach is unnecessary, because both signal probability and output reliability in this case can be computed within  $O(N_g)$  time (see [29] for details).

## 6. Observability-Based Reliability Analysis

Another reliability analysis method was presented in [26], which is based on the observation that an error at the output of any gate is the cumulative effect of a local error component attributed to the error probability of the gate, and a propagated error component was attributed to the failure of gates in its transitive fan-in cone. In [26], the observability of a gate (or its output signal) is the conditional circuit error probability given the single error at current gate. The value of this observability can be simply defined as  $o_i = (1 - R_C(\varepsilon_i = 1))$ , where  $R_C(\varepsilon_i = 1)$  is the circuit reliability given a single error with the current gate, and can be calculated using Boolean differences [29], symbolic techniques (such as BDDs), or simulation method. It can be expected that the gate observabilities are highly related to the input probabilities.

For a single-fault case (i.e., only one gate in the circuit is erroneous), the circuit reliability (assuming a single primary output) can be simply calculated by considering each fault case individually. Assume that the error rate and observability of the  $i$ th gate are  $\varepsilon_i$  and  $o_i$ , respectively. If gate  $i$  is erroneous while the other gates are fault-free, the output reliability simply is equal to  $o_i$ . Thus, the overall reliability can be easily calculated by

$$R_C = \sum_{\text{all } i} \left( \varepsilon_i \cdot o_i \cdot \prod_{j \neq i} (1 - \varepsilon_j) \right), \quad (8)$$

which is exact for the single-fault case.

If a multiple-error case is considered, the complexity of computing the reliability will grow exponentially with  $N_g$ . In order to improve the efficiency in this case, the following two assumptions are used in [26]: (a) the impacts of gate failures on the primary output are decoupled, which implies that the output is erroneous if an odd number of gates are simultaneously observable and (b) the observabilities of all gates are independent. As a result, the simultaneous observability of multiple gates is simply the product of their individual observabilities.

We took the example circuit of Figure 3 for illustration. First, let us assume all four gates ( $G_1 \sim G_4$ ) in the circuit are erroneous with the probabilities of  $\varepsilon_1$ ,  $\varepsilon_2$ ,  $\varepsilon_3$ , and  $\varepsilon_4$ , respectively (other cases can be analyzed similarly). Based on the above assumption (a), we only need to consider the cases where an odd number (1 or 3) of gates is simultaneously observable. This means that when an even number (0, 2, or 4) of gates is observable, the output signal  $Z$  will have correct value as gate errors are logically masked by one another. Secondly, under the assumption (b), the probability of only one gate being observable is given by  $\sum_i (o_i \cdot \prod_{j \neq i} (1 - o_j))$  (the probability of three gates being simultaneously observable can be calculated similarly). Based on these assumptions, a closed-form expression for the circuit reliability of the circuit (assuming a single primary output) can be written generally as a function of error probabilities and observabilities of all gates [26]; that is,

$$R_C = \frac{1}{2} \left( 1 + \prod_i (1 - 2\varepsilon_i o_i) \right), \quad (9)$$

which can be computed efficiently if all gate observabilities are known (however, this analysis is only suitable for small circuits or large ones with small values of gate error probabilities, which will be clear later). The gate observability can be determined using the PTM method. For instance, the observability of gate  $G_1$  in Figure 3 is calculated as the output reliability by setting  $r_1 = 0$  and  $r_2 = r_3 = r_4 = 1$ . The results are  $[o_1, o_2, o_3, o_4] = [0.25, 0.375, 0.375, 0]$ . We calculate the circuit reliability  $R_C$  using the above expression and plot the results against the accurate values given by the PTM in Figure 7(a) for different values of gate reliability. The relative error is shown in Figure 7(b). It can be seen clearly from these figures that the observability-based analysis is only accurate for small gate error rates, in which case the probability for single gate failure is significantly higher than that for multiple gate failures.

To reduce the computational complexity of the above observability-based reliability analysis, [26] also proposed



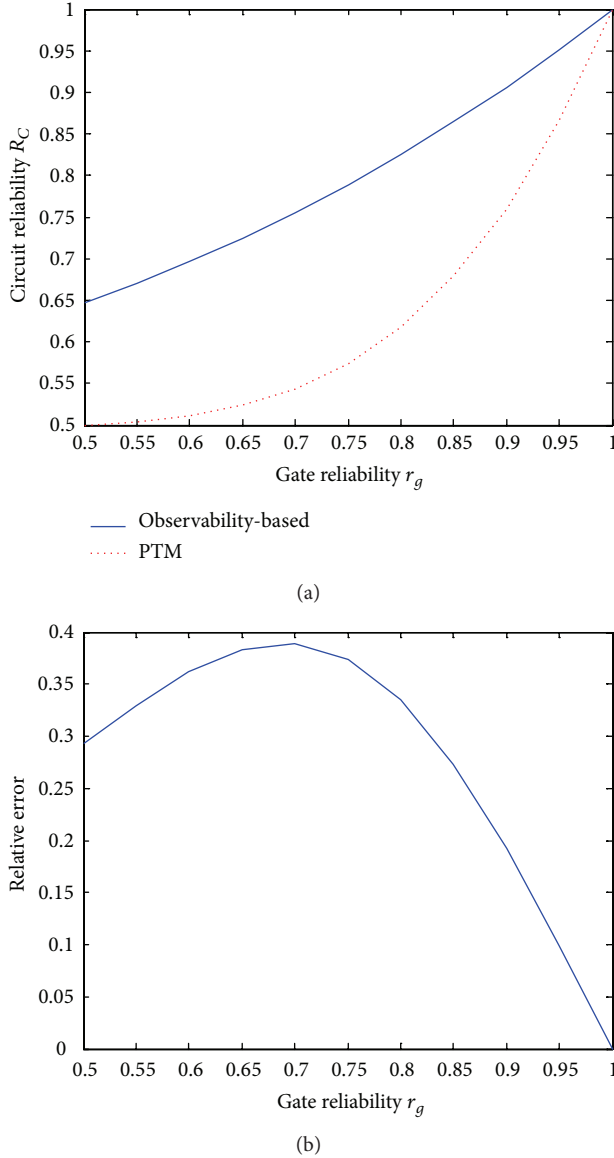


FIGURE 7: (a) Circuit reliability  $R_C$  versus gate reliability and (b) relative error versus gate reliability for the example circuit of Figure 3.

a sampling algorithm by considering the constraint that only a maximum of  $k$  gates can fail simultaneously. This algorithm first generates a set of samples for failed gates and guarantees that the total number of gates with error is no more than  $k$ . Then, a single-pass reliability analysis algorithm [26] was used to evaluate the error probability at the primary outputs, leading to the computational complexity of  $O(N_g \cdot k^2)$ , where  $N_g$  is the number of gates with error. For a specific sample, the reliabilities of gates in the sampling are set to be 0 and the rest are set to be 1. Finally, the overall circuit reliability is estimated by averaging the reliabilities over all samples. Therefore, this maximum- $k$  gate failure model can be viewed as a hybrid method that makes a trade-off between the accuracy of simulation-based method and the efficiency of analytical approach. It provides more accurate results than

the single-pass algorithm [26] and takes the shorter runtime than MC or SCM.

## 7. Correlation Coefficient Method (CCM)

CCM is a widely used approach that evaluates the signal probabilities for (fault-free) combination circuits [30]. As mentioned before, the reliability analysis can be transformed to signal probability computation. Therefore, the CCM can be used to evaluate the reliability estimation [8, 26–28, 30]. The main idea of CCM is briefly described below.

In order to compute the signal probability, the correlation coefficient between two probabilistic signals (denoted by  $i$  and  $j$ ) is defined as [8]

$$C_{i,j} = C_{j,i} = \frac{P(ij)}{P(i)P(j)} = \frac{P(j|i)}{P(j)}, \quad (10)$$

which is equal to 1 for signals  $i$  and  $j$  are independent. It should be noted that, here, only the first order correlation coefficients are considered, and the correlation of two signals with a third one (denoted by  $h$ ) is approximated as  $C_{ij,h} = C_{i,h} \cdot C_{j,h}$ . For reliability computation, four correlation coefficients for a pair of signals are needed. Each coefficient corresponds to a combination of events (i.e.,  $0 \rightarrow 1$  or  $1 \rightarrow 0$  error) on the signal pair. In other words, the signal error (or reliability) correlation coefficient between signals  $i$  and  $j$  is defined as [26]

$$\begin{aligned} C_{ij} &= \frac{P(i_{0 \rightarrow 1} j_{0 \rightarrow 1})}{P(i_{0 \rightarrow 1}) P(j_{0 \rightarrow 1})}, \\ C_{i\bar{j}} &= \frac{P(i_{0 \rightarrow 1} j_{1 \rightarrow 0})}{P(i_{0 \rightarrow 1}) P(j_{1 \rightarrow 0})}, \\ C_{\bar{i}j} &= \frac{P(i_{1 \rightarrow 0} j_{0 \rightarrow 1})}{P(i_{1 \rightarrow 0}) P(j_{0 \rightarrow 1})}, \\ C_{\bar{i}\bar{j}} &= \frac{P(i_{1 \rightarrow 0} j_{1 \rightarrow 0})}{P(i_{1 \rightarrow 0}) P(j_{1 \rightarrow 0})}, \end{aligned} \quad (11)$$

where the  $P(i_{0 \rightarrow 1})$  is the probability that the value of signal  $i$  flips to 1 from its correct value 0, that is, the error probability of  $i$  given that its error-free value is 0. Once the error correlations and error-free signal probabilities are generated, the single-pass analysis is conducted using the forward topological order with the computational complexity of  $O(N_g)$ . Since the computation complexity of CCM is linear with the number of levels ( $L$ ) and pseudoquadratic with the number of gates per level ( $N_L$ ), the overall complexity of CCM-based reliability analysis turns out to be  $O(N_g^{1.5})$  if a square circuit is assumed (i.e.,  $N_L = L = N_g^{0.5}$ ). This complexity is an upper bound as not all signals are correlated in real circuits.

In [26] which uses the CCM, an average relative error of up to ~13% over all outputs was reported for circuits with significant fanout (e.g., C499 and C1355) when the gate error rates range within  $[0, 0.5]$  (for other benchmark circuits,

the error was around 2~6%). Also, the relative errors may not be mitigated significantly by using more correlation coefficients. For instance, by using 0, 4, and 16 correlation coefficients, the relative errors for C499 are only improved to 13.1%, 11.2%, and 11.1%, respectively [26], where the zero-coefficient case means that all signals are treated as independent with the computation complexity of  $O(N_g)$ . It is shown in [26] that the runtime of using 4 coefficients is several orders of magnitude longer than the zero-coefficient case (~100 s versus ~1 s, for circuit with ~1000 gates). Therefore, it may not be worthwhile to calculate more correlation coefficients for slightly improved accuracy. In [30], the relative error for large circuits (with hundreds of gates) was reported at ~7% on average with the runtime of ~10 s, which is comparable to those from [26].

## 8. Comparison and Future Work

In summary, the ultimate goal of existing approaches for reliability analysis is to achieve more accurate results with as low computational cost as possible. Both accuracy and efficiency depend on specific circuit structures and their size, and, in most cases, the tradeoff between them needs to be made. The main features of each approach are described as follows.

- (a) If circuits have no reconvergent fanouts (e.g., a circuit with tree structure), both signal probability and reliability can be calculated exactly with linear time (i.e.,  $O(N_g)$ ). The readers are referred to [29] for further details.
- (b) For those circuits with reconvergent fanouts, the PTM method and accurate PGM model can promise exact results, while their computation costs are exponentially high. The PTM approach requires the space complexity of  $O(2^{N_{in}+N_{out}})$ , and the accurate PGM has the computation complexity of  $O(N_g \cdot 2^{N_{in}+N_f})$ . Thus, some sampling techniques are usually needed to handle large-scale circuits in these computation frameworks, leading to less accurate results.
- (c) Simulation-based methods (such as MC or SCM) can provide the results with high level of accuracy, as long as enough simulation sequences are applied. To achieve a required level of accuracy, the number of simulation runs need to be determined statistically or empirically. The time complexity can be estimated by  $O(N_g \cdot N_{MC})$  or  $O(N_g \cdot N_{SCM})$ , where  $N_g$  is again the circuit size and  $N_{MC}$  (or  $N_{SCM}$ ) represents the number of simulation runs. The SCM is more efficient than MC especially for small gate error rates, as the runtime of the former is approximately scaled by a constant factor.
- (d) The observability-based approach has some theoretical implications, since it gives reasonable results only for circuits with extremely-low gate error rates. The maximum- $k$  method can be viewed as the combination of CCM-based and simulation-based methods.

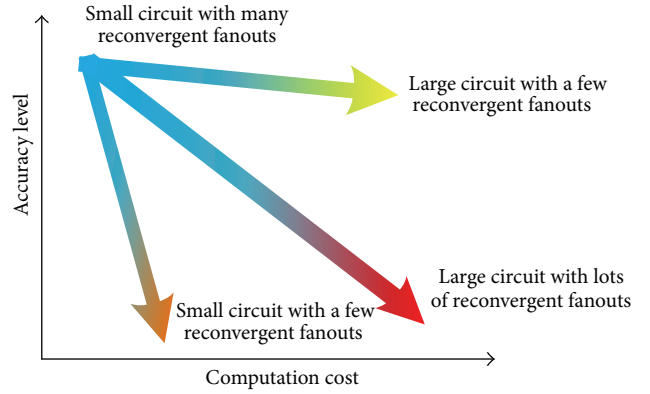


FIGURE 8: A general sketch of solution space for different circuit categories.

It shows better performance than the observability-based approach in terms of both accuracy and efficiency for lower gate error rates.

- (e) If all reconvergent fanouts within circuits originate from primary inputs, the simple PGM method gives exact results with the computational complexity of  $O(N_g \cdot 2^{N_{in}})$ . For circuits consisting of a few logic modules that are repetitively used, the modular PGM method is a good option that can provide good accuracy with short runtime.

From the above discussions, it can be concluded that errors in reliability analysis are mainly due to the reconvergent fanouts (or signal correlation) inherent in many circuits under consideration in the sense that the accurate results can be obtained efficiently for circuits with no or a few reconvergent fanouts. On the other hand, the circuit size (i.e., a large number of primary inputs or a large number of gates or both) is the main contributor to high computational costs for reliability analysis. Therefore, the most challenging problem is to analyze the reliability for large-scale circuits with a lot of reconvergent fanouts. Figure 8 illustrates the expected solution space in general in terms of accuracy and computational cost for different circuit categories. Any existing approach for the reliability analysis corresponds to a specific point in this space, which represents a tradeoff between accuracy and efficiency. For instance, the results from PTM, PGM, or MC fall into the right-upper corner of this figure with expensive computation and high level of accuracy. An ideal approach should be able to provide results somewhere near the left-upper corner where both accuracy and efficiency can be ensured.

While gate-level reliability analysis methods are well documented, there are some other important issues that remain to be tackled. First of all, most existing methods only deal with the reliability of each individual output and/or the averaged reliability over all outputs. However, the joint reliability for multiple outputs (i.e., the probability that all outputs are error-free simultaneously) is what really matters. This joint reliability could be totally different from any individual output reliability or the averaged output reliability,

depending on the possible correlation among individual output reliabilities. For an extreme case where all individual output reliabilities are independent, the joint reliability will simply be the product of all these reliabilities, which leads to a minimum value. As the correlation of output reliabilities becomes strong, the joint reliability tends to rise. In general, the complexity of computing the joint reliability would be an exponential function of the number of primary outputs. It is still an open question how to estimate the joint reliability for multiple-output circuits in an efficient way. Secondly, most of the current reliability analysis frameworks assume that the reliability for an error-free output being “0” (denoted by  $r_0$ ) is the same as that for an error-free output being “1” (denoted by  $r_1$ ). This is the so-called symmetric reliability model. However, this assumption does not always hold true in the real world. Thus, an asymmetric reliability model (where  $r_0 \neq r_1$ ) would make more sense for better estimation of reliability. This requires further research work that can take the asymmetric model into consideration. Finally, there is also plenty of room for gate-level reliability improvement using reliability-critical gates as well as considering other performance metrics (such as circuit area and delay and power consumption). Unfortunately, to the best of authors’ knowledge, little or limited study has been done so far in this regard.

## 9. Conclusion

We have reviewed the state-of-the-art methods for reliability analysis and shown their advantages and disadvantages. Some of these methods have been implemented on benchmark circuit examples to compare their performance in terms of accuracy and efficiency. While these methods seem to be effective for some specific cases/circuits, no single one of them stands out as an all-time winner due to the nature and complexity of the reliability analysis problem. Further work has also been suggested for the future research in this area.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] S. Borkar, “Designing reliable systems from unreliable components: The challenges of transistor variability and degradation,” *IEEE Micro*, vol. 25, no. 6, pp. 10–16, 2005.
- [2] J. Han, H. Chen, J. Liang, P. Zhu, Z. Yang, and F. Lombardi, “A stochastic computational approach for accurate and efficient reliability evaluation,” *IEEE Transactions on Computers*, vol. 63, no. 6, pp. 1336–1350, 2014.
- [3] S. Krishnaswamy, S. M. Plaza, I. L. Markov, and J. P. Hayes, “Signature-based SER analysis and design of logic circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 1, pp. 74–86, 2009.
- [4] C. Chen and Y. Mao, “A statistical reliability model for single-electron threshold logic,” *IEEE Transactions on Electron Devices*, vol. 55, no. 6, pp. 1547–1553, 2008.
- [5] C. Chen, “Reliability-driven gate replication for nanometer-scale digital logic,” *IEEE Transactions on Nanotechnology*, vol. 6, no. 3, pp. 303–308, 2007.
- [6] J. von Neumann, “Probabilistic logics and the synthesis of reliable organisms from unreliable components,” in *Automata Studies*, C. E. Shannon and J. McCarthy, Eds., pp. 43–98, Princeton University Press, Princeton, NJ, USA, 1956.
- [7] J. Han and P. Jonker, “A system architecture solution for unreliable nanoelectronic devices,” *IEEE Transactions on Nanotechnology*, vol. 1, no. 4, pp. 201–208, 2002.
- [8] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, “Estimate of signal probability in combinational logic networks,” in *Proceedings of the 1st European Test Conference*, pp. 132–138, Paris, France, April 1989.
- [9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, Calif, USA, 1979.
- [10] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, “Accurate reliability evaluation and enhancement via probabilistic transfer matrices,” in *Proceedings of the Design, Automation and Test in Europe*, vol. 1, pp. 282–287, March 2005.
- [11] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, “Probabilistic transfer matrices in symbolic reliability analysis of logic circuits,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 13, no. 1, article 8, 2008.
- [12] W. Ibrahim, V. Beiu, and M. H. Sulieman, “On the reliability of majority gates full adders,” *IEEE Transactions on Nanotechnology*, vol. 7, no. 1, pp. 56–67, 2008.
- [13] T. Rejimon, K. Lingasubramanian, and S. Bhanja, “Probabilistic error modeling for nano-domain logic circuits,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 1, pp. 55–65, 2009.
- [14] T. Rejimon and S. Bhanja, “Scalable probabilistic computing models using Bayesian networks,” in *Proceedings of the IEEE International 48th Midwest Symposium on Circuits and Systems (MWSCAS ’05)*, pp. 712–715, August 2005.
- [15] J. T. Flaquer, J. M. Daveau, L. Naviner, and P. Roche, “Fast reliability analysis of combinatorial logic circuits using conditional probabilities,” *Microelectronics Reliability*, vol. 50, no. 9–11, pp. 1215–1218, 2010.
- [16] R. I. Bahar, J. Chen, and J. Mundy, “A probabilistic-based design for nanoscale computation,” in *Nano, Quantum and Molecular Computing: Implications to High Level Design and Validation*, S. Shukla and R. I. Bahar, Eds., chapter 5, Kluwer Academic, Norwell, Mass, USA, 2004.
- [17] R. I. Bahar, J. Mundy, and J. Chen, “A probability-based design methodology for nanoscale computation,” in *Proceedings of the International Conference on Computer-Aided Design*, pp. 480–486, November 2003.
- [18] A. R. Kermany, N. H. Hamid, and Z. A. Burhanudin, “A study of MRF-based circuit implementation,” in *Proceedings of the International Conference on Electronic Design (ICED ’08)*, pp. 1–4, December 2008.
- [19] D. Bhaduri and S. Shukla, “NANOLAB—a tool for evaluating reliability of defect-tolerant nanoarchitectures,” *IEEE Transactions on Nanotechnology*, vol. 4, no. 4, pp. 381–394, 2005.
- [20] X. Lu, J. Li, and W. Zhang, “On the probabilistic characterization of nano-based circuits,” *IEEE Transactions on Nanotechnology*, vol. 8, no. 2, pp. 258–259, 2009.
- [21] H. Chen and J. Han, “Stochastic computational models for accurate reliability evaluation of logic circuits,” in *Proceedings*

of the 20th Great Lakes Symposium on VLSI (GLSVLSI '10), pp. 61–66, May 2010.

- [22] J. B. Gao, Y. Qi, and J. A. B. Fortes, “Bifurcations and fundamental error bounds for fault-tolerant computations,” *IEEE Transactions on Nanotechnology*, vol. 4, no. 4, pp. 395–402, 2005.
- [23] J. Han, E. Taylor, J. Gao, and J. Fortes, “Faults, error bounds and reliability of nanoelectronic circuits,” in *Proceedings of the IEEE 16th International Conference on Application-Specific Systems, Architectures, and Processors (ASAP '05)*, pp. 247–253, July 2005.
- [24] J. Han, H. Chen, E. Boykin, and J. Fortes, “Reliability evaluation of logic circuits using probabilistic gate models,” *Microelectronics Reliability*, vol. 51, no. 2, pp. 468–476, 2011.
- [25] J. Han, E. R. Boykin, H. Chen, J. H. Liang, and J. A. B. Fortes, “On the reliability of computational structures using majority logic,” *IEEE Transactions on Nanotechnology*, vol. 10, no. 5, pp. 1009–1022, 2011.
- [26] M. R. Choudhury and K. Mohanram, “Reliability analysis of logic circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 3, pp. 392–405, 2009.
- [27] L. Chen and M. B. Tahoori, “An efficient probability framework for error propagation and correlation estimation,” in *Proceedings of the IEEE 18th International On-Line Testing Symposium (IOLTS '12)*, pp. 170–175, Sitges, Spain, June 2012.
- [28] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, “Testability measures in pseudorandom testing,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 6, pp. 794–800, 1992.
- [29] N. Mohyuddin, E. Pakbaznia, and M. Pedram, “Probabilistic error propagation in logic circuits using the boolean difference calculus,” in *Proceedings of the 26th IEEE International Conference on Computer Design (ICCD '08)*, pp. 7–13, October 2008.
- [30] S. Sivaswamy, K. Bazargan, and M. Riedel, “Estimation and optimization of reliability of noisy digital circuits,” in *Proceedings of the 10th International Symposium on Quality Electronic Design (ISQED '09)*, pp. 213–219, March 2009.



