J Electron Test (2013) 29:143–158
DOI 10.1007/s10836-013-5365-0

CEP: Correlated Error Propagation for Hierarchical Soft Error Analysis

Liang Chen · Mojtaba Ebrahimi · Mehdi B. Tahoori

Received: 1 October 2012 / Accepted: 28 February 2013 / Published online: 2 April 2013

Abstract Due to the continuous technology scaling, soft error becomes a major reliability issue at nanoscale technologies. Single or multiple event transients at low levels can result in multiple correlated bit flips at logic or higher abstraction levels. Addressing this correlation is essential for accurate low-level soft error rate estimation, and more importantly, for the cross-level error abstraction, e.g. from bit errors at logic level to word errors at register-transfer level. This paper proposes a novel error estimation method to take into consideration both signal and error correlations. It unifies the treatment of error-free signals and erroneous signals, so that the computation of error probabilities and correlations can be carried out using techniques for signal probabilities and correlations calculation. The proposed method not only reports accurate error probabilities when internal gates are impaired by soft errors, but also gives quantification of the error correlations in their propagation process. This feature enables our method to be a versatile technique used in high-level error estimation. The experimental results validate our proposed technique showing that compared with Monte-Carlo simulation, it is 5 orders of magnitude faster, while the average inaccuracy of error probability estimation is only 0.02.

Keywords Soft error · Error probability · Logic masking · Signal correlation

Responsible Editor: H. Manhaeve

L. Chen (✉) · M. Ebrahimi · M. B. Tahoori
Institute of Computer Science and Engineering,
Karlsruhe Institute of Technology
Haid-und-Neu Str.7, Geb. 07.21, 76131 Karlsruhe, Germany
e-mail: liang.chen@kit.edu

1 Introduction

Radiation-induced soft error is becoming one of the major sources of unreliability in nano-scale digital circuits [5, 14]. Due to the shrinking transistor size and reduced operating voltage, particle strikes with low energy can generate sufficient charge to cause a soft error [3]. As a result, not only the rate of single transient error in logic circuits is becoming comparable with that of memories [20, 34], but also the occurring probability of multiple transient errors is no longer negligible [23, 31]. Therefore, in addition to traditional safety and mission critical applications, the mainstream systems also require resilient design against soft errors [33]. To improve the reliability of system with minimum impact on performance, area and power, fast and accurate soft error estimation is essential throughout various design phases.

In digital circuits, a single event transient at one circuit node can propagate to multiple outputs, and be captured by multiple flip-flops in the same clock cycle. The way this single event transient is seen at higher abstraction levels could typically manifest as *multiple correlated* errors (bit flips). This correlation is particularly important for high level error modeling of the commonly used multi-bit data/control words, IP macro interfaces and so on. Assuming independence among these bits can severely reduce the accuracy of soft error modeling at higher abstraction levels, and in turn the efficiency of the mitigation techniques.

However, currently at register-transfer or higher (e.g. architectural) levels, single bit-flip or multiple *independent* bit-flips are assumed and these errors are randomly injected for simulation-based techniques [17, 36]. The same assumption is used for analytical approaches as well [19, 25]. Although circuit and logic level techniques consider

detailed erroneous glitches for accurate soft error rate estimation [22, 28, 29, 38, 39], they do not quantify correlations among different erroneous signals.

To close this gap between low level error estimation and high level error abstraction, this paper proposes a novel method to explicitly consider and report quantified error correlations in soft error analysis. It is based on the concept of *error propagation function* and a new *super gate* representation to unify the treatment of error-free and erroneous signals. It not only calculates both signal and error probabilities in one pass, but also takes the complex correlations among them into account. The correlation coefficient method originating from signal probability calculation is adjusted to obtain error probabilities and correlations of primary outputs due to particle strike at internal nodes. The experimental results show that our method is five orders of magnitude faster than Monte-Carlo simulation while the average inaccuracy[1] of error probability estimation is only 0.02.

Preliminary versions of this work have been published in [6, 7]. Compared with our previous work on correlated soft error analysis, in this paper we carried out several significant extensions and made additional important contributions. Two efficient heuristic algorithms are introduced to further improve the scalability of our method:

- *Dynamic blocking of error propagation* dynamically sets gates with very low error probabilities as error-free to speedup the calculation with negligible accuracy loss.
- *Limited depth correlation analysis* reveals the opportunity to consider only highly correlated signal pairs to significantly improve the scalability.

These heuristics make our framework reach three orders of magnitude speedup compared with [7], while still maintaining satisfactory accuracy. In addition, in this paper we present a detailed case study of OpenRISC ALU to illustrate the necessity of error correlation analysis and the applicability of our proposed methodology.

The organization of the rest of this paper is as follows. Section 2 reviews related work on soft error and circuit reliability estimation. Section 3 introduces our adopted correlation model. Sections 4 discusses different procedures to calculate correlated error probabilities based on the proposed super gate representation and Section 5 discusses the extensions of our method. Section 6 describes the experimental results. Finally, the paper is concluded in Section 7.

2 Related Work

Soft error estimation techniques can be generally categorized into statistical fault injection [2, 11, 36] and analytical approaches [19, 22, 25, 28, 29, 38, 39]. Fault injection techniques are based on Monte-Carlo simulations and to achieve a reasonable accuracy, they require long simulation time. To overcome the scalability of fault injection, analytical probabilistic approaches have been developed and can be orders of magnitude faster than fault injection techniques. However, the accuracy and efficiency of analytical approaches rely heavily on the error and error propagation models. The circuit-level techniques presented in [22, 28, 29, 38, 39] handle three masking effects, namely logical, electrical and timing masking, with detailed models and obtain accurate results for soft error rate estimation, but they do not have good scalability and are not suitable for error analysis at high level.

Additionally, in the scope of correlation modeling for logic errors (bit-flips), the technique proposed in [39] uses random vector simulation to estimate the probability of logical masking, which does not give error correlations and furthermore, requires high simulation time for large circuits. In the techniques [22, 28, 29, 38], only the error probabilities of primary outputs are obtained without providing their correlations, which, however, are indispensable for lifting the abstraction level of soft error analysis. The error propagation probability technique [1] has better scalability with respect to runtime, but it addresses only the reconvergent effects of error propagation using 4-value logic. The error-free signals are assumed to be uncorrelated, even for the inputs of the reconvergent gate. The inaccuracy due to this simplification impairs its runtime advantage and again, it cannot report the error correlations at circuit outputs.

With bit-flip model, soft error probability estimation at logic level can also be handled by circuit reliability estimation techniques [8, 18, 24, 30, 35, 37], where each gate is assumed to be unreliable and has a specified failure probability. Regarding the modeling of error correlations, probabilistic transfer matrix approach [18] employs algebraic decision diagrams to facilitate matrix operations and capture correlations among internal signals, but it suffers from massive matrix storage and high runtime. Bayesian network approach [30] uses conditional probability table for each gate to capture all internal signal correlations, however, manipulating large circuits using Bayesian network potentially need very high runtime.

[1] As Monte-Carlo simulation can not obtain the *exact* results either, the *inaccuracy* here actually means the deviation of our estimated error probabilities from the results reported by Monte-Caro simulation.

One-pass reliability method [8] uses correlation coefficients to model the correlations of two bit-flip events, 0→1 and 1→0, on different wires. Four correlation coefficients are necessary for each pair of wires and up to 16 correlation coefficients if higher accuracy is required. Trigonometry-based probability calculation [37] uses trigonometric functions to manipulate signal and error probabilities, and obtains accurate results for low error probabilities ($<10^{-4}$), but it is not appropriate for large error probabilities. Boolean Difference-based Error Calculator [24] adopts mathematical differential equations to model error propagation, and several levels of logic from the fanout node are collapsed into one level to handle reconvergent effects, however, its scalability is questionable for larger correlated regions. The hybrid approach [35] combines exact analysis with probabilistic measures to estimate reliability, and establishes algebraic decision diagrams for each reconvergent region, which limits its application to large circuits.

To sum up, in the scope of error correlation with bit-flip model, the related work mentioned above suffers from at least one of the following disadvantages:

- Only the error probabilities of primary outputs are obtained, therefore it cannot report quantified error correlations.
- Due to the intrinsic scalability issue its applicability is always limited to small or medium size circuits.

Therefore, in this paper we propose a new technique to overcome these two disadvantages, aiming at quantification of error correlations to support hierarchical soft error analysis and scalability enhancement to handle large size circuits.

3 Correlation Model

In typical digital circuits, there are two kinds of correlations: *temporal* and *spatial* correlations [21]. Temporal correlation is related to the historical trends of bit streams, which is always considered in switching activity calculation and it is beyond the scope of this work on error propagation estimation in combinational circuits.

Spatial correlation, which describes the dependencies between signals at different locations in the netlist, originates from two main sources:

- *Structural dependence*: It is due to the reconvergent fanout, where two or more signals originate from the same gate, propagate on different paths and converge again to the inputs of another gate.
- *Primary input dependence*: It refers to the spatial correlations among primary input signals, which result from input vectors with workload dependencies.

The most familiar measure of dependence between two variables is the *Pearson product-moment Correlation Coefficient (PCC)*. It is obtained by dividing the covariance of the two variables with the product of their standard deviations [27]:

$$\rho_{X,Y} = \mathrm{corr}(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{1}$$

where X, Y are random variables, μ_X, μ_Y are the mean values and σ_X, σ_Y are the standard deviations. If X are Bernoulli variables (e.g. the logic signal with binary values), we know that

$$E(X) = P(X), \quad \sigma_X = \sqrt{P(X)(1-P(X))} \tag{2}$$

where $P(X)$ is the probability of X, therefore the PCC of two Bernoulli variables can be expressed as:

$$\begin{aligned} \rho_{X,Y} &= \frac{E[(X-P(X))(Y-P(Y))]}{\sqrt{P(X)(1-P(X))P(Y)(1-P(Y))}} \\ &= \frac{P(XY)-P(X)P(Y)}{\sqrt{P(X)(1-P(X))P(Y)(1-P(Y))}} \end{aligned} \tag{3}$$

The value of the $\rho_{X,Y}$ ranges from -1 to 1. A value of 0 implies that there is no linear relationship between the two variables, while the value of -1(1) implies they are perfectly negative(positive) correlated, i.e. whenever one variable has a high/low value, the other has a low/high(high/low)value.

3.1 Correlation Coefficient Method

PCC can clearly indicate the degree of correlation between two variables, but it is too complicated to be used for the correlation modeling in combinational network. To make it easier, *Correlation Coefficient Method* (CCM) [12] is adopted in our work for two main reasons. First, it provides accurate results, while has better scalability than the approaches in [18, 30]. Second, CCM can be easily extended to address the primary input dependencies as well [21].

3.1.1 Algorithm

Based on the topologically levelized circuit netlist, CCM propagates both signal probabilities and signal correlations level by level from primary inputs to primary outputs. It considers the correlations between any two signals in each level, therefore can handle both structural dependencies and primary input dependencies effectively.

In CCM with the notation of signal probability $p(i) \equiv P(i = 1)$, the *correlation coefficient* (CC) of signals i, j is defined as

$$C_{i,j} = C_{j,i} = \frac{p(ij)}{p(i)p(j)} = \frac{p(i|j)}{p(i)} = \frac{p(j|i)}{p(j)} \tag{4}$$

where $p(ij)$ is the joint probability $P(i = 1, j = 1)$, $p(i|j)$ is the conditional probability $P(i = 1|j = 1)$, therefore $C_{i,j}$ is in the range $[0, +\infty)$.

Compared with Eq. 3, it is clear that $C_{i,j}$ and $\rho_{i,j}$ can be derived from each other. For the special case that two signals are uncorrelated,

$$p(ij) = p(i)p(j) \rightarrow C_{i,j} = 1,\ \rho_{i,j} = 0$$

In this paper, PCC is only used to indicate the degree of correlation between signals, due to its normalized value range $[-1, 1]$, while CC is used mainly for correlation propagation in the circuits.

Take an *AND* gate $l = ij$ as example, given p_i, p_j and $C_{i,j}$ we can exactly calculate the signal probability of the output l using the following formula:

$$p(l) = p(i)p(j)C_{i,j}, \quad 0 \le C_{i,j} \le \frac{1}{p(i)p(j)} \tag{5}$$

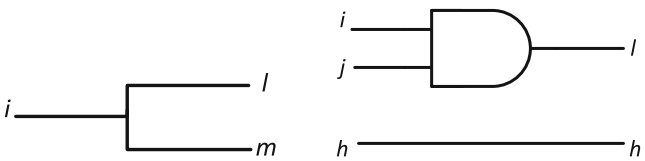
In addition, based on several basic propagation rules, the correlation coefficients between different signals can be analytically computed for all structural cases in the combinational network. Two typical cases are illustrated in Fig. 1 and the corresponding correlation formulas are $C_{l,m} = 1/p(i)$ and $C_{l,h} = C_{i,h}C_{j,h}$, respectively. Note that in CCM for a gate with fanout > 1, each fanout is treated as a separate wire for the correlation calculation.

3.1.2 Accuracy Issue

In Fig. 1b it is assumed that

$$C_{ij,h} \approx C_{i,h}C_{j,h} \tag{6}$$

thus the dependencies of two signals to a third one is neglected. Hence the second and higher order correlations among multiple signals are not taken into account in CCM. The signal probability estimation in [12] and switching activity estimation in [21] show this first-order approximation can provide accurate results in practice. However, neglecting high order correlations may cause the gate output probability to be smaller than 0.0 or larger than 1.0. Therefore, Inequality (5) is used to limit $C_{i,j}$ of the *AND* gate to avoid probability overflow. For other logic gates, similar inequalities can be derived for the purpose of probability bounding.



(a) Fanout node (b) *AND* gate and wire

Fig. 1 Typical structures for correlation calculation

Actually, we find that the upper bound of correlation coefficient in Inequality (5) is rather loose. Revisiting the definition of correlation coefficient in Eq. 4, we have:

$$C_{i,j} = \frac{p(i|j)}{p(i)} = \frac{p(j|i)}{p(j)}, \quad p(i|j) \le 1 \text{ and } p(j|i) \le 1$$
$$\Longrightarrow C_{i,j} \le min\{\tfrac{1}{p(i)}, \tfrac{1}{p(j)}\} \tag{7}$$

This new inequality gives a tighter upper bound, therefore provides better error bounding in the correlation propagation, especially for the signals with high order correlation.

3.1.3 Complexity Issue

It is shown in [12] that the complexity of CCM is linear in the number of topological levels L and pseudo-quadratic in the number of gates per level N_L, which could be expressed as

$$\text{Complexity of CCM} \le \sum_{L} N_L(N_L - 1)/2 \tag{8}$$

Please note that the right side of this inequality is the worst case, because to get output probabilities CCM only needs to calculate the correlation coefficients for these signals that are dependent from each other.

3.2 Error Correlation

In the scope of soft error modeling, the irregular structures in combinational logic make the error correlation modeling more complicated than that in the regular structures such as memory arrays.

After error occurs and propagates along multiple reconvergent paths, there exist various types of correlations, as shown in Fig. 2. Not only the error-free signals, whose probabilities determine the logical masking effects, but also the erroneous signals, are possible to be correlated (referred as *Signal Correlation* and *Error Correlation*, respectively). To make things worse, even the signal probabilities of error-free signals and error probabilities of erroneous signals are not independent (referred as *Cross Correlation*). It is indispensable to consider all these correlations to achieve good accuracy for error estimation.

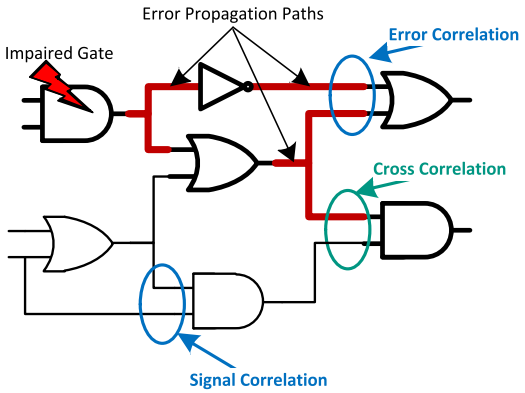


Fig. 2 Different types of correlation

4 Proposed Error Estimation Methodology

Soft errors are modeled as bit flips in our paper. For the sake of simplicity, we present our approach for the circuit mapped into a network of two-input elementary gates. Nevertheless, the presented approach can readily be applied to a network of arbitrary logic cells.

The proposed methodology for error probabilities and correlations estimation is divided into three main parts. First, error propagation in the network of elementary gates is efficiently modeled. Second, combining this error propagation model with CCM enables us to estimate both error probabilities and correlations. Third, due to the intrinsic complexity issue of CCM, scalability enhancement techniques are explored to make our method applicable to large size circuits.

4.1 Error Propagation Model

To illustrate the basic idea, we take the boolean *AND* function $l = ij$ as the running example. Totally there are four error-free input combinations. For $ij = 01$, only when i is erroneous (0→1) and j is error-free, the output l is erroneous (0→1). The other three combinations could be analyzed similarly.

If notation x_f is used to indicate whether the signal x is erroneous, i.e. $x_f = 1$ means a bit-flip occurs on signal x, caused by either particle strike or error propagation, the four cases for the *AND* gate can be combined and expressed with a single boolean function, called *Error Propagation Function (EPF)*:

$$l_f = i\bar{j}\bar{i_f}j_f + \bar{i}\bar{j}i_f j_f + \bar{i}ji_f\bar{j_f} + ij(i_f + j_f) \tag{9}$$

here i_f, j_f, l_f are just virtual error signals to facilitate the modeling of error propagation, and do not exist in real circuits. Although the virtual error signals are treated as normal ones, their property interpretations are different:

- *Logic value*: '1' is interpreted as bit-flip occurring in the corresponding real signal and '0' means error-free;
- *Signal probability (SP)*: interpreted as the error probability of the corresponding real signal.

If the error-free function $l = ij$ is combined with EPF, a four-input, two-output *super gate* can be constructed, as illustrated in Fig. 3.

This representation has several important features:

- *It simplifies the propagation and correlation modeling of the bit-flips in combinational network*: it does not differentiate two kinds of bit flips: 0→1 and 1→0 as in [8]. To model all three types of correlations between two wires in Fig. 2, our method needs only $\binom{4}{2} = 6$ rather than 16 correlation coefficients in [8].
- *The super gate representation is independent of any specific algorithm*: it is just an additional boolean function and could be analyzed using the well-known methods in the areas of signal probability [12], switching activity [21], power estimation [9], etc.
- *This concept can be easily extended to more complicated error modeling*: it can be adjusted for modeling single bit-flip, multiple bit-flips, transient errors, permanent errors, etc.

This modeling technique doubles the number of signals needed to be considered in the analysis. Recalling the runtime Inequality (8) and the intuitive gate-level implementation of super *AND* in Fig. 4, it is clear that the super-gate implementation increases not only the circuit level L, but also the number of gates N_L at each level, therefore introducing high runtime overhead.

This additional complexity is unavoidable to model propagation of bit-flip errors and their complicated correlations. Nevertheless, this complexity problem can be alleviated by exploration of the logic properties of EPF and limited correlation heuristic, as discussed later in this section.

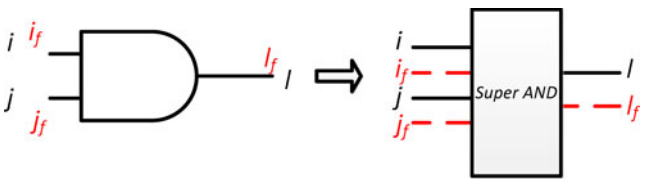
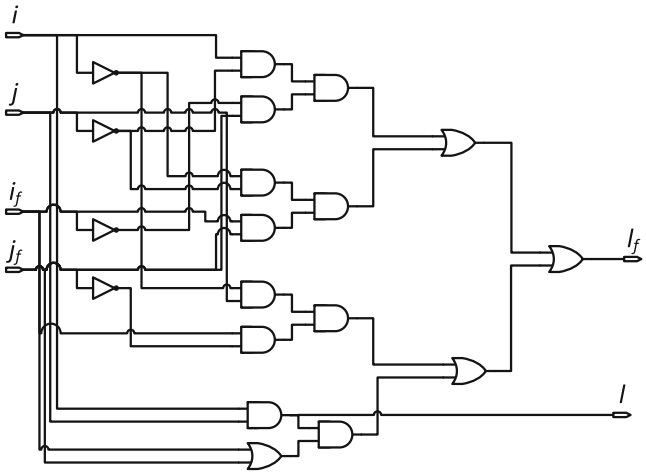
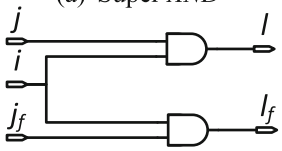


Fig. 3 Super gate representation



(a) Super *AND*



(b) Semi-super *AND*

Fig. 4 Super and semi-super *AND* gate-level implementation

4.2 Error Cone Extraction

Due to the complexity issue of CCM in Section 3.1.3, the straightforward approach to uniformly replace each gate in the original circuit with its corresponding super gate is not feasible. Actually, in the scope of soft error only the gates in the fanout cone of the error site (i.e. *error cone*) will contribute to the error probabilities of primary outputs, as shown in Fig. 5.

Moreover, for the gates at the boundary of the error cone, only one of its two inputs can be erroneous. This observation can be explored to efficiently reduce the complexity of super gate implementation in Fig. 4. Recalling the EPF in Eq. 9, and assuming the input i is error-free, i.e. $i_f = 0$, this EPF can be simplified as following:

$$l_f = i\bar{j}j_f + ijj_f = ij_f \tag{10}$$

In this way, the complex super gate collapses to only two basic gates, called *semi-super gate*: one for the error-free function, the other for the error propagation. Obviously, on the error propagation paths, the more gates collapsing into semi-super gate, the more benefit we get from this simplification, as shown in Fig. 5.

4.3 Super-gate Correlation Formulas

To avoid replacing super/semi-super gates with their complex gate-level implementation and the corresponding netlist transformation efforts, it is necessary to derive error

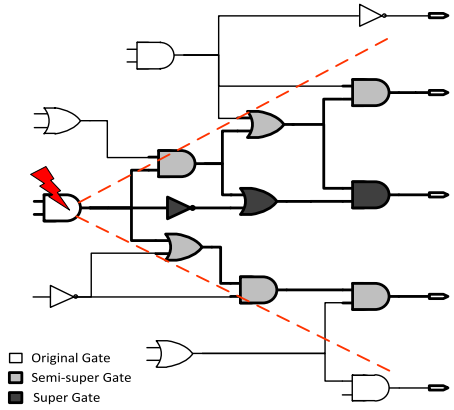


Fig. 5 Error propagation paths and error cone

probability and correlation propagation formulas for super, semi-super and original gate pairs [7].

The EPF of *AND* gate in Eq. 9 can be rewritten as

$$\begin{aligned} l &= ij \\ l_f &= \underbrace{i\bar{j}\bar{i}_f j_f}_{l_{f_0}} + \underbrace{\bar{i}\bar{j}i_f j_f}_{l_{f_1}} + \underbrace{\bar{i}ji_f\bar{j}_f}_{l_{f_2}} + \underbrace{ij(i_f + j_f)}_{l_{f_3}} \end{aligned} \tag{11}$$

One very important property of this function is that the four terms $l_{f_0}, l_{f_1}, l_{f_2}, l_{f_3}$ are *mutually exclusive*, which means

$$\begin{aligned} &p(l_{f_i} l_{f_j}) = 0, \quad i, j = 0, 1, 2, 3 \text{ and } i \neq j \\ &p(l_f) = p(l_{f_0}) + p(l_{f_1}) + p(l_{f_2}) + p(l_{f_3}) \end{aligned} \tag{12}$$

This logical property can be explored to simplify and accelerate the calculation of error probabilities, cross correlations and error correlations.

For the calculation of probability $p(l_f)$, each of the four terms could be calculated using basic gate rules introduced in Section 3.1, as every term contains only four variables. For the correlation coefficient C_{l,l_f}, instead of using the original CCM for gate-level implementation of EPF, with the boolean Eq. 11 we turn to correlation coefficient definition in Eq. 4 to derive this value:

$$\begin{aligned} p(ll_{f_i}) &= 0, \quad i = 0, 1, 2 \\ C_{l,l_f} = \frac{p(ll_f)}{p(l)p(l_f)} &= \frac{p[l(l_{f_0} + l_{f_1} + l_{f_2} + l_{f_3})]}{p(l)p(l_f)} \\ &= \frac{p(ll_{f_0}) + p(ll_{f_1}) + p(ll_{f_2}) + p(ll_{f_3})}{p(l)p(l_f)} \\ &= \frac{p(ll_{f_3})}{p(l)p(l_f)} = \frac{p(l_{f_3})}{p(l)p(l_f)} \end{aligned} \tag{13}$$

where $p(l_{f_3})$ and $p(l_f)$ can be obtained from Eq. 12 and $p(l)$ is the signal probability from error-free CCM.

To derive the general formulas for typical structures, e.g. Super-*AND*/wire pair in Fig. 6, a similar approach can

be applied to avoid unnecessary correlation computation efforts:

$$C_{h,l_f} = \frac{p(hl_f)}{p(h)p(l_f)} = \frac{p\left[h\left(l_{f_0}+l_{f_1}+l_{f_2}+l_{f_3}\right)\right]}{p(h)p(l_f)} = \frac{p(hl_{f_0})+p(hl_{f_1})+p(hl_{f_2})+p(hl_{f_3})}{p(h)p(l_f)}$$

Based on the mutually exclusive property, the complex EPF is broken up into four small parts, each of which can be calculated quickly.

Using De Morgan's law, and since *INV* gate does not logically mask any error, the above Super-*AND* correlation rules can be extended to Super-*OR* with minor modifications as shown in Fig. 7. Furthermore, for EPFs of the simplified semi-super gate, the derivation of corresponding formulas is straightforward and omitted here for brevity.

4.4 Dynamic Blocking of Error Propagation

Our focus is probabilistic analysis of the soft error propagation in combinational networks, and the area having erroneous signals is always limited to the error cone. In addition to the techniques introduced in Sections 4.2 and 4.3, *dynamic blocking of error propagation* can also be employed to speedup the calculation.

This idea comes from the observation that due to logical masking, the longer the error propagation path is, the smaller the error probability could be. Therefore, when the calculated output error probability of one gate is smaller than predefined threshold value, this gate could be assumed to be error-free, i.e. its type changes from super gate or semi-super gate to original gate, its error probability is reset to constant zero and the correlations between its corresponding virtual error signal to all other signals are eliminated. The choice of this predefined threshold is a trade-off between scalability and accuracy, and in our implementation this value is set to 10^{-4}. Note that this gate type change may cause the type of its successor chain to be dynamically adjusted, e.g. from super gate to semi-super gate or semi-super to original gate. Therefore, this dynamic error resetting technique can reduce the complexity of our estimation with negligible accuracy impact, especially for circuits where long propagation paths exist.

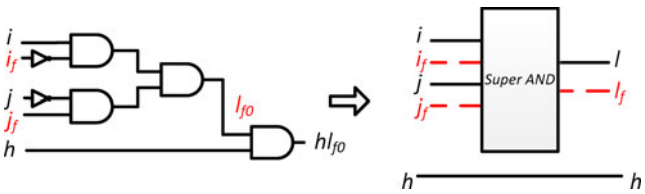


Fig. 6 Super-*AND* and wire correlation

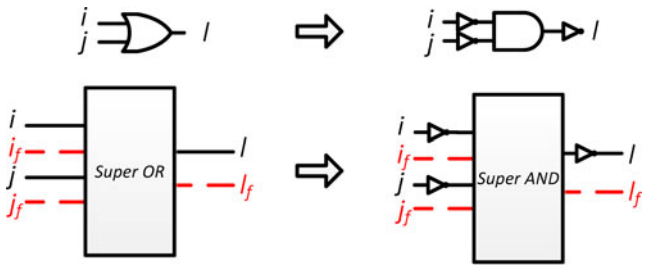


Fig. 7 Super-*AND* equivalent of Super-*OR* gate

4.5 Limited Depth Correlation Analysis

Due to the consideration of *pair-wise* signal correlations, the worst case complexity of CCM is quadratic $\binom{N}{2}$, i.e. $O(N^2)$, where N is the total number of gates. Although structural properties of the netlist can be explored to consider only correlated signals on reconvergent paths, its impact on scalability improvement is very limited. Therefore, to make our proposed method applicable to large circuits (e.g. more than 5000 gates), a heuristic algorithm is necessary to improve the scalability while maintaining satisfactory accuracy.

Inspired by the switching activity estimation work [9, 21], one heuristic algorithm - *limited depth correlation analysis* is proposed and integrated into our method to further improve its scalability. The basic idea is to consider spatial signal/error correlations within only limited logic depth d. Intuitively, starting from the gate with multiple fanouts, the farther the gates on different propagation paths are from the fanout point, the less correlated the corresponding output signals will be. The rational is that the signals outside the reconvergent regions will interact with these fanout signals and statistically *decouple* the corresponding gate outputs. We take one simple circuit in Fig. 8 to illustrate this phenomenon.

In this circuit there are three reconvergent regions: $\{I_1 \rightarrow Y_1 \rightarrow Z_1,\ I_1 \rightarrow X \rightarrow Z_1\}$; $\{I_1 \rightarrow Y_1 \rightarrow Y_2 \rightarrow Z_2,\ I_1 \rightarrow X \rightarrow Z_2\}$ and $\{I_1 \rightarrow Y_1 \rightarrow Y_2 \rightarrow Y_3 \rightarrow Z_3,\ I_1 \rightarrow$

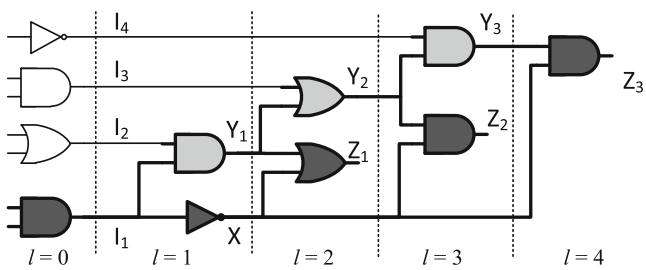


Fig. 8 Correlation reduction with signal/error propagation

$X \rightarrow Z_3\}$. Assume all the signals at level 0 are uncorrelated, $p(I_1) = p(I_2) = p(I_3) = p(I_4) = 0.5$, and additionally, $p(I_{1_f}) = 0.5$ (i.e. the error probabilities of signal I_1 is 0.5). From the circuit structure, the logic functions of concerned signals can be derived as follows:

$$X = \overline{I_1}, \qquad X_f = I_{1_f}$$
$$Y_1 = I_2 I_1, \qquad Y_{1_f} = I_2 I_{1_f}$$
$$Y_2 = Y_1 + I_3 = I_2 I_1 + I_3, \qquad Y_{2_f} = \overline{I_3} Y_{1_f} = \overline{I_3} I_2 I_{1_f}$$
$$Y_3 = Y_2 I_4 = I_2 I_1 I_4 + I_3 I_4, \qquad Y_{3_f} = I_4 Y_{2_f} = I_4 \overline{I_3} I_2 I_{1_f}$$

Base on these boolean functions and Eq. 3, we can derive the exact joint signal/error probabilities and PCCs of concerned signals at the reconvergent points Z_1, Z_2 and Z_3:

$$\rho_{I_1,I_1} = 1 \qquad \rho_{I_{1_f},I_{1_f}} = 1$$
$$\rho_{X,Y_1} = -0.5774 \qquad \rho_{X_f,Y_{1_f}} = 0.5774$$
$$\rho_{X,Y_2} = -0.2582 \qquad \rho_{X_f,Y_{2_f}} = 0.3780$$
$$\rho_{X,Y_3} = -0.1348 \qquad \rho_{X_f,Y_{3_f}} = 0.2582$$

The calculated values show that with the signal/error propagation, the highly correlated signals (i.e. PCC is 1 at the fanout point) become less and less correlated (i.e. PCC approaches 0). This trend holds not only for error-free signals, but also for erroneous signals. Therefore, using independent assumption for probability estimation of less correlated signals will cause smaller inaccuracy.

Based on this observation, the limited depth correlation heuristic is implemented and integrated into our method. As shown in Algorithm 1, the limited depth d is specified by the user, then for each gate g with multiple fanouts, a modified Depth First Search (DFS) procedure LIMITED_DEPTH_DFS (G, g, d) is carried out to identify the reconvergent regions within this depth. The gates in each reconvergent region will be recorded in one boolean correlation matrix m and their outputs will be considered as correlated signals in the later error estimation phase. Note that in our implementation of the function Depth_First_Search (G, g, d) in line 16, *INV* gate is excluded from the depth calculation, as it has no logical masking effect for error propagation at all.

This algorithm explores the structural properties of circuit netlist to identify the correlated gates with a single user-specified parameter: search depth d. Take Fig. 8 as example, the starting point of DFS is gate I_1 (fanout node). When $d = 1$, the DFS visiting sequence can be $I_1 \rightarrow X \rightarrow Z_1 \rightarrow Z_2 \rightarrow Z_3 \rightarrow Y_1$. Note *INV* gate X is considered as transparent with depth increase of zero. No gate is revisited and no reconvergent region is found, therefore signal pairs Y_1 and X will be considered as uncorrelated in error estimation. When $d = 2$, the DFS visiting sequence can become $I_1 \rightarrow X \rightarrow Z_1 \rightarrow Z_2 \rightarrow Z_3 \rightarrow Y_1 \rightarrow Z_1 \rightarrow Y_2$ and Z_1 is revisited, therefore reconvergent region is identified. In this case, signal pairs Y_1 and X will be considered as correlated when error probabilities should be calculated.

Algorithm 1 Limited Depth Correlation Analysis

```
1:  Input: Netlist G, search depth d, boolean correlation matrix m
2:  Correlated_Regions = empty set
3:  for each gate g in G do
4:      if fanout of g > 1 then
5:          tmp_Correlated_Regions = LIMITED_DEPTH_DFS(G, g,
    d);
6:          Add tmp_Correlated_Regions to Correlated_Regions
7:      end if
8:  end for
9:  for each region r in Correlated_Regions do
10:     for each two gates g1, g2 in r do
11:         m[g1, g2] = true;   ▷ These two gates should be considered
    as correlated in the error estimation
12:     end for
13: end for
14:
15: function LIMITED_DEPTH_DFS(G, g, d)
16:     revisting_gates = Depth_First_Search(G, g, d)          ▷ DFS
    procedure within depth d
17:     if number of revisting_gates > 0 then
18:         backtrack and generate reconvergent paths;
19:         return reconvergent regions;
20:     end if
21: end function
```

If not only the error probabilities but also the error correlations of primary output are of concern, we only need to add one *virtual* gate where all primary outputs converge, and Algorithm 1 can automatically take into account the necessary propagation paths to calculate primary output correlations.

Actually, the search depth d can either be specified by the user, or determined automatically by the algorithm. For example, this depth can be set to different value according to the number of fanout. The more fanout branches, the larger search depth will be used, therefore more reconvergent regions may be found. Again, such *variant* depth correlation search implies the flexibility of our proposed algorithm.

5 Extensions of the Proposed Method

As the proposed method addresses signal and error correlations explicitly in a unified way, it enables the extensions of this method to the following two directions.

5.1 Multiple Errors Propagation with Correlation

Our approach can be extended to account for more complex error models than single bit flip. As stated in [23, 31], multiple transient errors are no longer negligible in logic circuits.

To obtain accurate estimation, correlated error occurrence and the correlations in the propagation of errors originating from different error sites must be taken into account, rather than the independent assumption used in existing technique [13].

Our proposed approach needs only minor modification to address these correlations. Specifically, as error occurrence of impaired gates is no longer deterministic and independent, both error probabilities of the error sites and their correlation coefficients can be specified by the user as plug-in parameters. In the case of multiple errors modeling, there are multiple error cones and the *unified error cluster* must be considered which is the union of the gates in the fanout cones of all error sites. The gates within the unified error cluster are treated as super gates and those at the boundary as semi-super gates, as illustrated in Fig. 9. In this way, the same framework can be used to handle multiple error sites and their correlated propagation.

5.2 Block-level Error Estimation

Another advantage of the CCM method is that it explicitly calculates correlation coefficients from primary inputs to primary outputs. Therefore, it supports cascaded and hierarchical analysis of error propagation among different function blocks, which provides the possibility to raise the abstraction level from logic to register transfer level. However, two issues must be taken care of for this analysis.

First, it is essential to take into consideration the signal/error correlation of primary inputs. Actually, Algorithm 1 needs only small modifications to handle correlated inputs. For function LIMITED_DEPTH_DFS (G, g, d), the original single source DFS should be replaced with *multiple* source DFS, that is to say, the correlated primary inputs will be considered as the start points of DFS. In this way, both kinds of spatial correlations mentioned in Section 3 are covered by the proposed approach, since CCM intrinsically supports conveying signal statistics from primary outputs of one block to the primary inputs of its successors. To simplify the correlation calculation without loss of much accuracy, similar techniques as dynamic blocking of error propagation in Section 4.4 can be employed as well to treat primary outputs with low error probabilities as error-free ones.

Second, the reconvergency among multiple blocks needs to be handled as well. The technique in [16] provides one promising solution. First the signal probability of fanout node F is set to 0 or 1, i.e. this node has deterministic value 0 or 1, thus the statistical dependency of fanout signals can be eliminated. After the two deterministic cases are handled separately, the output error probabilities can be obtained by combining the two cases together according to the signal probability of node F. The scalability of this technique is questionable for netlists with large number of reconvergent regions. However, at block level the number of modules for the entire circuit is relatively small, therefore this technique is suitable regarding the complexity.

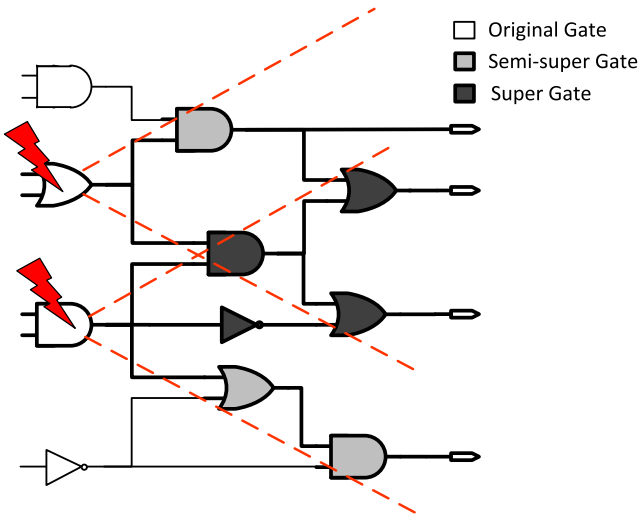


Fig. 9 Propagation paths with multiple error cones

6 Experimental Results

6.1 Experimental Setup

The proposed approach has been implemented in C++ using the igraph library [10], and experiments have been performed on a workstation with Intel E5540 2.53 GHz and 16 GB RAM.

The overall estimation and validation flow is illustrated in Fig. 10. The circuit is first synthesized with elementary gates, then the generated netlist is parsed and mapped to the corresponding graph structure. Both fault injection and the proposed Correlated Error Propagation (CEP) technique use this graph structure to evaluate the output error probabilities. After traversal of all error sites, the probabilities are compared to obtain the accuracy results.

In the experiments on benchmark circuits, the primary inputs of the circuits are assumed to be independent with signal probability 0.5. For the fault injection part, in the scenario of circuits with small number of primary inputs (<20), we use exhaustive simulation to obtain exact primary output error probabilities for all error sites, while for more primary inputs where exhaustive simulation is intractable, we turn to the statistical fault injection, based on Monte-Carlo simulation [32]. For the error probability of each primary output,

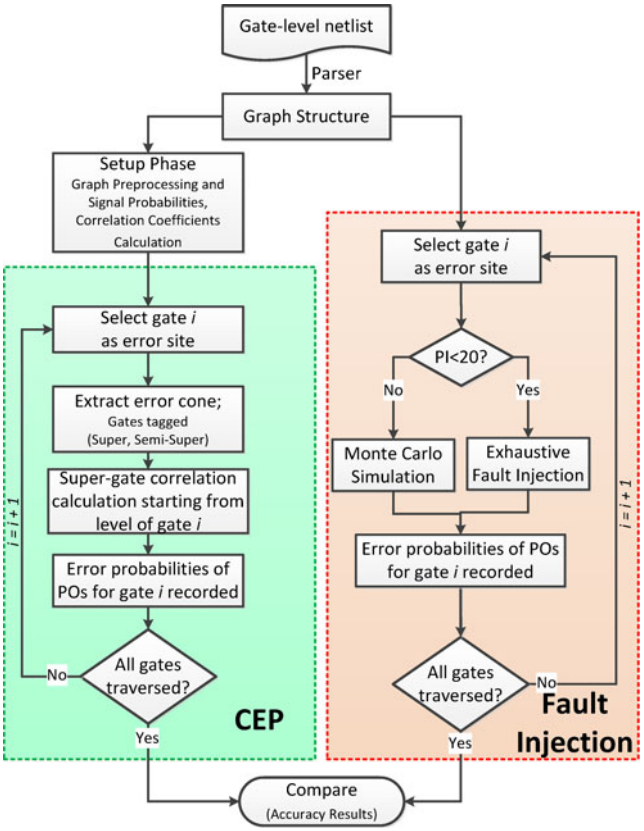


Fig. 10 Error probability estimation and validation flow

with 95% confidence level, the confidence interval (CI) is given by

$$\left(\hat{p_e} - 1.96\frac{S}{\sqrt{N}}, \hat{p_e} + 1.96\frac{S}{\sqrt{N}}\right)$$

where $\hat{p_e}$ is the estimated error probability of the primary output (i.e. mean estimation), S^2 is the variance estimation computed on the fly and N is the number of input vectors. Taking into consideration that the probability values are between 0.0 and 1.0, absolute width of the confidence interval

$$w_{CI} = 2 \times 1.96 \times \frac{S}{\sqrt{N}} \leq 0.005$$

is used as termination criteria.

In the proposed analytical method, graph preprocessing is carried out at setup phase and correlated gates within specified logic depth are identified. This step takes very short time because the search depth d is always very small (≤ 4). In this phase, error-free signal probabilities and correlation coefficients are also calculated to be used for later error estimation. Then for each error site, its error probability is set to 1.0 and its correlation coefficients with all the other signals at this level are set to 1.0 (independent error occurrence). After that the gates in the error cone are extracted and tagged as either super gates or semi-super gates. Starting from the logic level of error site, different formulas in Section 4.3 are used according to different gate types in the graph traversal. Finally, the error probabilities of primary outputs are obtained for later accuracy comparison.

Our proposed method has been validated for several ISCAS 85 benchmark circuits with regard to both accuracy and scalability. For the large combinational cores extracted from sequential circuit benchmarks ISCAS 89 and ITC 99, fault injection of each error site using simulation is not feasible. Therefore, only absolute runtime of our analytical method are reported to illustrate its scalability with different circuit sizes. In addition, to illustrate the necessity of error correlation modeling, we use the 32-bit ALU of OpenRISC 1200 processor [26] as a case study to investigate the correlated error probabilities at primary outputs. The realistic signal probabilities and correlation coefficients for primary inputs of this ALU are extracted from Value Change Dump (VCD) files of OpenRISC processor running Mibench benchmarks [15].

6.2 Benchmark Results

As our motivation is analyzing correlated error propagation from logic level to higher abstraction levels, individual error probabilities rather than an overall reliability metric are preferred. For each error site, the error probabilities of all primary outputs in the error cone are calculated first, and then they are compared one by one with fault injected Monte-Carlo simulation to obtain the accuracy results. For benchmarks with number of gate larger than 1000, the Monte-Carlo simulation takes very long time if *each* error site is simulated, so 100 gates are randomly selected for fault simulation. We report the maximum absolute inaccuracy (MAX) and average absolute inaccuracy (AVG), as defined in the following:

$$MAX_{inaccuracy} = \max_{1\leq n\leq N} \max_{1\leq i\leq O_n} |Pe_{CEP}(PO_i) - Pe_{MC}(PO_i)|$$

$$AVG_{inaccuracy} = \frac{1}{N \cdot O_n} \sum_{n=1}^{N} \sum_{i=1}^{O_n} |Pe_{CEP}(PO_i) - Pe_{MC}(PO_i)|$$

where N is the number of error sites evaluated in the benchmark circuit, O_n is the number of primary outputs in the error cone of error site n and $Pe(PO_i)$ is the error probability of primary output i. The relative inaccuracy is deliberately excluded due to its misleading meaning for small probability values.

To illustrate the tradeoff between accuracy and scalability discussed in Section 4.5, we also carried out the experiments with different correlation depth and the results are

shown in Fig. 11, in which *d=inf* means no depth limitation is considered. With regard to accuracy, for most of the benchmarks, the average inaccuracies are always reduced or almost the same when larger depth of correlation is considered. This means that with the logic depth increasing and correlation degree decreasing, the uncorrelation assumption causes less inaccuracy. At the same time, the speedup ratio of our analytical CEP approach with regard to Monte-Carlo simulation increases a lot, e.g. the average speedup ratio of *d=2* is approximately 2 orders of magnitude larger than that of $d = inf$.

However, for benchmark circuits *c432, c3540* and *c7552* the maximum inaccuracies for $d = inf$ are even larger than that for $d = 2$. After investigating the circuit structure, we find that it is due to the first-order approximation of CCM introduced in Section 3.1.2. After the propagation of errors on *multiple long* (e.g. fanout > 5) reconvergent paths, this first-order approximation results in larger inaccuracy compared to the case where the inputs of reconvergent gate are assumed to be independent. That is why the maximum inaccuracies of the benchmark *c432, c3540* and *c7552* with $d = inf$ are larger than that with $d = 2$. This interesting phenomenon illustrates that considering limited depth of correlation is beneficial not only to speedup our analytical method, but also to avoid inaccuracy accumulation of CCM for some special worst-case scenarios.

Table 1 shows the runtime and accuracy comparison between our proposed CEP ($d = 2$), Monte-Carlo simulation (MC) and the previously proposed 4-value error propagation probability (EPP) technique [1]. It is worth mentioning that our reported inaccuracy values are based on *node by node* comparison of the error probabilities of all concerned primary outputs for all error sites, rather than one overall accuracy metric for the whole circuit. Taking this into consideration, the maximum inaccuracies are still in reasonable range, as shown similarly in the probability estimation work [4, 21]. Regarding runtime, compared with Monte-Carlo simulation our proposed method with depth $d = 2$ exhibits different speedup ratios from 4 to 5 orders of magnitude, which reflect the complexity of CCM related to different circuit structures and additionally, the speedup potential of the proposed heuristics for various error cone structures.

Compared with 4-value EPP, our method has much smaller inaccuracy, less than half regarding both maximum and average inaccuracies. This is because the 4-value EPP considers only error correlations on reconvergent paths, while the signal correlations and cross correlations in Fig. 2 are all ignored. This simplification makes it one order

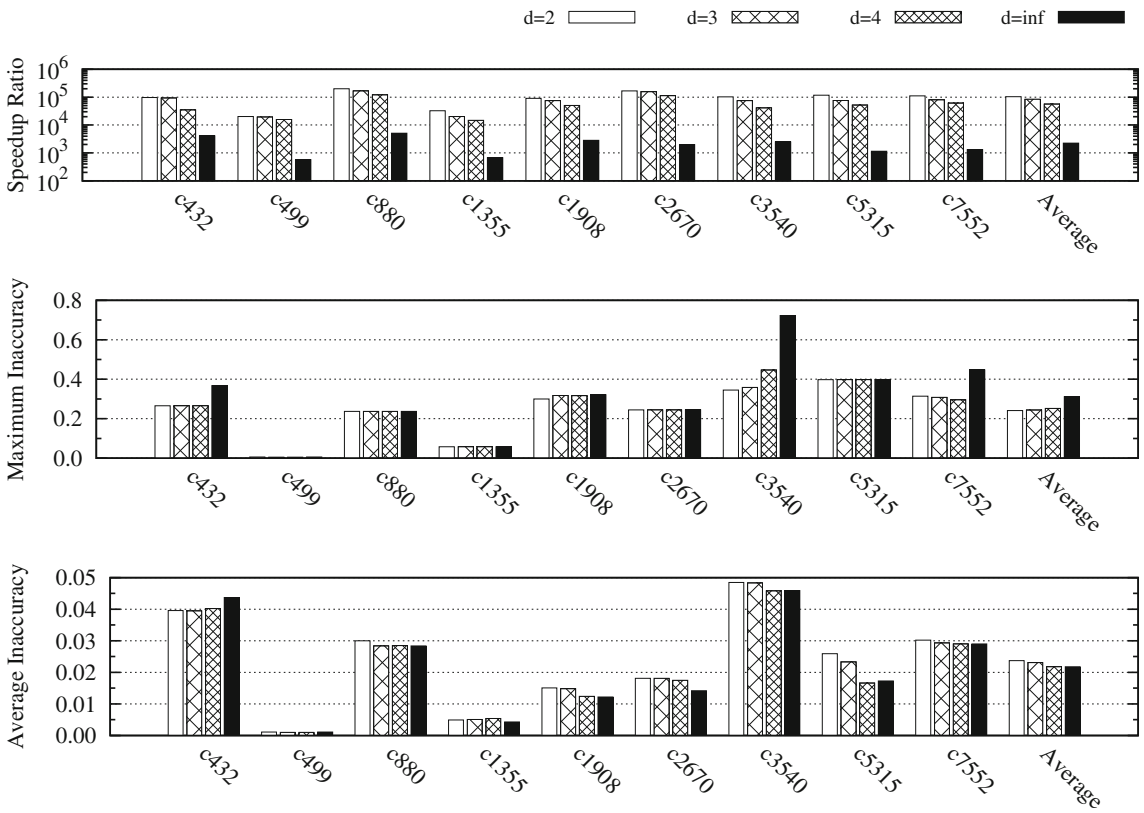


Fig. 11 Speedup, MAX and AVG inaccuracy for considering variant correlation depths

Table 1 Runtime(sec.) and accuracy of proposed CEP approach with correlation depth $d = 2$

Benchmark	# of Gates		MC runtime	4-value EPP[1] inaccuracy				CEP inaccuracy			
	Total gates	Error sites evaluated		MAX	AVG	Runtime	Speedup w.r.t CEP	MAX	AVG	Runtime	Speedup
c432	198	198	5412.58	0.3017	0.0314	0.0045	13.26	0.2658	0.0396	0.06	90210
c499	575	575	11112.80	0.5000	0.0316	0.0209	26.26	0.006	0.0011	0.55	20205
c880	459	459	30885.10	0.8251	0.0998	0.0070	22.93	0.2371	0.03	0.16	193032
c1355	588	588	17587.50	0.2736	0.0202	0.0234	23.13	0.0582	0.0049	0.54	32569
c1908	524	524	40892.90	0.6574	0.0409	0.0167	26.87	0.2996	0.0151	0.45	90873
c2670	834	834	106617.00	0.8250	0.0493	0.0156	40.96	0.2447	0.0181	0.64	166589
c3540	1088	100	15022.40	0.7223	0.0993	0.0035	42.14	0.3451	0.0485	0.15	102807
c5315	1666	100	21425.20	0.8725	0.0520	0.0027	58.51	0.3982	0.0258	0.16	133407
c7552	1999	100	28195.40	0.6996	0.0795	0.0030	68.28	0.3138	0.0301	0.21	136606
s38584	11818	11818	–	–	–	74.0401	1.42	–	–	105.15	–
s38417	12351	12351	–	–	–	81.0023	1.40	–	–	113.03	–
b20	38912	38912	–	–	–	163.4246	13.15	–	–	2149.73	–
b21	39168	39168	–	–	–	162.4681	12.64	–	–	2053.11	–
average	–	–	–	0.6308	0.0560	–	27.00	**0.2409**	**0.0237**	–	**107366**

of magnitude faster than our method (for fair comparison, the time to obtain signal probabilities, 4-value EPP using time-consuming logic simulation while CEP using analytical CCM, are excluded in the runtime reported in Table 1), but causes larger inaccuracy for the estimated probabilities, especially for the highly correlated circuit structures. In addition to the significant accuracy improvement, the main advantage of our method is that we are able to provide correlated error information but 4-value EPP can not.

6.3 Case Study of OpenRISC 1200 ALU

To illustrate the necessity of error correlation modeling, we investigated the ALU from OpenRISC processor as a case study. This ALU has 112 primary inputs including two 32-bit operands, one 32-bit multiply accumulation input, opcode, etc., and 38 primary outputs consisting of 32-bit computation results and some control signals. After synthesis there are 2854 elementary gates.

To investigate the error correlations of primary outputs under realistic workloads, we used the signal statistics extracted from VCD files as ALU primary inputs. The VCD files are dumped in the behavioral simulation of OpenRISC processor. Two typical applications - *StringSearch* and *BasicMath* from Mibench benchmarks [15] are selected to demonstrate the soft error statistics with workload dependencies. As shown in Fig. 12, for these two applications both the signal probabilities and signal correlations have significant difference. As *StringSearch* executes mostly logic comparison operations without arithmetic multiplication, all of primary input numbers $64 \sim 95$ have signal probabilities 0.0 and the corresponding PCCs are also 0.0, i.e. signals with constant logic values are independent from all the other ones.

Using the proposed analytical CEP, we evaluated all possible error sites, which, however, is intractable with Monte-Carlo simulation. The error probability $Pe_{AVG}(i)$ of each primary output i averaged on all possible error sites is obtained first. We also calculated the pairwise joint error probabilities and Pearson product-moment Correlation Coefficient (PCC) to explicitly demonstrate their correlation degree. At the same time, the most interesting metric *conditional error probabilities* for primary outputs are also reported, which can answer the following question:

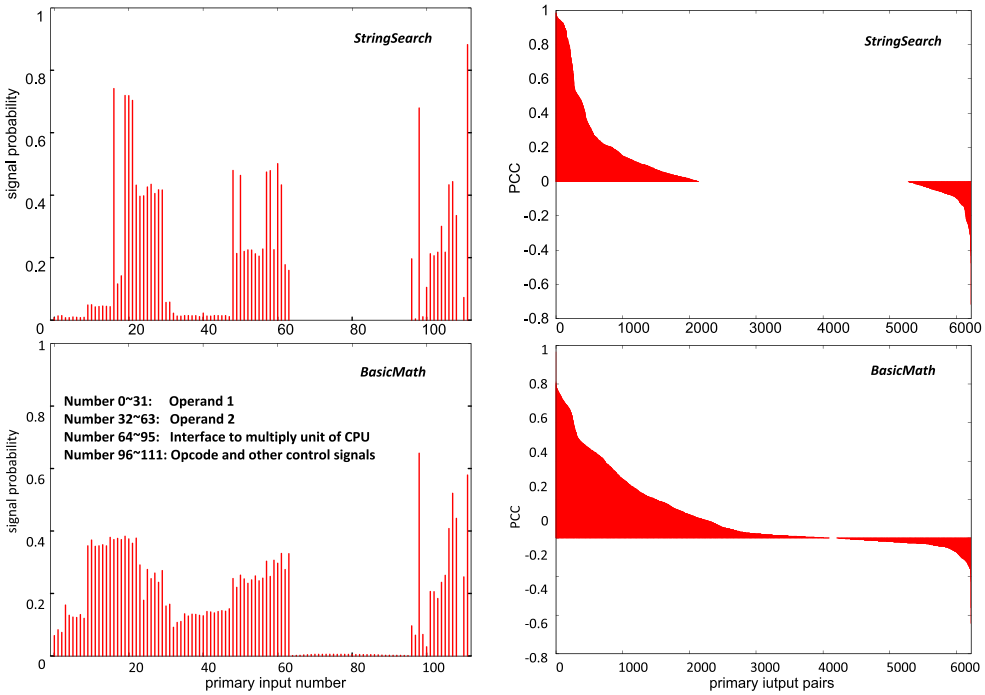
For soft errors occurring inside the functional block, given one primary output is erroneous, what is the probability that other outputs are erroneous at the same time?

The answer for this question is very useful for both high-level fault injection and efficient exploration of reliability enhancement.

The conditional error probability is defined as following:

$$Pe_{AVG}(i|j) = \frac{Pe_{AVG}(ij)}{Pe_{AVG}(j)} \tag{14}$$

Note that $Pe_{AVG}(i|j)$ is different from $Pe_{AVG}(j|i)$, so there are totally $38 \cdot 37 = 1406$ output pairs. Although it is possible to approximate second order correlations using

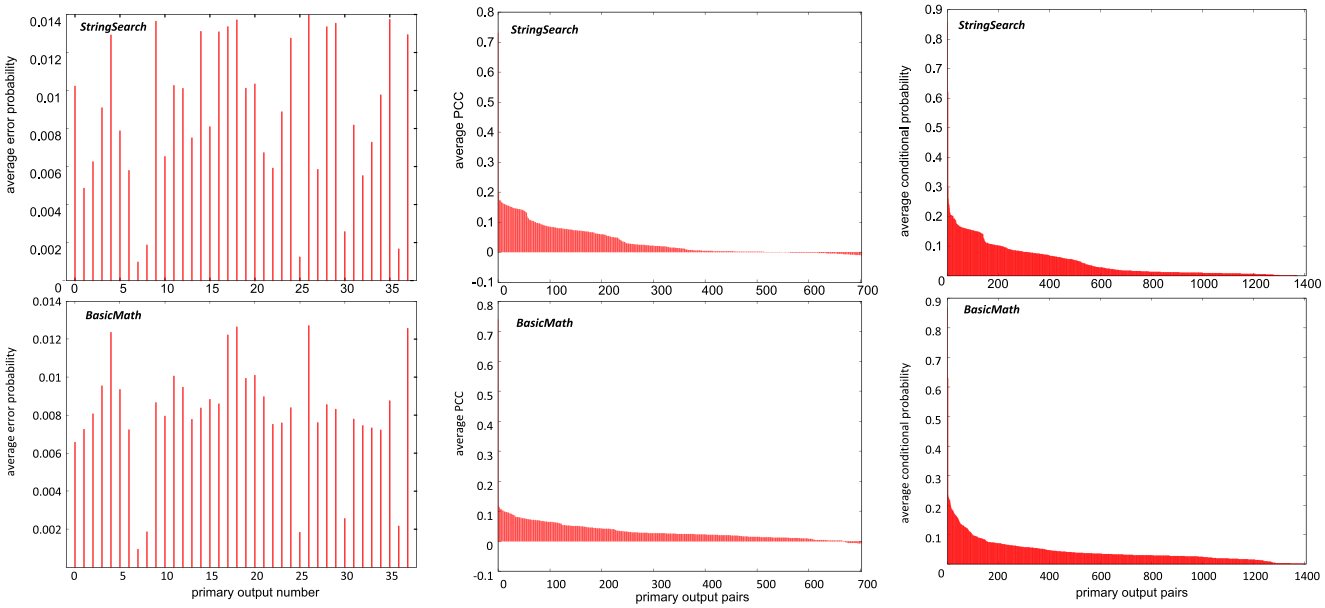


(a) Signal probabilities (b) Sorted pairwise PCCs

Fig. 12 Primary input statistics of OpenRISC ALU running applications *StringSearch* and *BasicMath*

Eq. 6, there will be $\binom{38}{2} \cdot (38-2) = 25308$ coefficients. For brevity it is sufficient to use pairwise correlations and conditional probabilities to illustrate the concept of *correlated bit flips*.

Figure 13a shows the average error probabilities of primary outputs of ALU running *StringSearch* and *BasicMath* applications. The average error probabilities of 32-bits computation results do not have large fluctuations, while the five outliers with particularly small error probabilities are listed in Table 2. From this table we can see that they are three control signals *cy_we, ov_we, flag_we* and two data signals *cyforw, ovforw*. Actually these three control signals become



(a) Error probabilities (b) Sorted pairwise PCCs (c) Sorted conditional error probabilities

Fig. 13 Error statistics for primary outputs of OpenRISC ALU running applications *StringSearch* and *BasicMath*

Table 2 Outliers of average error probabilities for primary outputs of OpenRISC 1200 ALU

StringSearch		BasicMath	
signal i	$Pe_{AVG}(i)$	signal i	$Pe_{AVG}(i)$
cy_we	0.0026	cy_we	0.0026
ov_we	0.0019	cyforw	0.0022
cyforw	0.0017	ov_we	0.0019
ovforw	0.0012	ovforw	0.0018
flag_we	0.0009	flag_we	0.0009

cy_we: carry write enable
ov_we: overflow write enable
cyforw: carry forward
ovforw: overflow forward
flag_we: comparison flag write enable

erroneous only when ALU executes wrong operations, and the average on all error sites including these executing operations on operands, makes the error probabilities of control outputs very small. For these two data signals, as the carry and overflow bits are generated in specific *ADDITION* operation and operands, the errors on these two signals are more probable to be masked, e.g. two erroneous operands are very likely to generate the same overflow bit '0' as two error-free operands, because *ADDITION* overflow occurs relatively rare. Due to this reason, the average error probabilities of these two data signals are also very small.

Figure 13b and c show that the PCCs and conditional error probabilities exhibit significant variations among all primary output pairs. The detailed experimental reports in Table 3 show that for both applications, the first two largest conditional error probabilities correspond to the same pair of outputs: carry write-enable signal and overflow write-enable signal, i.e. the carry write-enable signal has high probability to be erroneous given overflow write-enable signal is flipped (when the ALU executes the wrong operation due to internal soft errors), and vice visa. This observation is consistent with the functionality analysis of ALU, because normally these two signals are both '1' when the operation is *ADDITION*, otherwise they are both '0'. Our method can provide quantitative correlation values rather than the pure qualitative functional analysis.

Table 3 Top 10 average conditional error probabilities for primary outputs of OpenRISC 1200 ALU

StringSearch		BasicMath	
signal pair $i\|j$	$Pe_{AVG}(i\|j)$	signal pair $i\|j$	$Pe_{AVG}(i\|j)$
cy_we|ov_we	0.8572	cy_we|ov_we	0.8689
ov_we|cy_we	0.6229	ov_we|cy_we	0.6302
result[16]|ov_we	0.2958	result[0]|ov_we	0.2331
result[13]|ov_we	0.2598	result[3]|ov_we	0.2272
result[8]|ov_we	0.2470	result[2]|ov_we	0.2250
result[4]|ov_we	0.2407	result[7]|ov_we	0.2230
result[7]|ov_we	0.2392	result[1]|ov_we	0.2189
result[9]|ov_we	0.2288	result[8]|ov_we	0.2174
result[14]|ov_we	0.2183	result[5]|ov_we	0.2163
result[16]|cy_we	0.2150	result[4]|ov_we	0.2159

For the remaining eight large conditional error probabilities in Table 3, almost all of them are related to the overflow write enable signal. However, for the two applications most of the signals having high error probabilities given *ov_we* is erroneous are different. This significant difference is due to the workload dependencies as shown in Fig. 12. Different primary inputs statistics would generate different internal signal probabilities and correlations, therefore result in different masking effects in the error propagation.

In a word, our quantification of error correlations can not only identify highly correlated error signals due to the intrinsic functionalities, but also capture the influence of different workloads on error propagation and correlations. These quantified correlations and conditional probabilities can provide valuable insights and guides for the fault injection based simulation or analytical error estimation at higher abstraction levels.

7 Conclusion

Soft error is becoming one of the major reliability issues in nano era. Explicitly addressing the correlated bit-flips propagating from low level circuits is essential for accurate error estimation and error abstraction at high levels.

This paper proposed a novel approach to explicitly take into account both signal and error correlations in a unified way, therefore it can provide quantified error correlations. Based on the concept of error propagation function, a new super-gate representation was conceived to address error probability and correlation problems with signal probability and correlation techniques. Experimental results showed our approach is 5 orders of magnitude faster than Monte-Carlo simulation, while the average inaccuracy of error probability estimation is only 0.02. This method can be extended to model correlated error propagation due to multiple error sites at logic level as well as hierarchical and modular error analysis at higher levels.

Acknowledgments This work was partly supported by the German Research Foundation (DFG) as part of the national focal program "Dependable Embedded Systems". (SPP-1500, http://spp1500.ira.uka.de/)

References

1. Asadi H, Tahoori M (2006) Soft error derating computation in sequential circuits. In: IEEE/ACM international conference on computer-aided design, pp 497–501
2. Baraza J, Gracia J, Blanc S, Gil D, Gil P (2008) Enhancement of fault injection techniques based on the modification of VHDL code. IEEE Trans Very Large Scale Integration Systems 16(6):693–706
3. Baumann R (2005) Soft errors in advanced computer systems. IEEE Design and Test of Computers 22(3):258–266
4. Bhanja S, Ranganathan N (2004) Cascaded bayesian inferencing for switching activity estimation with correlated inputs. IEEE Tran Very Large Scale Integration Systems 12(12):1360–1370
5. Borkar S (2005) Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. IEEE Micro 25(6):10–16
6. Chen L, Tahoori M (2012a) An efficient probability framework for error propagation and correlation estimation. In: IEEE international on-line testing symposium Barcelona, Spain, pp 170–175
7. Chen L, Tahoori M (2012b) Soft error propagation and correlation estimation in combinational network. In: The first workshop on manufacturable and dependable multicore architectures at nanoscale (MEDIAN'12), Annecy, France
8. Choudhury M, Mohanram K (2009) Reliability analysis of logic circuits. IEEE Tran on Computer-Aided Design of Integrated Circuits and Systems 28(3):392–405
9. Costa JC, Silveira LM, Devadas S, Monteiro J (2004) Power estimation using probability polynomials. Design Automation for Embedded Systems, pp 19–52
10. Csárdi G, Nepusz T (2006) The igraph software package for complex network research. InterJournal Complex Systems 1695
11. Entrena L, Garcia-Valderas M, Fernandez-Cardenal R, Lindoso A, Portela M, Lopez-Ongil C (2012) Soft error sensitivity evaluation of microprocessors by multilevel emulation-based fault injection. IEEE Tran on Computers 61(3):313–322
12. Ercolani S, Favalli M, Damiani M, Olivo P, Ricco B (1989) Estimate of signal probability in combinational logic networks. In: European test conference, pp 132–138
13. Fazeli M, Ahmadian S, Miremadi S, Asadi H, Tahoori M (2011) Soft error rate estimation of digital circuits in the presence of Multiple Event Transients (METs). In: Proceedings of design, automation and test in Europe, pp 1–6
14. Gill B, Seifert N, Zia V (2009) Comparison of alpha-particle and neutron-induced combinational and sequential logic error rates at the 32nm technology node. In: IEEE international reliability physics symposium, pp 199–205
15. Guthaus MR, Ringenberg JS, Ernst D, Austin TM, Mudge T, Brown RB (2001) Mibench: a free, commercially representative embedded benchmark suite. In: IEEE international workshop on workload characterization, pp 3–14
16. Han J, Chen H, Boykin E, Fortes J (2011) Reliability evaluation of logic circuits using probabilistic gate models. Microelectronics Reliability 2:468–476
17. Kretzschmar U, Astarloa A, Lazaro J, Jimenez J, Zuloaga A (2011) An automatic experimental set-up for robustness analysis of designs implemented on SRAM FPGAs. In: International symposium on system on chip (SoC), pp 96–101
18. Krishnaswamy S, Viamontes G, Markov I, Hayes J (2005) Accurate reliability evaluation and enhancement via probabilistic transfer matrices. In: Proceedings of design, automation and test in Europe, pp 282–287
19. Li X, Adve SV, Bose P, Rivers JA (2005) SoftArch: an architecture-level tool for modeling and analyzing soft errors. In: International conference dependable systems and networks, pp 496–505
20. Mahatme N, Jagannathan S, Loveless T, Massengill L, Bhuva B, Wen SJ, Wong R (2011) Comparison of combinational and sequential error rates for a deep submicron process. IEEE Tran on Nuclear Science 58(6):2719–2725
21. Marculescu R, Marculescu D, Pedram M (1998) Probabilistic modeling of dependencies during switching activity analysis. IEEE Tran on Computer-Aided Design of Integrated Circuits and Systems 17(2):73–83
22. Miskov-Zivanov N, Marculescu D (2006) Circuit reliability analysis using symbolic techniques. IEEE Trans Comput-Aided Des Integr Circ Syst 25(12):2638–2649
23. Miskov-Zivanov N, Marculescu D (2010) Multiple transient faults in combinational and sequential circuits: a systematic approach. IEEE Tran on Computer-Aided Design of IntegratedCircuits and Systems 29(10): 1614–1627
24. Mohyuddin N, Pakbaznia E, Pedram M (2008) Probabilistic error propagation in logic circuits using the Boolean difference calculus. In: IEEE International conference on computer design, pp 7–13
25. Mukherjee S, Emer J, Reinhardt S (2005) The soft error problem: an architectural perspective. In: International symposium on high-performance computer architecture, pp 243–247
26. OpenRISC Project (2012) http://opencores.org/openrisc
27. Papoulis A (1990) Probability & statistics. Prentice Hall, Englewood Cliffs
28. Rajaraman R, Kim JS, Vijaykrishnan N, Xie Y, Irwin MJ (2006) SEAT-LA: a soft error analysis tool for combinational logic. In: International conference on VLSI design, pp 499–502
29. Rao R, Chopra K, Blaauw D, Sylvester D (2006) An efficient static algorithm for computing the soft error rates of combinational circuits. In: Proceedings of design, automation and test in Europe, pp 1–6
30. Rejimon T, Lingasubramanian K, Bhanja S (2009) Probabilistic error modeling for nano-domain logic circuits. IEEE Tran on Very Large Scale Integration Systems 17(1):55–65
31. Rossi D, Omana M, Toma F, Metra C (2005) Multiple transient faults in logic: an issue for next generation ics?. In: IEEE international symposium on defect and fault tolerance in VLSI systems, pp 352–360
32. Rubinstein R, Kroese D (2008) Simulation and the Monte Carlo method. Wiley, New York
33. Sanda P, Kellington J, Kudva P, Kalla R, McBeth R, Ackaret J, Lockwood R, Schumann J, Jones C (2008) Soft-error resilience of the IBM POWER6 processor. IBM Journal of Research and Development 52(3): 275–284
34. Shivakumar P, Kistler M, Keckler S, Burger D, Alvisi L (2002) Modeling the effect of technology trends on the soft error rate of combinational logic. In: International conference on dependable systems and networks, pp 389–398
35. Sivaswamy S, Bazargan K, Riedel M (2009) Estimation and optimization of reliability of noisy digital circuits. In: International symposium on quality of electronic design, pp 213–219
36. Wang N, Quek J, Rafacz T, Patel S (2004) Characterizing the effects of transient faults on a high-performance processor pipeline. In: International conference on dependable systems and networks, pp 61–70
37. Yu C, Hayes J (2011) Trigonometric method to handle realistic error probabilities in logic circuits. In: Proceedings of design, automation and test in Europe, pp 1–6

38. Zhang B, Wang W, Orshansky M (2006) FASER: fast analysis of soft error susceptibility for cell-based designs. In: International symposium on quality electronic design
39. Zhang M, Shanbhag N (2006) Soft-Error-Rate-Analysis (SERA) methodology. IEEE Tran on Computer-Aided Design of Integrated Circuits and Systems 25(10):2140–2155

Liang Chen received his Bachelor degree of Electronic Science and Technology at Xi'an Jiaotong University, China, in 2007. After one year Master study of Microelectronics, in July 2008 he took part in the International Graduate Program: Master of Science in Communications Engineering (MSCE) at the Technische Universitt Mnchen, Germany. After graduation he started working as a PhD student in the CDNC group of Prof. Tahoori at Karlsruhe Institute of Technology since December 2010. His current focus is logic-level soft error rate estimation and high-level soft error analysis.

Mojtaba Ebrahimi received the B.Sc. degree from Shahed University, Tehran, Iran, and the M.Sc. degree from Sharif University, Tehran, in 2008 and 2010, respectively, both in computer engineering. He is currently pursuing the Ph.D. degree with the Chair of Dependable and Nano Computing, Karlsruhe Institute of Technology, Karlsruhe, Germany. He was a Research Assistant with the Dependable System Laboratory, Sharif University, from 2010 to 2011.

Mehdi B. Tahoori received his Ph.D. and M.S. in Electrical Engineering from Stanford University in 2003 and 2002, respectively, and B.S. in Computer Engineering from Sharif University of Technology, Tehran, Iran in 2000. He is a full professor and Chair of Dependable Nano-Computing (CDNC) at the Institute of Computer Science and Engineering (ITEC), Department of Computer Science, Karlsruhe Institute of Technology (KIT), Germany. In 2003, he joined the Electrical and Computer Engineering Department at the Northeastern University as an assistant professor where he promoted to the rank of associate professor with tenure in 2009. During 2002–2003, he was a research scientist at Fujitsu Laboratories of America in advanced CAD research working on reliability issues in deep-submicron mixed-signal VLSI designs. In addition to five pending and granted U.S. and international patents for his work, Dr. Tahoori has over 140 publications in major journals and conference proceedings on wide-ranging topics from dependable computing and emerging nanotechnologies to system biology. Dr. Tahoori has served as the program committee member as well as workshop, panel and special session organizer of various conferences and symposia in the areas of VLSI test, reliability, and emerging nanotechnologies, such as ITC, ICCAD, DATE, ETS, ICCD, ASP-DAC, GLSVLSI, and VLSI Design. He is an associate editor of ACM journal of Emerging Technologies for Computing and Chair of ACM SIGDA Technical Committee on Test and Reliability. He was a recipient of National Science Foundation (NSF) Early Faculty Development (CAREER) award. Prof. Tahooris research interests include nano computing, reliable computing, VLSI testing, reconfigurable computing, emerging nanotechnologies, and system biology.