Accurate and scalable reliability analysis of logic circuits

Mihir R. Choudhury and Kartik Mohanram
Department of Electrical and Computer Engineering
Rice University, Houston, TX 77005
{mihir,kmram}@rice.edu

Abstract

Reliability of logic circuits is emerging as an important concern that may limit the benefits of continued scaling of process technology and the emergence of future technology alternatives. Reliability analysis of logic circuits is $\mathcal{NP}$-hard because of the exponential number of inputs, combinations and correlations in gate failures, and their propagation and interaction at multiple primary outputs. By coupling probability theory with concepts from testing and logic synthesis, this paper presents accurate and scalable algorithms for reliability analysis of logic circuits. Simulation results for several benchmark circuits demonstrate the accuracy, performance, and potential applications of the proposed analysis technique.

1. Introduction

It is widely acknowledged that there will be a sharp increase in manufacturing defect levels and transient fault rates in future electronic technologies, e.g., [1, 2]. Defects and faults impact performance and limit the reliability of electronic systems. This has led to considerable interest in practical techniques for reliability analysis that are accurate, robust, and scalable with design complexity. Reliability analysis of logic circuits refers to the problem of evaluating the effects of errors due to noise at individual transistors, gates, or logic blocks on the outputs of the circuit. The models for noise range from highly specific decomposition of the sources, e.g., single-event upsets, to highly abstract models that combine the effects of different failure mechanisms. Reliability analysis is $\mathcal{NP}$-hard because of the exponential number of inputs, combinations and correlations in gate failures, and their propagation and interactions at multiple primary outputs.

Standard techniques for reliability analysis use fault injection and simulation in a Monte Carlo framework. Although parallelizable and scalable, they are still not efficient for use on large circuits. Analytical methods for reliability analysis are applicable to very simple structures such as 2-input and 3-input gates, and regular fabrics [3,4]. Although they can be applied to large multi-level circuits with simplifying assumptions and compositional rules, there is a significant loss in accuracy. Recent advances in reliability analysis are based on probabilistic transfer matrices (PTMs) [5] and Bayesian networks [6]. However, both approaches require significant runtimes for small benchmark circuits. This can be attributed to the storage and manipulation overhead of large algebraic decision diagrams (ADDs) that support PTM operations, and large conditional probability tables that support Bayesian networks.

To the best of our knowledge, this is the first work that describes fast, accurate, and scalable algorithms for reliability analysis of logic circuits. The first algorithm described in this paper uses observability metrics to quantify the impact of a gate failure on the output of the circuit. The observability-based approach provides a closed-form expression for circuit reliability as a function of the failure probabilities and observabilities of the gates. The closed-form expression is accurate when the probability of a single gate failure is significantly higher than the probability of multiple gate failures, and has application to soft error rate estimation.

The observability-based approach provides useful insight into the effects of multiple gate failures that is leveraged to develop a single-pass algorithm for reliability analysis. Gates are topologically sorted and processed in a single pass from the inputs to the outputs. Topological sorting ensures that before a gate is processed, the effects of multiple gate failures in the transitive fanin cone of the gate are computed and stored at the inputs of the gate. Using the joint signal probability distribution of the gate's inputs, the propagated error probabilities from its transitive fanin stored at its inputs, and the error probability of the gate, the cumulative effect of failures at the output of the gate are computed. The effects of reconvergent fanout on error probabilities is addressed using correlation co-efficients. Simulation results for several benchmark circuits illustrate the accuracy, efficiency, and scalability of the proposed technique.

This paper is organized as follows. Section 2 provides a background in reliability analysis. Section 3 describes an observability-based algorithm for reliability analysis. Section 4 describes a single-pass algorithm for reliability analysis. Section 5 discusses simulation results and potential applications. Section 6 is a conclusion.

2. Background

The classical model for errors due to noise in a logic circuit was introduced by von Neumann in 1956 [3]. Noise at a gate is modeled as a binary symmetric channel (BSC), with a cross-over probability ϵ. In other words, following the computation at the gate, the BSC can cause the gate output to flip symmetrically (from $0 \rightarrow 1$ or $1 \rightarrow 0$) with the same probability of error, ϵ. Each gate has an $\epsilon \in [0, 0.5]$ associated with it, where ϵ equals 0 for an error-free gate and ϵ equals 0.5 for perfectly noisy gate (a gate with random output). When $\epsilon > 0.5$, it is equivalent to a gate with $\epsilon < 0.5$, but with the complement Boolean function. The BSC model allows the effects of different sources of noise such as crosstalk, terrestrial cosmic radiation, electromagnetic interference, etc. to be combined into the failure probability ϵ. Note that gates are assumed to fail independently of each other. Although this may not be a realistic assumption, it helps to simplify reliability analysis while still providing valuable insights into circuit reliability.

Reliability of a logic circuit is defined as the probability of error at the output of the logic circuit, δ, as a function of failure proba-

This research was supported in part by a gift from Advanced Micro Devices.
.

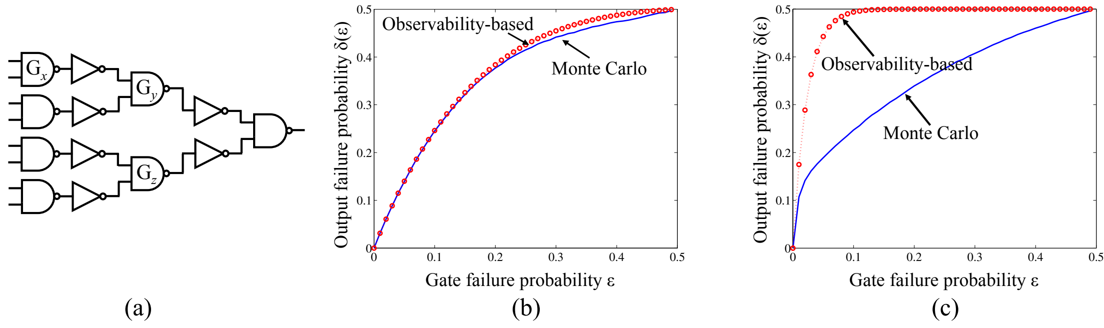

Figure 1: Circuit for illustrating the effect of noise and correlation on observability

bilities $\vec{\epsilon}$ of the gates, where $\vec{\epsilon}$ is the vector containing the failure probabilities $\{\epsilon_1, \epsilon_2, ...\}$ of the gates in the circuit. Reliability $\delta(\vec{\epsilon})$ can lie in the interval from 0 to 1. Reliability analysis is $\mathcal{NP}$-hard because of the exponential number of inputs, combinations and correlations in gate failures, and their propagation and interactions at multiple primary outputs.

The traditional approach to reliability analysis uses fault injection and simulation in a Monte Carlo framework. Recent progress in reliability analysis has seen the use of probabilistic transfer matrices (PTMs) [5] and Bayesian networks [6]. Without exception, these approaches suffer from the problem of scalability. Monte Carlo simulations have the added disadvantage of inflexibility, since the entire simulation has to be repeated for any change in circuit structure or $\vec{\epsilon}$. PTM-based reliability analysis uses transfer matrices to represent input-output behavior of noisy circuits. PTMs store the probability of occurrence of every input-output vector pair for each level in the circuit to compute the probability of error at the output of the circuit. This leads to massive matrix storage and manipulation overhead. Even with compaction of the matrices using ADDs, the high runtimes for benchmark circuits with 20–50 gates suggest their inapplicability to large circuits. Although this problem is somewhat mitigated in the Bayesian network approach for small circuits, manipulating Bayesian networks for large circuits is potentially intractable. Alternatively, analytical approaches developed to study fault-tolerant approaches like nand multiplexing and majority voting can be used for reliability analysis [3,4]. However, the simple compositional rules that these approaches use work best on regular structures. When used on irregular multi-level structures such as logic circuits, they suffer significant penalties in accuracy even on small circuits. By coupling probability theory with concepts from testing and logic synthesis, this paper presents accurate and scalable algorithms for reliability analysis of logic circuits.

3. Observability-based reliability analysis

In this section, an intuitive approach to reliability analysis is described. It is based upon the observation that a failure at a gate close to the primary output has a greater probability of propagating to the primary output than a gate several levels of logic away from the primary outputs. This is because a failure that has to propagate through several levels of logic has a higher probability of being logically masked. This can be quantified by applying the concept of observability, which has historically found use in the testing and logic synthesis domains [7].

For reliability analysis, the observability of any wire in the circuit can be defined as the probability that a $0 \rightarrow 1$ or $1 \rightarrow 0$ error at that wire affects the output of the circuit. Consider a combinational circuit with m inputs $x_1, x_2, \cdots, x_m$ and n gates. Without loss of generality, we assume that the circuit has a single output y. Denote the error probability (observability) of the ith gate by $\epsilon_i(o_i)$. Note that the o_is are the noiseless observabilities, i.e., all the gates are assumed noise-free when the o_is are calculated. Observabilities can be calculated using Boolean differences, symbolic techniques based on binary decision diagrams (BDDs), or simulation. Our implementation uses BDDs to compute observabilities. Using the observabilities o_i, a closed-form expression for the reliability $\delta_y(\vec{\epsilon})$ can be derived as follows.

Let Ω be the set of all the gates in the circuit. Consider a set $G \subseteq \Omega$ of gates that have failed. The output y will be in error when an *odd number* of gates in G are observable. This is because if an even number of gates in G are observable, the errors of these failed gates will mask each other at y. Given G, the probability that y is in error is given by

$$\Pr(y_{\text{error}}|G) = \frac{1}{2}\left(\prod_{j\in G}(o_j + (1-o_j)) - \prod_{j\in G}((1-o_j) - o_j)\right)$$
$$= 1/2\left(1 - \prod\nolimits_{j\in G}(1-2o_j)\right) \quad (1)$$

Here, $\left(\prod_{j\in G}(o_j + (1-o_j)) - \prod_{j\in G}((1-o_j)-o_j)\right)$ is two times the probability that an odd number of gates in G are observable at y (if the product terms are expanded, the even terms cancel). When this is generalized by considering all sub-sets of Ω of size k, S_k, the probability that y is in error is given by

$$\Pr(y_{\text{error}}|S_k) = \sum_{G\in S_k}\left(\prod_{i\in G}\epsilon_i \prod_{j\in G^c}(1-\epsilon_j)\right)\Pr(y_{\text{error}}|G) \quad (2)$$

Here, $\prod_{i\in G}\epsilon_i \prod_{j\in G^c}(1-\epsilon_j)$ is the probability that the gates in G are in error and that the gates in G^c $(\Omega \backslash G)$ are error-free. Combining Eqns. (1) and (2) and summing S_k over all k yields the following expression for the probability of error $\delta_y(\vec{\epsilon})$:

$$\delta_y(\vec{\epsilon}) = 1/2\left(1 - \prod\nolimits_{i\in\Omega}(1-2\epsilon_i o_i)\right) \quad (3)$$

Eqn. (3) is a closed-form expression for the reliability of the output of a circuit as a function of error probabilities at each gate. Since the product of $(1 - 2\epsilon_i o_i)$ is over all gates in the circuit, it can be computed very efficiently once the observability of each gate is known. Eqn. (3) is intuitive because the error probability of each gate ϵ_i is scaled by o_i. Hence, errors at gates close to the primary outputs (high o_i) are more likely to cause output errors than errors at gates that are several levels of logic deep (low o_i).

3.1 Noise and correlation distort observability

Simulation results indicate that the closed-form expression for $\delta(\vec{\epsilon})$ is highly accurate for small circuits, and deviates by a small margin for ϵ close to 0.5. Note that the same value of gate failure probability has been used for each gate, and hence $\vec{\epsilon}$ is replaced

by ϵ. For example, the Monte Carlo and observability-based curves for $\delta(\vec{\epsilon})$ for the circuit in Fig. 1(a) are shown in Fig. 1(b). Simulation results also indicate that the closed-form expression performs well for small values of ϵ in large circuits, and that the accuracy depends on the number of gates in the circuit with $\epsilon > 0$. For example, Fig. 1(c) compares the $\delta(\vec{\epsilon})$ curves for a single output of the benchmark circuit b9, where a large error is observed as ϵ increases.

The observability-based reliability analysis is accurate for small ϵ because the probability of single gate failures is significantly higher than the probability of multiple gate failures. Since the effect of an error at a single gate is given by the gate failure probability scaled by its observability, it is exactly accounted for in the closed-form expression for reliability. As ϵ increases, the effect of multiple gate failures starts becoming significant and a deviation of the observability-based curve from the Monte Carlo curve is observed. There are two reasons for the inaccuracy of observability-based analysis in computing the effects of multiple gate failures. Both are related to the fact that the observability calculations are done statically:

i) On individual gates in the circuit: When observability computation is performed on gates one-at-a-time in the derivation of the closed-form expression for $\delta(\vec{\epsilon})$, the events of two or more gates being simultaneously observable is computed assuming that the events are independent. For instance, consider gates G_x and G_y in the circuit of Fig. 1(a). Assuming independence suggests that G_x is observable even when G_y is not because $o_x(1 - o_y) > 0$. However, since G_x is in the transitive fanin of G_y, it is clear that G_x is observable *only if* G_y is observable. Assuming independence thus introduces inaccuracies in the closed-form expression.

ii) In the absence of noise: When the observability calculations are performed in the absence of noise, it is assumed that a path remains sensitized irrespective of failures at gates that contribute to sensitizing that path. However, a failure at one or more of these gates may increase or decrease the observability of the original gate. For instance, consider gates G_x and G_z in the circuit of Fig. 1(a). Exhaustive analysis indicates that if both G_x and G_z fail, the probability of an output failure is $46/256$. However, the closed-form expression ignores the effects of how failures at G_z influence the propagation of failures from G_x and estimates this probability to be $19/256$. This problem is further exacerbated by the effects of reconvergent fanout that is common in logic circuits, since observability calculation at the source of reconvergent fanouts becomes more complex and expensive.

In conclusion, the observability-based closed-form expression is highly suitable for reliability analysis of small circuits and for small values of gate failure probabilities in large circuits. The algorithm is simple, yet efficient and flexible because a change in the value of noise at any gate(s) just requires recomputation of the closed-form expression (3). Since gate failure rates in current CMOS technologies are of the order of 10^{-6}–10^{-2}, it can easily be applied to reliability analysis and design optimization to enhance reliability.

4. Single-pass reliability analysis

The efficient single-pass reliability analysis technique described here addresses the accuracy drawbacks of the observability-based algorithm. At the core of this algorithm is the observation that an error at the output of any gate is the cumulative effect of a local error component attributed to the ϵ of the gate, and a propagated error component attributed to the failure of gates in its transitive fanin cone. When the components are combined, the total error probability at gate g is given by (i) a $0 \to 1$ error probability given that its error-free value is 0, $\Pr(g_{0\to1})$ and (ii) a $1 \to 0$ error probability given that its error-free value is 1, $\Pr(g_{1\to0})$.

Input vector	Weight	Weighted $0 \to 1$ input error component
00	$\mathcal{W}_{00}$	$\Pr(i_{0\to1})\Pr(j_{0\to1})\mathcal{W}_{00}$
01	$\mathcal{W}_{01}$	$\Pr(i_{0\to1})(1-\Pr(j_{1\to0}))\mathcal{W}_{01}$
10	$\mathcal{W}_{10}$	$(1-\Pr(i_{1\to0}))\Pr(j_{0\to1})\mathcal{W}_{10}$
Total	$\mathcal{W}(0)$	$\mathcal{PW}(0)$
Input vector	**Weight**	**Weighted $1 \to 0$ input error component**
11	$\mathcal{W}_{11}$	$\big(\Pr(i_{1\to0})+\Pr(j_{1\to0})-\Pr(i_{1\to0})\Pr(j_{1\to0})\big)\mathcal{W}_{11}$
Total	$\mathcal{W}(1)$	$\mathcal{PW}(1)$

Table 1: Expressions for weighted input error components

In general, $\Pr(g_{0\to1}) \neq \Pr(g_{1\to0})$ for an internal gate in a circuit. Initially, $\Pr(x_{i,0\to1})$ and $\Pr(x_{i,1\to0})$ are known for the primary inputs x_i of the circuit. In the core computational step of the algorithm, the $0 \to 1$ and $1 \to 0$ error components at the inputs to a gate are combined using a weight vector $\mathcal{W}$ to obtain a weighted input error vector $\mathcal{PW}$. The $\mathcal{PW}$ vector is then combined with the local gate failure probability ϵ to obtain $\Pr(g_{0\to1})$ and $\Pr(g_{1\to0})$ at the output of the gate. Computation of the (i) weight vector and (ii) weighted input error vector is described below.

Single-pass reliability analysis is performed by applying the core computational step of the algorithm recursively to the gates in a topological order. At the end of the single-pass, $\Pr(y_{0\to1})$ and $\Pr(y_{1\to0})$ is obtained for the output y of the circuit. The reliability δ_y of an output y is then given by the weighted sum of $\Pr(y_{0\to1})$ and $\Pr(y_{1\to0})$ as follows:

$$\delta_y(\epsilon) = \Pr(y=0)\Pr(y_{0\to1}) + \Pr(y=1)\Pr(y_{1\to0})$$

Given the weight vectors at all gates, the time complexity of the algorithm is $O(n)$, where n is the number of gates in the circuit. Note that single-pass reliability analysis gives the exact values of probability of error at the output in the absence of reconvergent fanout.

i) Weight vector: The weight vector for a gate stores the probability of occurrence of every combination of inputs at that gate. For instance, the weight vector of a 2-input (3-input) gate consists of 4 (8) entries. Since the weight vector is just the joint signal probability distribution of the inputs of a gate, it can be computed by random pattern simulation or symbolic techniques based on BDDs. Weight vectors are independent of $\vec{\epsilon}$ and change only if the structure of the logic circuit changes. To improve the efficiency of the algorithm, weight vector computation may be performed once at the beginning and used over several runs of reliability analysis. The BDDs for the gates in the circuit are manipulated to compute the components $\mathcal{W}_{00}$, $\mathcal{W}_{01}$, etc. of $\mathcal{W}$. For example, if b_1 and b_2 are the inputs to a gate, $\mathcal{W}_{00}$ is given by the number of minterms in $\overline{b}_1\overline{b}_2$ divided by the total number of input vectors to the circuit.

ii) Expressions for weighted input error vector: Expressions for the components of $\mathcal{PW}$, for a 2-input AND gate with inputs i and j, are given in Table 1. The calculation of $\mathcal{PW}(0)$ to propagate the $0 \to 1$ error component using the entries in the upper part of Table 1 is described here. Propagation of the $1 \to 0$ input error component is similar, using the entries in the lower part of Table 1.

Since the probability of a $0 \to 1$ error is actually the probability of a $0 \to 1$ error given that the error-free output of the gate is 0, there are only 3 rows in the upper table, one for each input vector for which the output of the AND gate is 0. The first column in the table is the input vector under consideration. The input vector has been ordered as ij. The second column is the probability of occurrence of the input vector, i.e., the weight vector. The third

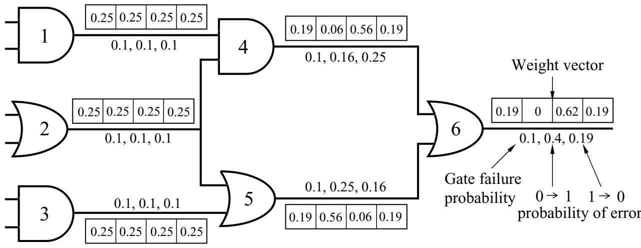

Figure 2: Illustration of single-pass reliability analysis

column is the probability of a $0 \to 1$ error at g, caused only due to errors at its inputs (when g itself does not fail). The entries in the third column are computed using $\Pr(i_{0\to1})$, $\Pr(i_{1\to0})$, $\Pr(j_{0\to1})$, and $\Pr(j_{1\to0})$ as illustrated below with an example.

Consider the input 10, whose error-free output is 0. For g to be in error only due to errors at the inputs, j has to fail and i has to be error-free so that the input to the gate is 11 instead of 10. Thus, the probability of a $0 \to 1$ error at g due to this input vector is $(1 - \Pr(i_{1\to0}))\Pr(j_{0\to1})$. To compute the effect of the input vector 10, this probability of error is weighted by its probability of occurrence, i.e., by $\mathcal{W}_{10}$. Thus, the value in the third column for the vector 10 is $\mathcal{W}_{10}(1 - \Pr(i_{0\to1}))\Pr(j_{0\to1})$. Similar entries for the inputs 00 and 01 are derived, and summed to obtain an expression for the weighted input error probability $\mathcal{PW}(0)$.

Since we are calculating the weighted $0 \to 1$ input error probability at g *given* that the error-free output is 0, $\mathcal{PW}(0)$ has to be divided by $\mathcal{W}(0)$ to restrict the inputs to a set for which the error-free output is 0. Thus, the weighted $0 \to 1$ and $1 \to 0$ input error probability at g are given by

$$\Pr(g_{0\to1}|g \text{ does not fail}) = \mathcal{PW}(0)/\mathcal{W}(0)$$
$$\Pr(g_{1\to0}|g \text{ does not fail}) = \mathcal{PW}(1)/\mathcal{W}(1)$$

iii) Expressions for $\Pr(g_{0\to1})$ and $\Pr(g_{1\to0})$: If g fails with a probability of ϵ, $\Pr(g_{0\to1})$ is given by

$$\Pr(g_{0\to1}) = (1-\epsilon)\left(\frac{\mathcal{PW}(0)}{\mathcal{W}(0)}\right) + \epsilon\left(1 - \frac{\mathcal{PW}(0)}{\mathcal{W}(0)}\right)$$

Similarly, $\Pr(g_{1\to0})$ is given by

$$\Pr(g_{1\to0}) = (1-\epsilon)\left(\frac{\mathcal{PW}(1)}{\mathcal{W}(1)}\right) + \epsilon\left(1 - \frac{\mathcal{PW}(1)}{\mathcal{W}(1)}\right)$$

Note that the two terms $(1 - \Pr(i_{1\to0}))$ and $\Pr(j_{0\to1}))$ are multiplied in the computation of the entries in the third column of Table 1. This implies that the events of i being correct and j failing are assumed independent. This assumption is valid if the gate is not a site for reconvergence of fanout. Since reconvergence causes the two events to be correlated, it is handled separately in Sec. 4.1.

Although the computation has been illustrated for an AND gate, the computation for an OR gate is symmetric, i.e., there are 3 rows for the probability of $1 \to 0$ error table and a single row for the probability of $0 \to 1$ error table. Inverters, nands, nors, and xors are all handled in a similar manner and the tables have been excluded for brevity.

Single-pass reliability analysis is illustrated for the circuit shown in Fig. 2. The weight vector, gate failure probability (ϵ), and probability of $0 \to 1$ and $1 \to 0$ error are indicated for each gate. The gates are numbered in the order in which they are processed. Since all the gates in the circuit have only 2-inputs, the weight vector for each gate consists of 4 entries. All entries of the weight vector for gate 1 are 0.25 because the primary input vectors are equally likely.

The fanout at gate 2 reconverges at gate 6 via gates 4 and 5. Thus, the event of $0 \to 1$ and $1 \to 0$ error at the outputs of gates 4 and 5 are correlated. However, independence is assumed and the probability of these events are used in the computation of $0 \to 1$ and $1 \to 0$ probability of error values for the output of gate 6.

4.1 Handling reconvergent fanout

The presence of reconvergent fanout renders the single-pass reliability analysis approximate because the events of $0 \to 1$ or $1 \to 0$ error for the inputs of a gate may not be independent at the point of reconvergence. Handling reconvergent fanout has been the subject of extensive research in signal probability computation. In this section, the theory of correlation co-efficients used in signal probability computation [8], is extended to make single-pass reliability analysis more accurate in the presence of reconvergent fanout.

This approach relies on the propagation of the correlation co-efficients for a pair of wires from the source of fanout to the point of reconvergence. Note that the word "wire" has been used as opposed to "node" because for a gate with fanout > 1, each fanout is treated as a separate wire, but they constitute the same node. The correlation co-efficient for events on a pair of wires is defined as the joint probability of the events divided by the product of their marginal probabilities. For signal probability computation, an event on a wire is defined as the value of the wire being 1. Thus, for a pair of wires, a single correlation co-efficient is sufficient to compute the joint probability of a 1 on both the wires.

In our analysis, an event is a defined as a $0 \to 1$ or $1 \to 0$ error on a wire. Hence, instead of a single correlation co-efficient, 4 correlation co-efficients for a pair of wires, one for every combination of events on the pair of wires. If *v* and *w* are two wires, the 4 correlation co-efficients for this pair are denoted by $\mathcal{C}_{vw}$, $\mathcal{C}_{v\tilde{w}}$, $\mathcal{C}_{\tilde{v}w}$, and $\mathcal{C}_{\tilde{v}\tilde{w}}$, where v, w, $\tilde{v}$, and $\tilde{w}$ refers to the event of a $0 \to 1$, $0 \to 1$, $1 \to 0$, and $1 \to 0$ error at v and w respectively.

The correlation co-efficients come into play at the the gates whose inputs are the site of reconvergence of fanout. At such gates, the events of $0 \to 1$ or $1 \to 0$ error at the inputs are not independent. Thus, the entries in the third column of Table 1 are weighted by the appropriate correlation co-efficient, e.g., $\Pr(i_{0\to1})(1 - \Pr(j_{1\to0}))$ becomes $\Pr(i_{0\to1})(1 - \Pr(j_{1\to0})\mathcal{C}_{i\tilde{j}})$.

Correlation co-efficient computation: The correlation co-efficient for a pair of wires can be calculated by first computing the correlation co-efficients for the wires in the fanout source that cause the correlation, and then propagating these correlation co-efficients along the appropriate paths leading to the pair of wires. Note that all four correlation co-efficients for two independent wires are 1. The computation of correlation co-efficients for the fanout source and the propagation of correlation co-efficients at a 2-input AND gate are described below.

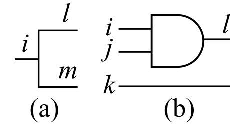

Figure 3: Computation/propagation of correlation co-efficient

i) Computation at fanout source node: The fanout source node i is shown in Fig. 3(a). The correlation co-efficient for the pair of wires $\{l, m\}$ is computed as follows:

$$\Pr(l_{0\to1}) = \Pr(l_{0\to1}, m_{0\to1}) = \Pr(l_{0\to1})\Pr(m_{0\to1})\mathcal{C}_{lm}$$
$$\text{i.e., } \mathcal{C}_{lm} = \frac{1}{\Pr(m_{0\to1})}$$

$$\Pr(l_{0\to1}|k_{0\to1}) = \epsilon + \frac{(1-2\epsilon)}{\mathcal{W}(0)}\big(\mathcal{W}_{00}\Pr(i_{0\to1}|k_{0\to1})\Pr(j_{0\to1}|k_{0\to1})\mathcal{C}_{ij} + \mathcal{W}_{01}\Pr(i_{0\to1}|k_{0\to1})(1-\Pr(j_{1\to0}|k_{0\to1})\mathcal{C}_{i\tilde{j}})$$
$$+\mathcal{W}_{10}(1-\Pr(i_{1\to0}|k_{0\to1})\mathcal{C}_{\tilde{i}j})\Pr(j_{0\to1}|k_{0\to1})\big)$$
$$= \epsilon + \frac{(1-2\epsilon)}{\mathcal{W}(0)}\big(\mathcal{W}_{00}\Pr(i_{0\to1})\mathcal{C}_{ik}\Pr(j_{0\to1})\mathcal{C}_{jk}\mathcal{C}_{ij} + \mathcal{W}_{01}\Pr(i_{0\to1})\mathcal{C}_{ik}(1-\Pr(j_{1\to0})\mathcal{C}_{\tilde{j}k}\mathcal{C}_{i\tilde{j}})$$
$$+\mathcal{W}_{10}(1-\Pr(i_{1\to0})\mathcal{C}_{\tilde{i}k}\mathcal{C}_{\tilde{i}j})\Pr(j_{1\to0})\mathcal{C}_{jk}\big)$$

Figure 4: Derivation of $\Pr(l_{0\to1}|k_{0\to1})$ in terms of correlation co-efficients of its inputs.

$\mathcal{C}_{\tilde{l}\tilde{m}}$ can be computed in a similar manner. $\mathcal{C}_{\tilde{l}m}$ and $\mathcal{C}_{l\tilde{m}}$ are zero because it is not possible to have a $0 \to 1$ error on m and a $1 \to 0$ error on l, or vice-versa.

ii) Propagation at an AND gate: Propagation of correlation co-efficients is illustrated for the AND gate in Fig. 3(b). Let i, j, k be three wires whose pairwise correlation co-efficients are known. Computation of the correlation co-efficients for the pair $\{l, k\}$ involves propagation of the correlation co-efficients through the AND gate, using the correlation co-efficients of i, j with k.

$$\mathcal{C}_{lk} = \frac{\Pr(l_{0\to1}|k_{0\to1})}{\Pr(l_{0\to1})}$$

The expression for $\Pr(l_{0\to1}|k_{0\to1})$ in terms of the correlation co-efficients of the inputs i, j with k is shown in Fig. 4. The terms in the expression for $\Pr(l_{0\to1}|k_{0\to1})$ are similar to the terms in the third column of the upper part of Table 1. The only difference is that the probability of $0 \to 1$ and $1 \to 0$ errors have been multiplied by appropriate correlation co-efficients. Note that the terms of the weight vector $\mathcal{W}$ include the signal probability of k. The expression for $\mathcal{C}_{\tilde{l}k}$ is derived in a similar manner using the lower part of Table 1, and is left out for brevity. Expressions for $\mathcal{C}_{l\tilde{k}}$ and $\mathcal{C}_{\tilde{l}\tilde{k}}$ are derived by replacing k by $\tilde{k}$ in the expressions for $\mathcal{C}_{lk}$ and $\mathcal{C}_{\tilde{l}k}$ respectively. In Fig. 5, the consolidated probability of error at two correlated primary outputs of benchmark circuit b9 is used to illustrate the accuracy achieved with correlation co-efficients.

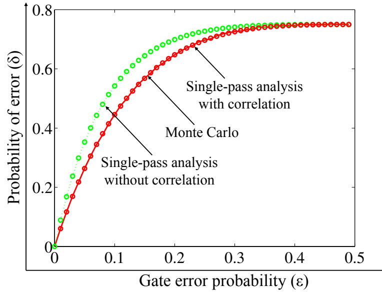

Figure 5: Handling reconvergent fanout in single-pass reliability analysis with correlation co-efficients.

5. Results

Simulation results comparing single-pass reliability analysis with Monte Carlo simulations are reported in Table 2. The simulations were run on a 2.4 GHz Opteron-based system with 4 GB of memory. A 64-bit parallel pattern simulator was used to implement a Monte Carlo framework for reliability analysis based upon fault injection. The sample size used for reliability analysis was 6.4 million random patterns. In the table, columns 1 and 2 give the name and number of gates in the benchmark circuit. Both the Monte Carlo and single-pass reliability frameworks were used to compute

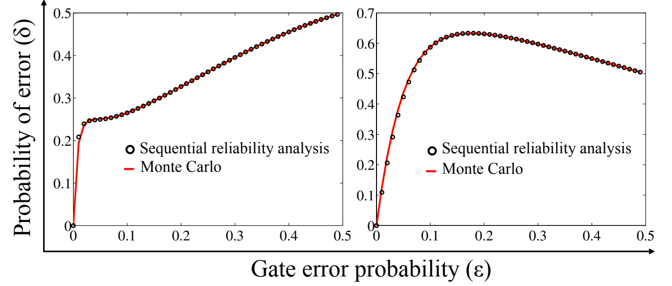

Figure 6: $\delta(\vec{\epsilon})$ curves for two outputs of i10.

$\delta(\vec{\epsilon})$ for 50 different values of ϵ over the range 0 to 0.5. Note that the same value of ϵ has been used for all the gates in the circuit, and hence $\vec{\epsilon}$ is replaced by ϵ. The third column reports the percentage error in single-pass reliability analysis for 6 values of ϵ ranging from 0.05 to 0.3. For each entry, the error in $\delta(\vec{\epsilon})$ with respect to Monte Carlo simulation is measured, and the average error over all outputs is reported. For $\epsilon > 0.3$, $\delta(\vec{\epsilon})$ saturates at 0.5 for several outputs and is hence not reported here. The cumulative run-time for 50 runs is reported in the fourth column.

The maximum percentage error in $\delta(\vec{\epsilon})$ is less than 3% for the largest benchmark circuit, i10. For circuits with significant reconvergent fanout, e.g., c499 and c1355, the maximum percentage error in $\delta(\vec{\epsilon})$ is 12.16% and 8.91% respectively. It is clear from the results that the proposed single-pass reliability analysis technique is highly accurate. Although a head-to-head performance comparison with approaches based on PTMs and Bayesian networks was not possible, it is our belief based on the results reported in [6] that the proposed technique affords at least a 500X speed-up over Bayesian networks on the largest circuit b9 (2.5s versus 0.005s (0.25/50s)) reported therein. Note also that results reported in [6] show that Bayesian networks afford a 1000X speed-up over PTMs. In summary, it is reasonable to conclude that the strengths of the proposed single-pass reliability analysis algorithm are its accuracy, scalability to large circuits, and speed-up in performance.

Fig. 6 presents $\delta(\vec{\epsilon})$ for two outputs of benchmark i10. The cone sizes of the two outputs are 662 and 1034 gates respectively. Each graph has two curves, one from Monte Carlo reliability analysis and one from single-pass reliability analysis. The two curves are indistinguishable as seen in the figure. The diverse shapes of the curves illustrates not only the complexity of the relation between δ and $\vec{\epsilon}$, but also the accuracy of single-pass reliability analysis.

Fig. 7 shows the percentage error in $\delta(\vec{\epsilon})$ for each of the 32 outputs of benchmark circuit c499. On each run, the ϵ for each gate was derived from a uniform random distribution over the interval $[0, 0.5]$. The percentage error in $\delta(\vec{\epsilon})$ for each output, averaged over 1000 runs, is 1.5–3.5%. This illustrates that the single-pass reliability analysis is highly accurate even when the ϵ values are allowed to vary independently at every gate.

Table 2: Simulation results for reliability analysis. Six values of ϵ are used for comparison.

Benchmark	Size	Average error over all outputs (in %)						Runtimes (for 50 runs)	
		$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.15$	$\epsilon = 0.2$	$\epsilon = 0.25$	$\epsilon = 0.3$	Monte Carlo	Single-pass analysis
x2	56	1.3	0.92	0.52	0.28	0.15	0.08	8m 41s	0.065s
cu	59	1.58	0.83	0.37	0.14	0.09	0.06	9m 50s	0.107s
b9	210	0.3	0.22	0.12	0.07	0.06	0.03	37m 15s	0.25s
c499	650	12.16	9.63	6.97	4.61	2.75	1.43	134m 55s	1.91s
c1355	653	8.91	7.48	5.58	3.79	2.32	1.24	135m 7s	2.09s
c1908	699	8.67	6.06	4.42	3	1.84	1	145m 5s	0.781s
c2670	756	3.04	1.99	1.35	0.88	0.54	0.31	208m 41s	2m 51.2s
frg2	1024	2.4	1.53	0.94	0.54	0.3	0.15	286m 38s	0.533s
c3540	1466	6.2	2.67	1.18	0.53	0.23	0.11	431m	5m 42s
i10	2643	2.43	1.58	1.01	0.62	0.37	0.21	1668m 44s	5.42s

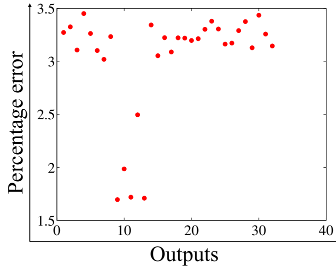

Figure 7: Average error in $\delta(\vec{\epsilon})$ per output of circuit c499 over 1000 runs. On each run, $\epsilon_i \in \text{Uniform}(0, 0.5)$ for each gate.

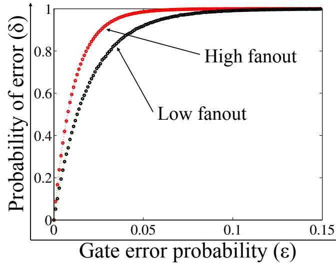

Figure 8: Redundancy-free improvements in reliability

5.1 Applications

Single-pass reliability analysis can be used in redundancy-free design space exploration. It is called redundancy-free because no redundancy is used in the circuit to achieve improvements in reliability. This is illustrated using two synthesized versions of benchmark b9: (i) a low fanout version with a maximum fanout of 2 and (ii) a high fanout version with a maximum fanout of 6. Fig. 8 compares the consolidated output error curves for the two versions of b9. The consolidated output error curve gives the probability that at least one of the outputs is in error, and is obtained by performing correlation-based analysis described in Sec. 4.1 on the individual $\delta(\vec{\epsilon})$ curves. In Fig. 8, $\epsilon \in [0, 0.15]$ because $\delta(\vec{\epsilon})$ for both circuits saturates at 1 for $\epsilon > 0.15$. Note that the same value of ϵ has been used for all the gates, and hence $\vec{\epsilon}$ is replaced by ϵ. It is clear from the figure that the low fanout version of b9 has higher reliability than the high fanout version. This can be explained by examining the levels of logic present in both circuits. The high (low) fanout version of b9 has a maximum of 12 (9) levels of logic and a total of 164 (111) levels of logic over all the outputs. As the number of levels of logic increase, the noise-free inputs have to pass through more levels of noise before they reach the primary outputs. This results in a higher consolidated output error probability.

Single-pass reliability analysis also provides $\delta(\vec{\epsilon})$ curves for each node in the circuit. This information can be used to introduce redundancy at selected gates, instead of introducing redundancy at every gate in the circuit. The proposed analysis technique also provides information about the $0 \rightarrow 1$ and $1 \rightarrow 0$ probability of error separately at each node in the circuit. This is valuable information for explicit introduction of asymmetric redundancy. For instance, in quadded logic, the redundancy introduced for mitigating a $0 \rightarrow 1$ and $1 \rightarrow 0$ error are different by construction. Single-pass reliability analysis can be used to direct such fine-grained insertion of asymmetric redundancy to enhance reliability at a lower cost.

Observability-based reliability analysis is accurate when the probability of a single gate failure is significantly higher than the probability of multiple gate failures. This makes it directly applicable for soft-error rate estimation in logic circuits because failures due to single-event upsets are usually localized to the gate that is the site of the strike.

6. Conclusions

Even as reliability gains wide acceptance as a significant design challenge, there is a lack of effective techniques for its analysis and optimization. This paper described two accurate, scalable, and highly efficient techniques for reliability analysis of logic circuits. These techniques have several potential applications, including redundancy-free reliability optimization, asymmetric and fine-grained redundancy insertion, and reliability-driven design optimization.

7. References

[1] J. D. Meindl *et al.*, “Limits on silicon nanoelectronis for terascale integration,” *Science*, vol. 293, pp. 2044–2049, Sep. 2001.

[2] G. Bourianoff, “The future of nanocomputing,” *IEEE Computer*, vol. 36, pp. 44–53, Aug. 2003.

[3] 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, 1956.

[4] A. Sadek, K. Nikolić, and M. Forshaw, “Parallel information and computation with restitution for noise-tolerant nanoscale logic networks,” *Nanotechnology*, vol. 15, pp. 192–210, Jan. 2004.

[5] S. Krishnaswamy *et al.*, “Accurate reliability evaluation and enhancement via probabilistic transfer matrices,” in *Proc. Design Automation and Test in Europe*, pp. 282–287, 2005.

[6] T. Rejimon and S. Bhanja, “Scalable probabilistic computing models using Bayesian networks,” in *Proc. Intl. Midwest Symposium on Circuits and Systems*, pp. 712–715, 2005.

[7] T. Larrabee, “Test pattern generation using Boolean satisfiability,” *IEEE Trans. Computer-aided Design*, vol. 11, pp. 4–15, Jan. 1992.

[8] S. Ercolani *et al.*, “Estimate of signal probability in combinational logic networks,” in *Proc. European Test Conference*, pp. 132–138, 1989.