

Modeling and Optimization for Soft-Error Reliability of Sequential Circuits

Natasa Miskov-Zivanov, *Student Member, IEEE*, Diana Marculescu, *Member, IEEE*

Abstract – Due to reduction in device feature size and supply voltage, the sensitivity to radiation induced transient faults of digital systems increases dramatically. In this paper, we present two approaches to evaluating the susceptibility of sequential circuits to soft errors. The first approach uses Markov chain theory, but can only provide steady-state behavior information. The second approach uses symbolic modeling based on BDDs/ADDs and circuit unrolling. The *SER* evaluation using this approach is demonstrated by the set of experimental results, which show that, for most of the benchmarks used, the *SER* decreases well below a given threshold (10^{-7} FIT) within ten clock cycles after the hit. The results obtained with the proposed symbolic framework are within 4% average error and up to 11000X faster when compared to HSPICE detailed circuit simulation. The framework can be used for selective gate sizing targeting radiation hardening leading up to 80% *SER* reduction when applied to a subset of *ISCAS'89* benchmarks.

Index Terms – combinational logic circuits, reliability, sequential logic circuits, symbolic manipulation.

I. INTRODUCTION

ONCE regarded as a concern only for space applications, transient faults caused by radiation are becoming a major barrier to robust system design manufactured at advanced technology nodes like 90nm or smaller. The high data-integrity and reliability requirements make these faults an extremely important design aspect for microprocessors or other commodity components. Therefore, the protection from radiation induced transient faults has become as important as other product characteristics such as performance or power consumption [10].

A radiation-induced charged particle passing through a microelectronic device ionizes the material along its path. The free carriers that are created around the particle track can be affected (attracted/rejected) by an internal electric field of the device and result in an electrical pulse, *single-event transient* (SET), large enough to disrupt normal device operation. This disruption is not associated with any permanent damage to the device and is thus called a *soft error* or a *single-event upset*

(SEU). The effect of soft errors is measured by the *soft error rate* (*SER*) in FITs (*failure-in-time*), which is defined as one failure in 10^9 hours.

Traditionally, memory elements have been much more sensitive to soft errors than combinational logic circuits. Three factors prevented logic from becoming more susceptible to soft errors:

- *logical masking* – to be latched, a SET has to propagate on a sensitized path from the location where it originates to a latch;
- *electrical masking* – due to the electrical properties of the gates the glitch is passing through, it can be attenuated or even completely masked before it reaches the latch;
- *latching-window masking* – the glitch will be latched only if it reaches the latch in time by satisfying setup and hold time conditions.

Technology scaling decreases the impact of the three masking factors on radiation-induced SET. The reduction in feature sizes and supply voltages allows lower energy particles to result in SET. Reduced logic depth and smaller gate delays decrease attenuation when the glitch propagates through the circuit. Finally, the increase in clock frequency decreases latching-window masking. Thus, *SER* in combinational logic is increasing with every technology node and is expected to become an issue beyond 90nm technology node. Moreover, once a SET can propagate freely through the combinational circuit, sequential circuits will become very sensitive to such events [3]. This is due to the fact that, once latched, soft errors can propagate through the sequential circuit in subsequent clock cycles and thus affect the outputs of the circuit more than once.

When an estimated *SER* for a given product is higher than a given threshold, mitigation techniques need to be considered. The most obvious way to eliminate soft errors would be to get rid of the radiation sources that cause them. The solution for the remaining *SER* would be to make different process and technology choices. Furthermore, radiation sensitivity can be reduced significantly by design and layout changes. Any change, which increases critical charge while maintaining or reducing collected charge, will improve the *SER* of a device.

In this work, we estimate the likelihood that a SET in a sequential circuit will lead to errors in clock cycles following the particle hit. Our main goal is to allow for *symbolic modeling* and *efficient estimation* of the susceptibility of a sequential circuit to soft errors. We apply the model proposed in this work to find the gates that have the highest soft error impact, that is, the gates that contribute the most to the soft error failure rate of the logic circuit. We use this information

Manuscript received May 15, 2007, revised September 20, 2007, accepted November 6, 2007. This work was supported in part by National Science Foundation Grant CCF-0542644.

Copyright © 2007 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

Natasa Miskov-Zivanov is with the Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213 USA (phone: 412-268-6648; e-mail: nmiskov@ece.cmu.edu).

Diana Marculescu is with the Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: dianam@ece.cmu.edu).

for selective gate resizing in order to significantly harden the circuit with a reasonable area overhead.

The rest of this paper is organized as follows. In Section II we give an overview of related work and outline the contribution of our work. In Section III we briefly review the sequential circuit preliminaries. Section IV presents the application of Markov chain theory on steady-state *SER* analysis. In Section V, we describe in more detail our methodology for determining sequential circuit susceptibility to soft errors. Section VI presents in more detail the proposed radiation hardening approach. In Section VII, we report experimental results for a set of common benchmarks. Finally, with Section VIII we conclude our work.

II. RELATED WORK

Intensive research has been done so far in the area of analysis of transient faults in both combinational and sequential circuits [1]-[5], [8]-[11], [13]-[15], [20]-[24]. One obvious approach is to inject the fault into the given node of the circuit and simulate the circuit for different input vectors in order to find whether the fault propagates [24]. However, this approach becomes intractable for larger circuits and larger number of inputs and thus gives way to approximate approaches that use analytical and symbolic methods to evaluate circuit susceptibility to soft errors. In this section, we describe these methods that were used to find the susceptibility to soft errors of combinational and sequential circuits. We also briefly outline the contributions of our work and compare it to previous work.

A. *SER* in combinational circuits

A number of methods have been proposed recently to evaluate the susceptibility of combinational logic circuits to soft errors. In [11],[21], the authors separate the analysis of the three masking factors and include different heuristics to speed up the evaluation of the soft error susceptibility. Noise rejection curves or HSPICE simulation of inverter chains is used to evaluate the effect of electrical masking while path tracing or logic simulation is used to find the probability of logical masking. These methods may not reflect the impact of the internal node location and attenuation on the observability of the glitch at the latched output and can also become very inefficient for larger circuits.

Recently, several *symbolic models* have been developed to estimate the susceptibility of logic circuits to soft errors. The authors of [20] use Binary Decision Diagrams (BDDs) to represent sensitized path information, as well as upset events. However, their approach appears to rely on explicit enumeration of BDDs corresponding to all input conditions and assumes simple superposition of reconvergent glitches, without considering their possible mutual masking. Furthermore, since it doesn't rely on using Algebraic Decision Diagrams (ADDs), the approach in [20] cannot model arbitrary input distributions, which can be handled with ADDs via Dynamic Markov Models [7].

The approach proposed in [9] uses both BDDs and ADDs to allow for a unified treatment of logical, electrical and latching-

window masking effects and has been shown to be very efficient. When compared to another recent work [13] that also includes all three masking factors, the method proposed in [9] computes the *SER* much faster, while being more accurate. For example, one run of the algorithm in [13] assumes one specific input vector, and thus applies Monte Carlo analysis leading to an average error of 16%, while the work presented in [9] accounts for all possible input vectors in one run by using BDDs and ADDs with the average error of 4%. The approach proposed in [9] is incorporated into our analysis of sequential circuits and its main aspects will be discussed in Section III.B.

B. *SER* in sequential circuits

Compared to the number of methods proposed for modeling soft error susceptibility of combinational circuits, sequential circuits have received less attention. Most of the previous work in evaluating *SER* in sequential circuits has been done using simulation. Approaches to this problem addressed different levels of abstraction, starting with the device level and circuit level, up to the system level. At the device level, the most commonly used methods were the simulation based on drift-diffusion model or Monte Carlo simulation. Although very accurate, the drawback of methods used at this level of abstraction is very high computational complexity. Circuit and logic simulation are the next levels of abstraction at which the effects of soft errors in circuits can be evaluated, but it again requires large computational time, which rapidly increases with the size of the circuit [5]. On the other hand, simulations that can be used at the architectural level are much faster than in previous methods that use fault injection, but are usually applicable to very specific designs and are not general enough. As in the case of lower levels of abstraction, these simulations will also depend on chosen stimuli.

Similar to combinational circuits, the alternative to simulation is analytical/symbolic modeling. However, while in case of logic circuits one pass through the circuit is enough to evaluate its susceptibility to given particle hit, in the case of sequential circuits this evaluation becomes much more difficult. Since sequential circuits have a feedback loop leading back to the state inputs of the circuit, it is possible that errors latched at state lines propagate through the circuit more than once. Thus, the effect of a single particle hit can affect outputs during several clock cycles. To consider this effect, the analysis of the propagation of an SET through sequential circuit in more than one clock cycle is necessary. In the worst case, this analysis and evaluation would have to consider an infinitely large number of cycles. Therefore, to be able to model and analyze sequential circuit susceptibility to soft errors, we need approximate methods.

Although there has been a lot of work in the area of modeling the probabilistic behavior of finite state machines (FSMs) [6], [7], the main goal of those methods was calculating steady-state behavior of the circuit, which can be applied, for example, in estimating the switching activity of the circuit for the purpose of power evaluation. In the case of soft errors, the transient behavior of the circuit is more

important, that is: (i) the time the circuit spends transitioning through erroneous states until it reaches a steady-state behavior; and (ii) the effect this transitioning has on the outputs, that is, the susceptibility to soft errors of the target sequential circuit.

One method that evaluates the probability of latching the error in sequential circuit in the cycles following the particle hit was recently proposed by Asadi *et al.* [1]. In that work, the authors assume hits can happen at state flip-flops only and then, based on this information, find the error probability at each output due to each individual flip flop hit. This analysis excludes cases where internal gates of circuit's combinational logic are hit. In their case, the error at the output of combinational logic due to a state line error is found using a framework that analyzes logical masking only. Such an approach does not hold for the case of internal gate hits when electrical and latching-window masking need to be included as well. Thus, the approach proposed in [1] will not give accurate results if hits are assumed to happen not only at flip-flops, but also inside the combinational logic part of sequential circuit. Furthermore, the logical masking method that is used in [1] relies on path tracing, which is much slower than our proposed approach. Finally, the authors report their results in terms of the mean time to manifest error (MTTM) metric, and not in terms of *SER*, which is the most common metric for measuring the soft error susceptibility of circuits.

C. Soft error hardening

Soft error mitigation techniques can be classified into three distinct categories [5]: device-, circuit- and system-level techniques. *Device-level* hardening approaches mainly aim to reduce and mitigate the effects of charge collection at the site of the particle strike and require fundamental changes in the manufacturing process [3]. *Circuit-level* techniques rely on changes in the circuit design for hardening memories, combinational or sequential circuits. *System-level* techniques deal with soft errors at the system architecture level and usually involve the introduction of redundancy into the design.

One cost-effective approach that uses a fundamental method to harden the circuit against soft errors was proposed by Zhou *et al.* in [22] and further described in [23]. The authors of [22] propose selective gate resizing, that is, the increase in critical charge of the gates that have the largest impact on the soft error susceptibility of the *combinational circuit* through scaling of the transistors in those gates. In this work, we incorporate a similar hardening method into our framework, but for *sequential circuits* (not only combinational circuits). Thus, we describe the gate resizing in more detail in Section VI. There are, however, two important differences between the work presented in [22],[23] and our work. First, the work in [22],[23] is primarily focused on proposing gate resizing as a viable soft error mitigation technique, while our work presents this technique only to show how circuit hardening techniques can be incorporated into the proposed modeling and analysis framework. Second, the authors in [22],[23] assume the same glitch propagation modeling methodology, which was previously described in [11] and is different from what we use

in our work. More precisely, the impact of three masking factors is modeled and computed separately in [11], while our approach applies important (as we will describe here in Section II.D) unified modeling of these masking factors. There are few more recent approaches that also apply gate resizing as a soft error mitigation technique [2],[14]. In our framework, we included analysis of the impact of **different glitch sizes** on the *SER* before and after gate resizing, while the work presented in [2] provides no observation of glitch size or collected charge. Next, the authors in [2] use their previous model (described in [1]) for computing the *SER*, and that model includes logical masking only, without considering electrical or latching-window masking. The work presented in [14] uses the tool previously proposed in [13] for soft error susceptibility computation of combinational circuits. An important difference between the work in [14] and our symbolic modeling methodology is the **scalability**, which stems from these different circuit error susceptibility evaluation methods (as already described in Section II.A).

D. Paper contribution

In order to estimate probability of soft errors in sequential circuits, we use the unrolling method described in detail in Section V. As opposed to Markovian analysis approaches that allow only steady-state analysis, this method allows for *both transient and steady-state* evaluation of the propagation of SET and the soft error susceptibility of sequential circuits. As already described, a number of methods have been proposed recently for the evaluation of the soft error susceptibility of combinational circuits. From among those, we chose to use the symbolic modeling framework presented in [9] that relies on BDDs and ADDs.

1) Unified symbolic treatment

The framework proposed in [9] for soft error susceptibility evaluation of combinational circuits was chosen as the basis for sequential circuit analysis due to the fact that it provides a unified *treatment* of the three masking factors: *logical*, *electrical* and *latching-window masking*. More precisely, by using BDDs and ADDs, the information about the masking factors is implicitly generated inside the decision diagrams, therefore including their *joint* dependency on input patterns

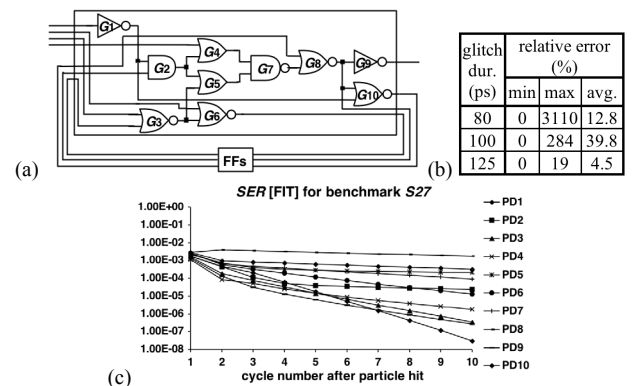


Fig. 1. Example: (a) circuit S27, (b) results for separate and unified treatment of masking factors for three initial glitch durations (80ps 100ps and 125ps) and (c) *SER* changes during several clock cycles for different input probability distributions.

and circuit topology. This allows for efficient concurrent computation of output error susceptibility due to hits on various internal nodes.

The *unified* treatment of three masking factors is important as it can be seen from the example in Fig. 1 (a) and (b). We consider separately the effect of logical masking, on one hand, and the effect of electrical and latching-window masking, on the other hand, for the *ISCAS'89* benchmark *S27* (Fig. 1 (a)). There are two cases of reconvergent paths in circuit *S27*. From gate G_2 there are two paths that reconverge at gate G_7 , and thus affect the probability of error propagation to the output of the circuit and two next-state lines. From gate G_1 , there is one path leading directly to gate G_6 and one that goes through gate G_2 creating overall three possible reconvergent paths to one of the next-state lines and two reconvergent paths to the output and another next-state line. The separate computation of different masking factors will incur an error, since it sums separately (i) probabilities of sensitization of all reconvergent paths, and (ii) probabilities of latching on all reconvergent paths; and then it multiplies the two terms. This will not take into account the relative arrival time and durations of the glitches at the reconvergence point.

The results shown in the table in Fig. 1 (b) represent minimum, maximum and average relative error of the model that evaluates electrical, latching-window and logical masking separately, compared to the unified model averaged across ten different input vector probability distributions, for three different initial glitch durations. As it can be seen from these results, multiplying the probability of logical masking with the probability of electrical and latching-window masking that were computed separately leads to the error in the probability of latching the glitch which can be as large as 3100%. However, for smaller glitch duration (80ps), the average error is not very large, due to the fact that most glitches are masked, and separate and unified methods give similar results. For the case of large initial glitches (125ps), all glitches propagate, and the only difference between the two methods comes from the way reconvergent paths are handled.

2) Exact and approximate methods for SER estimation in sequential circuits.

To take into account the joint effect of logical, electrical and latching-window masking and, at the same time, to allow for the *efficient* estimation of the effects in time of SET on the outputs of the sequential circuit, we rely on two proposed methods for exact and approximate evaluation of *SER* in sequential circuits, as described in detail in Sections IV and V. The *exact* method relies on Markov Chain (MC) analysis-based *SER* estimation, which is able to provide *steady-state SER estimates* following a hit. To cope with potential state explosion/complexity problems associated with this type of analysis and to allow for modeling of *transient effects in SER evaluation*, we also propose a low-cost, *approximate* method based on *circuit unrolling*.

For a better understanding of the methodology proposed in this work, we show in Fig. 1 (c) the results obtained using our approximate method for the example circuit *S27* for several input vector probability distributions (PD). The results

presented in Fig. 1 (c) describe the effect of a particle hit on circuit behavior, that is, the output error probability variation in time. As it can be seen, in most cases *SER* converges to very low values, except for a few cases in which it stays almost constant. This shows that *SER* transient behavior is heavily dependent on the input distribution, and thus classic MC analysis may not be appropriate for capturing it. Our framework is not only scalable, but also accurate when compared to detailed circuit simulation. As shown in Section VII.B, the proposed framework is within **4%** accurate when compared to HSPICE, at an **11000X** speedup.

3) Gate resizing for sequential circuit hardening.

Once soft error impact of individual gates is known, we can determine sensitive areas of the chip and therefore apply specific radiation hardening techniques. As already mentioned, in this paper we focus on circuit-level hardening technique that resizes selected gates such that the critical charge needed to change the output of a gate is increased.

We determine the mean error impact (*MEI*) of a gate by averaging its error impact across all outputs and all probability distributions. All gates with *MEI* larger than a given threshold are resized, such that the outputs of those gates are not affected when hit by particles with energies in a given interval. As criteria for choosing gates to be resized, we use: (i) the *MEI* of a gate averaged across all cycles under consideration in the target circuit, and (ii) the *MEI* of a gate determined only during the cycle when hit happens.

In Fig. 2, we show how the *SER* changes in the cycles following the particle hit, before and after gate resizing, for a set of benchmark circuits, when the second criterion is used. The results on both curves in Fig. 2, “original” and “resized” are presented as a percentage of the original *SER* value during the first cycle. It can be seen from Fig. 2 that the *SER* decreases rapidly after the first cycle both before and after resizing. Furthermore, after resizing, the *SER* improves even during the first cycle for as much as 83%.

As it will be seen from the results presented in Section VII, using the first criterion leads to a smaller number of resized gates (and potentially less area overhead). This is due to the fact that *MEI*, as well as *SER*, most often decreases when propagating through the unrolled circuit, while, usually, it is

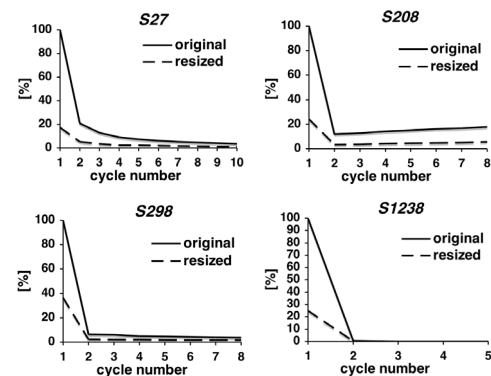


Fig. 2. *SER* changes in the cycles following the particle hit, compared to the *SER* during the first cycle, before and after gate resizing, for four benchmark circuits.

largest during the first cycle (when the hit occurs). Thus, this once more supports the fact that: (i) considering only the combinational logic effects during the cycle when hit occurs is not sufficient for *SER* analysis; and (ii) time-dependent analysis is necessary in sequential circuits as opposed to just steady-state analysis which cannot give any insight into the transient behavior of the circuit.

III. SEQUENTIAL CIRCUITS - PRELIMINARIES

A typical sequential circuit consists of combinational logic and flip-flops (FFs). The inputs to the combinational logic are the primary inputs and the outputs of FFs, while the outputs of combinational logic are the primary outputs and inputs of the FFs.

When a charged particle hits the circuit, there are two possible cases:

1. The particle hits an internal gate of the combinational logic
2. The particle hits the flip-flop that stores information about the next state.

In the case when the particle hits an internal gate, the analysis of the propagation of single-event transient during the current clock cycle is the same as for combinational logic. On the other hand, if an SET occurs at the output of one of the flip-flops, the analysis must consider the pulse propagation through the logic part of the circuit with the assumption that the error occurred at one of the inputs (state-line input) of the logic. Therefore, to analyze the propagation of the glitch occurring due to a particle hit in the sequential circuit, that is, the effect of logical, electrical and latching-window masking, we can apply the same analysis as for the combinational logic. However, the main difference between combinational and sequential circuit analysis, when considering their susceptibility to soft errors is that, after being latched in state FFs in sequential circuit, the error can actually be propagated back to the combinational part of the circuit. While the outputs of combinational circuit are affected by the error during a single clock cycle only, in sequential circuits the outputs can be affected during several consecutive clock cycles.

We present in this work two possible symbolic approaches to the modeling and analysis of sequential circuit susceptibility to soft errors. The first approach uses Markov chain theory and the finite state machine description of the circuit and is applicable only to steady-state analysis. The second approach relies on BDDs and ADDs and unrolling of sequential circuits and it is suitable for analysis of time-dependent *SER* behavior. We thus present in this section the basic definitions and notation on finite state machines and Markov chains that will be used throughout the paper, and the main aspects of a BDD/ADD based analysis of SET propagation and *SER* evaluation in combinational circuits.

A. Markov chain analysis of sequential circuits

As an abstraction for sequential circuits, we use the finite state machine (FSM), which can be represented using a state transition graph (STG).

The probabilistic behavior of a sequential circuit is often analyzed using concepts of Markov chain (MC) theory, as

described before [6], [7]. An STG that represents state transitions of the circuit, given input values, can be transformed into the discrete-parameter MC by attaching to each out-going edge of each state a label that represents the transition probability.

The transition probabilities of MC for a given circuit can be calculated when the input distribution that exercises the inputs of the finite state machine (FSM) is known. More precisely, given the transition relation T_M for a FSM M , and input vector probability distribution, $\mathbf{q} = (q_1, q_2, \dots, q_l)$, where l is the number of possible input vector values, the probability p_{ij} of transitioning from state i to state j can be found as:

$$p_{ij} = \sum_{k, T_M(i,k,j)=1} q_k \quad (1)$$

From the transition probability matrix P , it is always possible to calculate the n -step transition probability matrix P^n .

It is often required to determine the long-run behavior of MCs, that is, the *limit state probability*:

$$\pi_j = \lim_{n \rightarrow \infty} p_j(n) \quad j = 0, 1, \dots \quad (2)$$

If, for a given MC, the limit probabilities π_j exist for all states j in the state space I , where π_j does not depend on the initial state i , then $\sum_{j \in I} \pi_j = 1$ and the π_j 's, $j \in I$, are called the *steady-state probabilities* of the MC.

B. BDD/ADD based modeling of SET in combinational circuits

The framework in [9] captures all *gate-output* combinations, i.e., it determines the probability of a soft error at any output due to a fault originating at any internal gate, by using BDDs and ADDs.

For each output F_j , initial duration d_{init} and initial amplitude a_{init} at the output of gate hit by radiation, the authors in [9] find *mean error susceptibility* (MES) as the probability of output F_j failing due to errors at internal gates:

$$MES(F_j^{d_{init}, a_{init}}) = \frac{\sum_{k=1}^{n_f} \sum_{i=1}^{n_G} P(F_j \text{ fails} \mid G_i \text{ fails} \cap \text{init_glitch} = (d_{init}, a_{init}))}{n_G \cdot n_f} \quad (3)$$

where n_G is the cardinality of the set of internal gates of the circuit, $\{G_i\}$ and n_f is the cardinality of the set of probability distributions, $\{f_k\}$, associated to the input vector stream. For each gate G_i , d_{init} and a_{init} , one can find minimum, maximum, mean and median error impact over all outputs F_j that are affected by a glitch occurring at the output of gate G_i . Mean error impact (MEI) for gate G_i is defined as:

$$MEI(G_i^{d_{init}, a_{init}}) = \frac{\sum_{k=1}^{n_f} \sum_{j=1}^{n_F} P(F_j \text{ fails} \mid G_i \text{ fails} \cap \text{init_glitch} = (d_{init}, a_{init}))}{n_F \cdot n_f} \quad (4)$$

where n_F is the cardinality of the set of primary outputs of the circuit, $\{F_j\}$. Similarly, one can find minimum, maximum and median error impact across all outputs and all output probability distributions. For each input probability distribution used, one can also find the number of gates that do not affect any of the outputs.

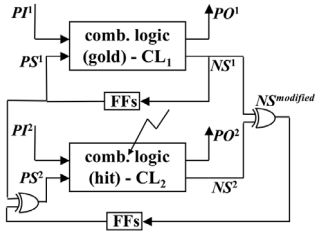


Fig. 3. Circuit model used to perform Markov chain analysis for a given sequential circuit.

The probability of output F_j failing, $P(F_j)$ can be defined using *MES* metric, as described in [9], leading to the expression for soft error rate (*SER*) [21]:

$$SER_{F_j} = P(F_j) \cdot R_{eff} \cdot R_{PH} \cdot A_{circuit} \quad (5)$$

where R_{PH} is the particle hit rate per unit of area, R_{eff} is the fraction of particle hits that result in charge generation, and $A_{circuit}$ is the total silicon area of the circuit. Once $P(F_j)$ is computed for every output (including state lines), one can use the error probability for the state lines to determine steady-state and time-dependent behavior of error propagation in the sequential circuit. We describe in the sequel two such approaches.

IV. MARKOV CHAIN THEORY FOR STEADY-STATE *SER* ANALYSIS

As described in the previous section, the probabilistic behavior of a sequential circuit can be analyzed using Markov chain (MC) theory. Therefore, it naturally leads to the conclusion that we can apply MCs to the probabilistic analysis of sequential circuit soft error susceptibility. In the approaches used in [6], [7], it was shown how to calculate the steady-state behavior of FSMs by means of MC analysis. We describe here one possible method that uses MCs for *SER* analysis.

We propose to modify the original sequential circuit as shown in Fig. 3. The new circuit consists of two copies of the combinational logic of the original circuit, *Combinational logic (gold)*, CL_1 , and *Combinational logic (hit)*, CL_2 . Logic CL_1 is used to collect the information about the correct behavior of the circuit, having as inputs primary input vector (PI^1) and the correct present-state vector (PS^1) and as outputs the correct primary output vector (PO^1) and the correct next state vector (NS^1). On the other hand, circuit CL_2 has as inputs primary input vector (PI^2 , where $PI^2 \equiv PI^1$) and possibly erroneous present-state lines (PS^2) and as outputs possibly erroneous primary output vector (PO^2) and possibly erroneous next-state vector (NS^2). We can define the state vectors of the *gold* and *hit* circuit as:

$$NS^1 = \delta^1 = (\delta_1^1, \delta_2^1, \dots, \delta_m^1) \quad \text{and} \quad NS^2 = \delta^2 = (\delta_1^2, \delta_2^2, \dots, \delta_m^2)$$

where vectors δ^1 and δ^2 can take values from the finite set S of the states of the original circuit. m is the number of state variables. The modified circuit has a new state vector consisting of the state lines (variables) of the original (gold) circuit and an error vector $\epsilon = (\epsilon_1, \epsilon_2, \dots, \epsilon_m)$:

$$NS^{modified} = (\delta^1, \epsilon) = (\delta_1^1, \delta_2^1, \dots, \delta_m^1, \epsilon_1, \epsilon_2, \dots, \epsilon_m)$$

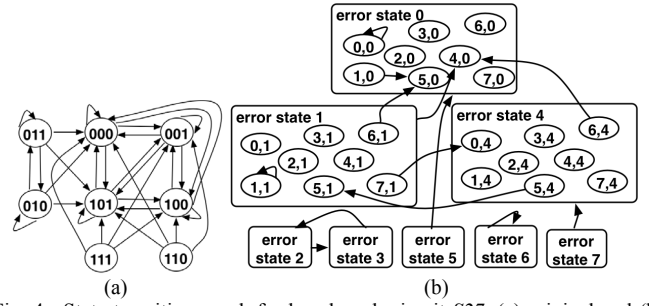


Fig. 4. State transition graph for benchmark circuit S27: (a) original and (b) aggregated set of states (only states 0, 1 and 4 are enlarged to show the aggregation and a small number of transitions is shown).

The error vector ϵ is defined as:

$$\epsilon = \delta^1 \oplus \delta^2 = (\delta_1^1 \oplus \delta_1^2, \delta_2^1 \oplus \delta_2^2, \dots, \delta_m^1 \oplus \delta_m^2)$$

and can take values from the finite set E representing possible errors in the state lines of the original circuit. In other words, a "1" in a component of vector ϵ represents an error in the corresponding element of the state vector δ^1 of the circuit CL_2 , when compared to the state vector δ^1 of the circuit CL_1 : $\epsilon_i = 1$, when there is an error in state line δ_i , and $\epsilon_i = 0$ otherwise, for $i=1,2,\dots,m$. PS^2 vector at the input of CL_2 is then obtained by XOR-ing the PS^1 vector δ^1 and error vector ϵ .

The main goal of the soft error susceptibility analysis for sequential circuits is to find the transition probabilities between the erroneous states from the set E and from there to determine the behavior of the sequential circuit when the soft error occurs. In other words, we are interested in finding the steady-state probability distribution for the values that the error vector ϵ can take. This can be found from the probability vector $\pi^{modified}$ representing the steady-state distribution for the modified circuit by summing the probabilities $\pi^{modified}_{ij} = \pi^{modified}(\delta^{1,i}, \epsilon^j)$ over all vectors $(\delta^{1,i}, \epsilon^j)$ that have the same values $\epsilon_1, \epsilon_2, \dots, \epsilon_m$:

$$\pi_j^{error} = \sum_i \pi_{i,j}^{modified} = \sum_{\delta^{1,i}} \pi^{modified}(\delta^{1,i}, \epsilon^j) \quad (6)$$

We find the STGs for the given original circuit and for its modified version shown in Fig. 3. Fig. 4 (a) shows an STG for an example circuit S27, which has 3-bit state vector (8 states). The modified version of S27 according to Fig. 3 will have 6-bit state vector (64 states). From the STGs of both circuits and given the input vector probability distribution and particle hit probability, we can find their corresponding MCs.

Thus, given the set of states $\{(\delta^1, \epsilon)\}$ and transition probabilities for the modified circuit, $P^{modified}$, and given the initial error state probability $\epsilon(0)$, by using MC theory, we can determine the behavior of the sequential circuit after a soft error occurs. Starting with the initial probability vector $\mathbf{p}(0)$, we can apply various techniques (e.g., power method) on the transition probability matrix $P^{modified}$ to determine the steady-state behavior, under given state error probabilities.

We applied power method to the benchmark circuit S27 for ten different input probability distributions. The initial probability distribution for the error vector has been determined using the approach described in Section III.A. This circuit has the property of fast convergence to steady state. It

can be seen from the STG of this circuit that it has one strongly connected component (SCC) and after entering one of these four states it stays within this part of the STG. Thus, if an error occurs while the circuit is in one of these states, it will be masked after just a few steps. However, this example circuit is very small (only 10 gates, 4 input lines and 3 state lines) and in the case of larger circuits, this method may become inefficient. The pseudo code for this method is given in ALGORITHM 1 (STAGE II – Markovian (power method)).

Working with the full (modified) MC can be prohibitive in terms of cost. While this approach is feasible for small benchmarks like S27 where the modified FSM has 64 states, this can become prohibitively large for larger benchmarks. Since we are interested in transitions between erroneous states only, one possible solution to the complexity problem is to use an approximation of the transition probability matrix $P^{modified}$. An example of such a method is to partition and aggregate the states such that the size of the matrix $P^{modified}$ decreases. This method has been previously used in power analysis and evaluation of sequential circuits [6], [7]. First, we find the transition probability matrix $P^{modified}$ for the modified circuit and assume the starting stationary distribution $\pi^{modified} = (\pi_1^{modified}, \pi_2^{modified}, \dots, \pi_m^{modified})$. Let $A = (A_1, A_2, \dots, A_N)$ be a partition of the state space of the original STG into N blocks (macrostates). We can define a new $N \times N$ matrix $Q = [q_{ij}]$ such that [6]:

$$q_{ij} = \frac{\sum_{i \in A_j} \pi_i^{modified} p_{ij}}{\sum_{i \in A_j} \pi_i^{modified}} \text{ and } p_{ij} = \sum_{j \in A_j} p_{ij} \quad (7)$$

The problem that arises with this method is that it requires the MC to be nearly completely decomposable (NCD) [16], for the approximation to have small error. However, even though the aggregation method converges to an approximate solution, and thus represents an attractive solution for the efficiency of MC approach, the NCD requirement cannot be satisfied for each circuit. We applied the state partition/aggregation method on the example circuit S27, as shown in Fig. 4 (b). For this benchmark circuit, the transition matrix Q obtained using equation (7) converges to the same stationary distribution for erroneous states as the one found using power method on the original transition matrix $P^{modified}$. The pseudo code for this method is given in ALGORITHM 1 (STAGE II – Markovian (aggregation)).

Although established and easy to use, MC analysis has one major drawback: while allowing for the evaluation of long-term or steady-state behavior of the sequential circuit, it fails short in the following when applied to the SER estimation:

- It cannot capture the effect of the error on the outputs of the circuit as a function of time – it only estimates what is the steady-state distribution;
- It cannot include the effect of electrical and latching-window masking, and instead can model only logical masking, unless information is available about the likelihood of a latched error in a state line after a particle hits;
- It becomes impractical for analyzing circuits with larger number of state lines, and thus exponentially larger number

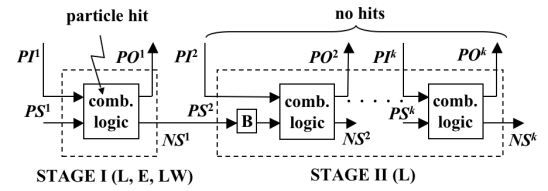


Fig. 5. k -times unrolled sequential circuit divided into two main stages: STAGE I and STAGE II. STAGE II is further subdivided into $k-1$ sub-stages (PI^i : primary inputs of the i^{th} sub-stage, PO^i : primary outputs of the i^{th} sub-stage, PS^i : present state of the i^{th} sub-stage, NS^i : next state of the i^{th} sub-stage, B: state line buffers). In STAGE I, all three masking effects (L, E, LW: logical, electrical and latching-window masking, respectively) are modeled, while in STAGE II only logical masking (L) needs to be considered.

of states. One possible solution is to use the approximation techniques such as aggregation or Monte Carlo simulation, but this can negatively affect the accuracy of the method.

V. A PRACTICAL APPROACH FOR TIME-DEPENDENT SER ANALYSIS

In order to estimate the probability of errors in sequential circuits in an efficient manner that captures both transient and steady-state effects while easily incorporating the *joint* impact of *logical, electrical, and latching window* masking, we use the symbolic framework presented in [9] and briefly described in Section III.B in conjunction with *circuit unrolling*. Since the framework in [9] is used only for combinational circuits, we modified it such that it can be applied to sequential circuits as well.

The main idea of this work is to use unrolling of the sequential circuit, as shown in Fig. 5, in order to allow for efficient time-dependent analysis of the effect of SET on outputs of sequential circuit.

It is important to note here that, when the glitch occurs either at state lines PS^1 or at the output of some internal gate of the combinational logic, it can have a duration much shorter than the clock period and an amplitude smaller than V_{dd} . This means that the glitch can be affected by electrical and latching-window masking. However, if the glitch results in an error in a FF, it will further propagate as a full-cycle error and thus will only be logically masked when not on a sensitized path. Therefore, to use the framework from [9] to analyze the soft error propagation in the clock cycles subsequent to the cycle when the hit occurred, we need to turn off the effect of electrical and latching-window masking in all stages following the first stage.

One possible approach is to analyze the k -unrolled circuit as having two main stages: STAGE I – 1st cycle, STAGE II – 2nd to k^{th} cycles (sub-stages). We can then find the probability of error at each output and each next-state line in STAGE I as described in [9]. In STAGE II, we can lump the logic of sub-stages 2 to k into a single logic circuit for which soft error analysis can be performed using the techniques described in Section III.B. Similar to the analysis of STAGE I, we can employ the same approach applicable to combinational circuits [9], except that we now assume that the glitch occurs only at state line inputs as a full “0” or “1” and thus, can only be masked logically. STAGE-II logic will have $(k-1)$ times more inputs and $(k-1)$ times more outputs. We can then find

ALGORITHM 1. Error probability computation.

STAGE I:

```

set technology parameters;
parse input netlist;
create gate node list;
sort gates topologically;
pass through the sorted list, create all ADDs;
compute initial probabilities {
  for each output and each next state line
  for each gate and each state line
    compute the probability of error; }

```

STAGE II:**UNROLLING:**

```

create k-unrolled circuit gate netlist;
sort gates topologically;
pass through the sorted list, create all BDDs;
compute final probabilities {
  for each output {
    for each state-line
      compute the probability of error; //conditional
    compute final probability of error; }
  }

```

MARKOVIAN:

```

create modified circuit  $M$ ;
create state transition graph for  $M$ ;
for each probability distribution {
  find transition matrix  $P^{modified}$ ;
  case (method) {
    power_method: {
      apply power method to find  $\pi^{modified}$ ;
      find  $\pi^{error}$ ; }
    aggregation: {
      assume initial  $\pi^{modified}$ ;
      apply iterative aggregation to find  $\pi^{error}$ ; }
  }
}
compute final probability of error;

```

The algorithm for STAGE I initial error probability computation and STAGE II final error probability computation, using unrolling and two Markovian approaches (power method and aggregation).

the probability of error for each pair (*state line* – *output*), that is, the probability that the wrong value is latched at the *output*, given that it occurred at *state line*. Therefore, the probability of error at each *output* of STAGE II is a conditional probability, given that an error did occur at the *state line*. For each such output probability value found for STAGE II, we need to multiply it with the probability of error at the corresponding state line. We find these probabilities, for a given input probability distribution using the symbolic framework described in Section III.B, as following:

$$P(F_j^{k,d_{init},a_{init}}) = \sum_l P(F_j^k | F_l^{1,d_{init},a_{init}}) P(F_l^{1,d_{init},a_{init}}) \quad (8)$$

where $P(F_j^{k,d_{init},a_{init}})$ is the probability of output j at the sub-stage k failing, given an initial glitch duration and amplitude, a_{init} and d_{init} . $P(F_j^k | F_l^{1,d_{init},a_{init}})$ is the probability of error at the output j at the stage k , given that an error was latched at the state line l after the first stage with the probability of error at state line l given by:

$$P(F_l^{1,d_{init},a_{init}}) = \frac{\sum_{i=1}^{n_G} P(F_i \text{ fails} | G_i \text{ fails} \cap \text{init_glitch} = (d_{init}, a_{init}))}{n_G} \quad (9)$$

It is important to note here that we need to assume only a hit in the STAGE I of the unrolled circuit and no hits in the consecutive cycles. According to [2],[11], particle hits are sufficiently rare and therefore this assumption is realistic. The

probability $P(F_j^{k,d_{init},a_{init}})$ can be averaged across input probability distributions to find *MES* as in equation (3). As shown in [9], the *MES* value can further be used to find the probability $P(F_j^k)$ of output j failing at sub-stage k and then to compute *SER* as in equation (5).

There is, however, one issue that may arise with this approach. In STAGE I, a single pulse can result in an error on more than one state line. An accurate approach would be to use the global state vector probability distribution and take into account the correlation of errors on state lines, instead of using individual state-line probability distribution. Obviously, the assumption we make (equation (8)) leads to an approximation of output error probability estimation. However, it has been suggested [18] that accurate results using this approach could be obtained by unrolling the logic an infinitely large number of times. This is impractical, but it has been shown [18] that, for the case of switching activity estimation, unrolling the circuit a finite number of times, k , leads to negligible approximation error. More specifically, when using $k=2$, the average error per gate is found to be 2%. In our experiments, we use on average ten unrolled stages for each benchmark and thus, we expect to decrease this error even further.

Since the analysis of the circuit that we convey is probabilistic in nature, we use initial input vector probability distribution for determining output error. More specifically, the input vector for STAGE II of the unrolled circuit is comprised of inputs PI^2 to PI^k to sub-stages 2 to k (which are characterized by the same input probability distribution as PI^1) and PS^2 which are the present state lines after being affected by a possible particle hit in STAGE I. The probability distribution characterizing PS^2 is determined by steady-state analysis of the original sequential circuit (e.g., using MC analysis as in Section IV), while any potential state line error probabilities are determined by using the approach described in Section III.B. Thus, the STAGE II circuit can now be analyzed for individual latched errors on state lines using the approach in Section III.B, but only relying on logical masking effects.

The pseudo code for this method is given in ALGORITHM 1 (STAGE II – unrolling).

VI. GATE RESIZING FOR RADIATION HARDENING

When a high-energy charged particle passes through a semiconductor material, it frees electron-hole pairs along its path as it loses energy and a charge collection can occur, usually within a few microns of the junction. A charge necessary to trigger a change in the data state is called critical charge and it decreases to 10fC for technology nodes below 90nm [15].

One possible efficient solution to increase the critical charge in a logic circuit would be to size the most sensitive gates in the circuit [9],[22]. We use the *MEI* metric (equation (4)) to determine the impact of individual gates on the error susceptibility of the circuit. From the *MEI* values per gate, one

ALGORITHM 2. Gate resizing.

```

resize {
  createResizingList (threshold);
  updateProbabilities (gate_resizing_list);
}


---


createResizingList (threshold) {
  for each gate {
    if Case 1 then
      MEInew ← average MEI across all sub-stages;
    else // Case 2
      MEInew ← MEI at STAGE I;
    if MEInew > threshold then
      add gate to gate_resizing_list;
  }
}


---


updateProbabilities (gate_resizing_list) {
  for each gate {
    if gate in gate_resizing_list then {
      change gate_delay;
      set initial amp. and dur. ADD to zero;
      update output ADDs;
      compute initial probabilities;
      compute final probabilities;
    }
  }
}


---



```

The algorithm for gate resizing that uses *MEI* value from the first stage only (Case 1) and averaged across all sub-stages (Case 2) to create the list of gates to be resized.

can determine which gates have largest impact on *SER* and resize them in order to decrease the *SER*.

When the gate width-length ratio (*W/L*) is changed, the impact that radiation has on that gate is affected. In other words, if this ratio is larger, more charge needs to be generated by a radiation event, so as to result in a glitch of a magnitude larger than the switching threshold of that gate. The current pulse that resulted from the collection of charge induced by radiation, $I_{in}(t)$, can be modeled as [22]:

$$I_{in}(t) = \frac{Q_{coll}}{\tau_{fall} - \tau_{rise}} \cdot \left(e^{-t/\tau_{fall}} - e^{-t/\tau_{rise}} \right) \quad (10)$$

where Q_{coll} is the charge collected by gate and τ_{rise} and τ_{fall} are the collection time constant of the junction and the ion-track establishment time constant, respectively. The voltage at the output of the gate can then be found as shown in [22], when the total capacitance at the output of the gate hit by radiation is known. It has been shown before that, in the case of combinational circuits, resizing the gates with large error impact has a beneficial effect on *SER* [8].

The three major design constraints, area, power consumption and delay, are all affected by the sizing of transistors. The radiation hardening approach proposed in this work is applied only to the nodes that have the highest soft error impact, that is, the nodes that contribute the most to the soft error failure rate of the logic circuit. This decreases the overhead in area when compared to the approach where all gates are hardened. From the gate delay perspective, the effects of sizing a gate are not localized, since other gates are affected as well. This is due to the fact that sizing changes not only the drive strength of a gate, but also the input and output capacitances. As described in [17], the delay of a logic gate can be modeled as:

$$d = \kappa \cdot R \cdot (C_{out} + C_p)$$

where κ is a constant characteristic of the fabrication process, R is the equivalent resistance of the part of the circuit (pull-down or pull-up) that is turned on, C_{out} is the external capacitance driven by the circuit and C_p is the internal (or

parasitic) capacitance driven by the circuit. Given an original gate for which $R = R^{orig}$, $C_{in} = C_{in}^{orig}$ and $C_p = C_p^{orig}$, we can describe its delay when its width is scaled by a factor α and length by a factor β as following:

$$\begin{aligned} d^{new} &= \kappa \cdot R^{new} \cdot (C_{out} + C_p^{new}) = \kappa \cdot R^{new} \cdot C_{in}^{new} \cdot \left(\frac{C_{out}}{C_{in}^{new}} + \frac{C_p^{new}}{C_{in}^{new}} \right) \\ &= \kappa \cdot \left(\frac{R^{orig} \cdot \beta}{\alpha} \right) \cdot (\alpha \cdot \beta \cdot C_{in}^{orig}) \cdot \left(\frac{C_{out}}{\alpha \cdot \beta \cdot C_{in}^{orig}} + \frac{\alpha \cdot \beta \cdot C_p^{orig}}{\alpha \cdot \beta \cdot C_{in}^{orig}} \right) \\ &= \kappa \cdot R^{orig} \cdot \beta^2 \cdot C_{in}^{orig} \cdot \left(\frac{C_{out}}{\alpha \cdot \beta \cdot C_{in}^{orig}} + \frac{C_p^{orig}}{C_{in}^{orig}} \right) \end{aligned} \quad (11)$$

If, for an inverter described with the same model, the equivalent resistance is $R = R^{inv}$, and input capacitance $C_{in} = C_{in}^{inv}$, then the previous expression for gate delay can be written as:

$$\begin{aligned} d^{new} &= \kappa \cdot \beta^2 \cdot R^{inv} \cdot C_{in}^{inv} \cdot \frac{R^{orig} \cdot C_{in}^{orig}}{R^{inv} \cdot C_{in}^{inv}} \cdot \left(\frac{C_{out}}{\alpha \cdot \beta \cdot C_{in}^{orig}} + \frac{C_p^{orig}}{C_{in}^{orig}} \right) \\ &= \kappa \cdot R^{inv} \cdot C_{in}^{inv} \cdot \left(\beta^2 \cdot \frac{R^{orig} \cdot C_{in}^{orig}}{R^{inv} \cdot C_{in}^{inv}} \cdot \frac{C_{out}}{\alpha \cdot \beta \cdot C_{in}^{orig}} + \beta^2 \cdot \frac{R^{orig} \cdot C_p^{orig}}{R^{inv} \cdot C_{in}^{inv}} \right) \end{aligned} \quad (12)$$

We assumed in deriving equations (11) and (12) that both resistance R and capacitance C_{in} are linearly dependent on transistor length, as described in [19]. The resistance R also depends on the threshold voltage [19], but in the equations we used, we assumed an explicit linear dependence of R on channel length, without taking into account possible changes in threshold voltage as the result of changes in transistor length. If necessary, this could easily be modified, and would only affect the impact of the length scaling factor β on resistance R .

As described in [17], we can define unit delay, $\tau = \kappa \cdot R^{inv} \cdot C_{in}^{inv}$, logical effort, $g = \frac{R^{orig} \cdot C_{in}^{orig}}{R^{inv} \cdot C_{in}^{inv}}$, electrical effort, $m = \frac{C_{out}}{C_{in}^{orig}}$, parasitic delay, $p = \frac{R^{orig} \cdot C_p^{orig}}{R^{inv} \cdot C_{in}^{inv}}$, and thus write the gate delay as:

$$d^{new} = \tau \cdot \left(\beta \cdot g \cdot m \cdot \frac{1}{\alpha} + \beta^2 \cdot p \right) \quad (13)$$

We use the previous expression to incorporate the changes in transistor sizes into the delay model used in the symbolic framework. This expression is derived under the assumption that gate sizing is symmetrical, that is, both pMOS and nMOS parts of a gate are scaled by the same factors α and β . In this work, we assume that $\beta = 1$ and α is at most equal to 8 such that the gates selected for resizing cannot be affected by the particles within given energy range. According to equation (12), the delay of a resized gate will be affected by the parameter α and thus will decrease.

It is important to note here that the input capacitance of the gate that is resized is changed such that the new value is:

$$C_{in}^{new} = \alpha \cdot \beta \cdot C_{in}^{orig}$$

For the fanin gates of the resized gate, this capacitance is part of the load capacitance C_{out} , and gate resizing will

TABLE I
METHOD COMPARISON

Comparison of number of steps to reach the (approximate) steady state and error relative to MC power method of three methods: MC power method, MC aggregation and circuit unrolling.

	MC power method	MC aggregation	unrolling
no. steps	163	109	10
relative error [%]	0	0	3E-6

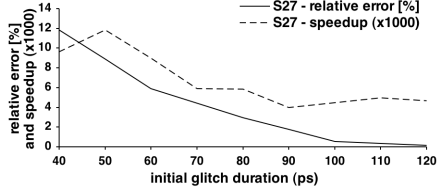


Fig. 6. Comparison of results obtained from HSPICE simulation and symbolic method on benchmark circuit S27.

therefore affect the delay of fanin gates through the electrical effort m .

To find the gates that have largest error impact, we compare their MEI with a given threshold. MEI values computed as in equation (4) represent the error impact of a gate at each sub-stage of the unrolled circuit. Therefore, as a criteria for resizing, we can choose either an average MEI of a gate across all sub-stages (STAGE I and all sub-stages of STAGE II – Case 1), or an MEI for STAGE I only (Case 2). Since for most benchmarks analyzed SER decreases fast within a few cycles, applying the resizing mechanism in Case 2 will be more effective. This is due to the fact that more gates will be resized in this case, leading to a lower overall SER . The pseudo code for gate resizing is given in ALGORITHM 2.

VII. EXPERIMENTAL RESULTS

In this section, we first compare the results obtained using MC analysis and HSPICE simulator with the results obtained using our framework on a small example circuit S27. Then, we show the results of our symbolic model for seven sequential circuits, given different glitch durations and different sets of input probabilities. The technology used is 70nm, Berkeley Predictive Technology Model [4], [25]. The clock cycle period (T_{clk}) used is 250ps, and setup (t_{setup}) and hold (t_{hold}) times for the latches are assumed to be 10ps each. V_{dd} is assumed to be 1V. The delay of an inverter in the given technology is determined by simulating a ring oscillator in HSPICE and found to be 6.5ps. The delays for other gates are found by using logical and electrical effort methodology [17]. The benchmark circuits are chosen from ISCAS'89 suite. The symbolic modeling framework is implemented in C++, and run on a 3GHz Pentium 4 workstation running Linux.

A. MC analysis vs. circuit unrolling

We compare the two MC methods (power method and aggregation) with the unrolling of sequential circuits on benchmark S27 for ten different input probability distributions. In TABLE I, we show the maximum number of steps needed for the power method applied on matrix $P^{modified}$ to converge to steady-state distribution (column MC power method), the number of steps needed for the error state matrix

TABLE II
MINIMUM, MAXIMUM AND AVERAGE SER

SER computed as in equation (5) for the range of glitch durations (60-140)ps, average number of clock cycles needed to reach the steady-state and number of stages and number of probability distributions (PDs) used, run time and memory usage for three glitch durations: small (60ps), medium (100ps) and large (140ps).

Bench.	no. gates	no. PIs	no. POs	no. NSs	SER [FIT]			no. cycles	glitch size	no. sub-stages & PDs	run time (s)	memory usage (MB)
					min	max	average					
S27	10	4	1	3	4.78e-5	0.00134	0.00023	10	small	10,10	0.027	1.4
									medium	10,10	0.028	57.4
									large	10,10	0.032	57.4
S208	68	10	1	8	0.00025	0.00209	0.00054	10+	small	10,10	994	61
									medium	10,10	1000	67.2
									large	10,10	1000	67.2
S298	86	3	14	14	5.25e-5	0.00135	0.00023	10+	small	10,10	6900	71.4
									medium	10,10	6900	71.4
									large	10,10	6950	71.4
S444	153	3	2	21	0	0.00132	0.00029	5	small	5,10	385	61.8
									medium	5,10	365	61.8
									large	5,10	360	61.8
S526	165	3	21	21	1.80e-5	0.001260	0.00027	5+	small	5,10	570	14.3
									medium	5,10	551	18.0
									large	5,10	550	18.0
S1196	487	14	13	18	0	0.00213	0.00043	5	small	5,10	57	17.4
									medium	5,10	68	20.0
									large	5,10	61	20.5
S1238	540	14	13	18	0	0.00212	0.00043	4	small	5,10	64	15.1
									medium	5,10	70	15.9
									large	5,10	71	15.9

found using aggregation to converge to steady-state distribution (column MC aggregation), and the number of sub-stages of the circuit in the unrolling method (column unrolling) needed to reach the SER value smaller than a given threshold (10^{-7} FIT). We also show the error of the method when used given number of steps relative to the power method. As it can be seen, circuit unrolling provides sufficiently accurate results, with orders of magnitude lower time complexity.

B. Symbolic modeling vs. simulation

We use HSPICE simulation to evaluate the accuracy of the results we obtain using approximate symbolic model of the circuit. In Fig. 6, we show the relative error and relative speedup of our model when compared to the HSPICE simulation for benchmark circuit S27 for several initial glitch durations ranging from 40ps to 120ps, assuming exhaustive input sets and considering all gate-output pairs. We find the relative error of our model for a given initial glitch size as:

$$relative_error = \frac{\sum_{k=1}^{n_V} \sum_{i=1}^{n_G} \sum_{j=1}^{n_F} |D_{symbolic}^{ijk} - D_{HSPICE}^{ijk}|}{n_G \cdot n_F \cdot n_V} \cdot D_{HSPICE}^{ijk}$$

where n_G is the number of gates as in equation (3), n_F is the number of outputs as in equation (4), n_V is the number of input vectors, $D_{symbolic}^{ijk}$ and D_{HSPICE}^{ijk} are the durations of the glitch for input vector k and the gate-output pair G_i-F_j , found using our model and HSPICE, respectively. Note that this error includes a *node-by-node* analysis and not just a lumped SER comparison. As it can be seen from Fig. 6, the error stemming from the approximate gate delay model and the attenuation model we are using ranges from less than 1% to about 12% in

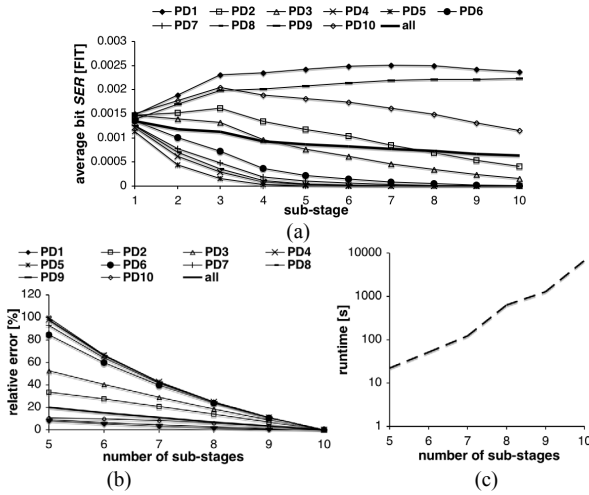


Fig. 7. (a) Average bit *SER* changes in circuit *S298* for ten input probability distributions (PD1-PD10) and averaged across all distributions during ten cycles following the particle hit; (b) Relative difference in *SER* of (5-9) unrolled sub-stages compared to ten unrolled sub-stages; (c) Runtime of the framework for different number of unrolled sub-stages (5-10).

one instance (40ps glitch), while averaging 4% overall for an effective 5500X average speedup (up to 11000X in some cases).

C. *SER* evaluation

In TABLE II, we present *SER* for several *ISCAS'89* benchmark circuits found using equation (5). The allowed interval for the initial duration of the glitch is assumed to be $(d_{\min}, d_{\max}) = (60, 140)$ ps, while initial amplitude is in the range $(a_{\min}, a_{\max}) = (0.8, 1)$ V. Since for glitches smaller than 60ps all benchmark circuits (except for a few that have very small number of gates) have output error induced mostly by output gates and their fanin gates in STAGE I, we use this duration as the lower bound of our interval. Similarly, as already explained, for glitches longer than 140ps, all benchmarks propagate almost all the glitches, and thus we use this as an upper bound. *MES* for each output is found within these allowed intervals at incremental steps $\Delta d = 20$ ps and $\Delta a = 0.1$ V. The R_{PH} used is $56.5 \text{ m}^2\text{s}^{-1}$, R_{eff} is $2.2 \cdot 10^{-5}$, and the total silicon area for each benchmark circuit is derived as a function of gate count. As it can be seen from results presented in TABLE II, the *SER* behavior is different among the benchmark circuits, that is, the *SER* decreases very fast (e.g., for circuits *S1196*, *S1238*) or stays at about the same level for all ten clock cycles for which the circuit is unrolled (e.g., for circuit *S208*). This difference in number of cycles needed for the circuit to go back to non-erroneous state stems from the logic of the circuit and logical masking as well as from the number of state lines that can drive errors back to the state line inputs of the circuit.

The benchmark circuits for which the results are presented in TABLE II have up to 20 primary inputs (PIs) or next state lines (NSs). In our previous work [8], we have shown results for some combinational circuits that have up to 40 PIs. As already described, the number of variables, i.e., non-terminal nodes in BDDs/ADDs represent primary inputs and state lines and thus, number of flip-flops in conjunction with the number of primary inputs, can increase the size of BDDs/ADDs. This

can further increase the time necessary to create final output ADDs and the time needed to find the probabilities from those ADDs. This runtime becomes an issue only for circuits with overall number of primary input and state lines of more than 40 or 50. We reported in TABLE II the results for those benchmark circuits for which running of the framework finished in a reasonable amount of time. However, our framework is not limited to only these circuits and can be run on larger benchmarks as well, but it will require longer run time. It is also important to note that, when compared to the previous work [1],[14],[20],[21], our approach gives better run time, while being within 4% accurate. Moreover, this scalability “limitation” of the current work can further be overcome by dividing the circuit into smaller parts (sub-circuits) and computing output ADDs for those sub-circuits. We intend to apply this approach in our future work and thus, improve even more the scalability of our framework.

There is also an additional aspect of *SER* in sequential circuits that we analyzed and is shown in Fig. 7. When different input probability distributions are applied, the behavior of the circuit is uniform in most cases, that is, the *SER* usually decreases (except for the circuit *S208*) through the sub-stages of the unrolled circuit. However, in a few cases, different input probability distributions result in a different *SER* trend. As shown in Fig. 7(a), for circuit *S298*, input probability distributions PD2 and PD10 results in an *SER* jump in the second stage and a *SER* decrease afterwards, while PD9 leads to the increase in *SER* in the sub-stages following the hit. For PD4, PD5 and PD8, the *SER* decreases fast after the hit. This is due to the fact that different input probability distributions will result in different sensitization probabilities of the paths in the circuit, thus affecting the output error probabilities differently. On the other hand, the experiments we conducted have shown that the time-dependent behavior of different outputs of a sequential circuit varies such that for

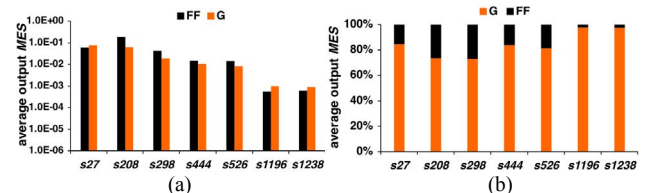


Fig. 8. Output *MES* for seven benchmark circuits for 120ps initial glitch duration (a) averaged assuming hits at flip-flops (FF) or internal gates (G) only and (b) normalized with respect to the overall number of gates and flip-flops, assuming hits at both flip-flops (FF) and internal gates (G).

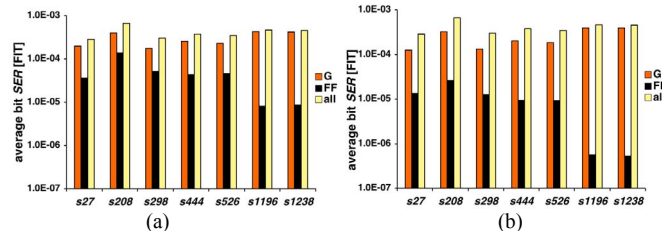


Fig. 9. Output *SER* (a) assuming hits at flip-flops only (FF), gates only (G) and averaging *MES* across overall number of gates and flip-flops, or assuming hit at both flip-flops and gates (all) and (b) assuming hits at flip-flops only and averaging *MES* across all flip-flops (FF), gates only and averaging *MES* across all gates (G) or assuming hits at both flip-flops and gates (all).

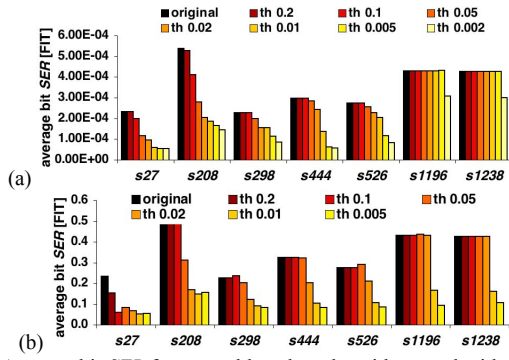


Fig. 10. Average bit *SER* for several benchmarks without and with gate sizing for several *MEI* thresholds: (a) Case 1: *MEI* averaged across all stages and (b) Case 2: *MEI* in the 1st stage.

some outputs the *MES* decreases in the sub-stages following the hit, while for the others it either increases in the second sub-stage and then decreases, increases slightly, or remains at the same level.

Since the average *SER* for circuit *S298* decreases in cycles following the particle hit, larger number of unrolled copies gives better accuracy, closer to the steady-state value. However, as the difference between the *SER* computed in subsequent cycles is decreasing, we evaluate the accuracy of different number of unrolled sub-stages as the following. First, we find the average *SER* in each cycle (sub-stage) following the particle hit (presented in Fig. 7(a), line **all**). Next, we find the average *SER* across all sub-stages (assuming different cases, i.e., different number of unrolled sub-stages, 5-10). Finally, we compute the relative difference in this average *SER* between the case with ten unrolled copies and the cases with smaller number of unrolled copies. This relative difference in *SER* for circuit *S298* is presented in Fig. 7(b), and it can be concluded from the figure that the difference decreases to 3% when the results for nine and ten unrolled copies are compared.

We also show the changes in runtime for different number of unrolled sub-stages for circuit *S298* (Fig. 7(c)). We used the large glitch (120ps long) to measure the runtime, since in that case the effect on runtime is emphasized the most. However, in order to compute *SER*, we used several different glitch sizes. As it can be seen from Fig. 7(c), the runtime for circuit *S298* increases exponentially when increasing the number of unrolled copies. This may seem as the limitation of the proposed approach, but as we described in Section V, and as the comparison with HSPICE simulations has shown (Section VII.B), unrolling of up to ten sub-stages provides a reasonable runtime and the results that are on average 4% accurate.

The results for one small benchmark *S444* (153 gates, 3 inputs) and one larger benchmark, *S1196* (487 gates, 14 inputs) are presented in Fig. 11 (two left charts). As it can be seen from Fig. 11, both circuits converge to steady-state after five clock cycles after the hit. The only difference between these two circuits is the magnitude of *SER*.

D. Flip-flop vs. gate error impact

Assuming that a particle hit can occur at either flip-flops only or gates only, we show in Fig. 8 (a) the average output *MES* for several benchmark circuits. Note that, due to the

assumption we make, we find the *MES* for each output using equation (3), with the difference that the total number of gates n_G is given by number of flip-flops only or number of internal gates *only* (and not the overall number of gates *and* flip-flops). Doing so, we can see that if only flip-flops or only internal gates are considered, the output *MES* values are comparable and have similar values in both cases.

However, these results may be misleading, since in typical sequential circuits, the number of internal gates is usually much larger than the number of flip-flops. Thus, in Fig. 8 (b), we show the *MES* values assuming effects from both flip-flops and internal gates, normalized to 100% so the effect of each component (flip-flops vs. internal gates) can be determined easily. As it can be seen from Fig. 8 (b), when normalized with respect to all possible particle hit locations, the impact of flip-flops is 3-49X smaller than the impact of internal gates. While this confirms the fact that the impact of internal gates on output error susceptibility is much larger than that of flip-flops (e.g., larger circuits like *S1196* and *S1238*), it also shows that for smaller sequential circuits, the impact of flip-flops cannot be ignored either (e.g., circuits *S208* and *S298*).

We further consider the *SER* stemming from either flip-flops only or gates only, and compare it to the overall *SER*. Since the *SER* can be computed using *MES* values [9], we can take two different approaches: either assume that flip-flop or gate *MES* is found by averaging across flip-flops or gates only (as in Fig. 8 (a)), or assuming that the *MES* is found by averaging across overall number of gates and flip-flops (as in Fig. 8 (b)). The *SER* results for these two possible approaches are shown in Fig. 9 (a) and Fig. 9 (b). While the results presented in Fig. 8 (a) and Fig. 8 (b) are found for only one glitch size, the results in Fig. 9 (a) and Fig. 9 (b) are computed across different glitch sizes. Furthermore, the results in **FF** and **G** bar in Fig. 9 (a) are found by multiplying the probability of error at a given output with the area of flip-flops and gates, respectively, while the results in **all** bar are found from overall circuit *MES* and by multiplying with the overall circuit area. The final *SER* results in Fig. 9 (b) are found similarly, while the only difference is in computation of *MES*, as already described. As it can be seen from Fig. 9 (a) and Fig. 9 (b), the *SER* stemming from gates is higher than the *SER* stemming from FFs, for all benchmark circuits.

E. Gate resizing impact on *SER*

In the HSPICE experiments that were conducted to find the necessary radiation hardening resizing factors for gates, the parameter values that were used are $Q_{coll} = 60\text{fC}$, $\tau_{rise} = 1\text{ps}$ and $\tau_{fall} = 40\text{ps}$. These values are found according to the data from [22] and also assuming that the resulting current pulse lies on the upper bound of the glitch sizes that we used in our *SER* experiments. It is important to note that the main goal here was to show that it is possible to incorporate soft error mitigation techniques into our framework, and gate resizing was one example of such technique. Therefore, the parameters that we assumed in our HSPICE experimental setup are by no means the limiting factor for the overall framework, and can easily be updated to accommodate more realistic cases, if such

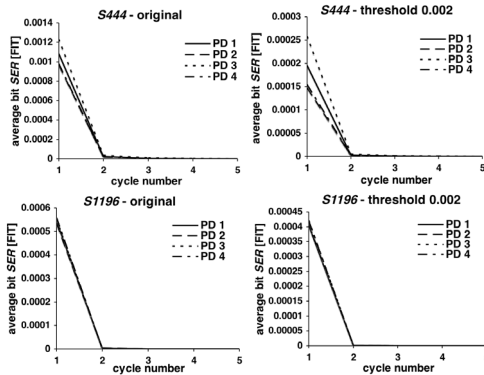


Fig. 11. *SER* changes in circuits *S444* and *S1196* during five clock cycles for different input probability distributions without gate resizing (left charts) and with gate resizing (right charts).

cases exist. Next, we create the HSPICE circuit description in which current $I_{in}(t)$ (equation (10)) is included in the chain of gates as a current source at the output of a middle gate. The size (width) of the gate is increased, until the point where the current source is not strong enough to flip the value of the output signal of the gate is reached. This procedure is repeated for each gate type from the library that we are using. We can further use the new gate width in our framework to find new delay of the resized gate and its input neighbors.

We show in Fig. 10 the *SER* for several benchmark circuits before (**baseline**) and after gate resizing (**threshold 0.2, 0.1, 0.05, 0.02, 0.01, 0.005 and 0.002** columns). Additionally, in Fig. 11 (right two charts) we show the impact of gate resizing in five cycles following the particle hit, for a threshold 0.002 (Case 1) for benchmarks *S444* and *S1196*. As it can be seen from Fig. 11, gate resizing can have different impact on the overall circuit *SER* for different input probability distributions and the improvement in *SER* can range up to 90% in STAGE I. In Fig. 10 (a), resizing is applied to gates whose *MEI* averaged across all stages (STAGE I and STAGE II) of an unrolled circuit is larger than a given threshold ranging between 0.002 and 0.2. The difference between the range of *SER* values for circuits in Fig. 11 and Fig. 10 are due to the fact that, in Fig. 11 only four distributions are used, as opposed to ten different distributions for Fig. 10. In Fig. 10 (b), we show the results when the resizing is applied to gates whose *MEI* in STAGE I only is larger than a given threshold ranging between 0.005 and 0.2. As it can be seen from the presented results, resizing can improve *SER* by as much as 80% for a threshold of 0.01 (for circuit *S444*).

The results presented in Fig. 10 (a) (Case 1 criteria) show that *SER* decreases monotonically with the decrease in resizing threshold, while in Fig. 10 (b) (Case 2 criteria) this is not the case. There are two factors that result in this kind of a behavior. First, different gates in the circuit can have different impact on overall circuit error: some gates have *MEI* that decreases fast through sub-stages of the unrolled circuit, while for some gates *MEI* remains at about the same level. Thus, there are cases when one gate has higher STAGE I *MEI* than some other gate, but on average, the *MEI* of the first gate is smaller than the *MEI* of the second one. Second, when gates are resized in order to increase the critical charge, the gate delay is actually decreased and this affects the glitch

attenuation. If the impact of the glitches originating at the resized gate is not significant, but the attenuation that this gate provides is important, it may happen that the overall circuit *SER* increases due to resizing. Since for the Case 2 criteria the order in which gates are chosen for resizing does not reflect the monotonicity in their average *MEI*, the *SER* increase for smaller thresholds is more emphasized. On the other hand, Case 2 criteria allows for more gates to be resized leading to the faster decrease in *SER*. However, there is a tradeoff, since this also leads to higher area overhead. The area overhead varies for different glitch sizes and different benchmarks. For example, when the initial glitch is 60ps long, the number of gates resized is minimal and varies from 1.7% for benchmark *S444* (threshold 0.01) to 38% for benchmark *S27* (threshold 0.002) in Case 1, and 0.3% for benchmark *S1238* (threshold 0.01) to 55% for benchmark *S208* (threshold 0.005) in Case 2. Moreover, Case 2 approach treats the sequential circuit as if it is a combinational circuit, since its criterion is defined with respect to the first cycle only. Therefore, Case 2 is not suitable for *SER* reduction in sequential circuits and (as shown in Fig. 10 (b)) it does not guarantee a reduction in *SER* in subsequent cycles.

VIII. CONCLUSION

In this paper, we presented a symbolic modeling methodology for efficient estimation of the soft error susceptibility of a sequential circuit. We have demonstrated the efficiency of our method by comparing it to another symbolic model that relies on MC theory, to HSPICE detailed circuit simulation and applying it on a subset of *ISCAS'89* benchmarks of various complexities. We have also shown that, by using the information obtained from the framework, we can resize the gates that have largest impact on circuit reliability, such that their impact is decreased and *SER* is improved. Finally, we have shown that the choice of criterion for gate resizing can affect the *SER* reduction in cycles following the particle hit.

REFERENCES

- [1] G. Asadi and M. B. Tahoori. *Soft Error Modeling and Protection for Sequential Elements*. In Proc. of IEEE Symposium on Defect and Fault Tolerance (DFT) in VLSI Systems, pp. 463-471, October 2005.
- [2] H. Asadi and M. B. Tahoori. *Soft error hardening for logic-level designs*. In Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS), pp. 4139-4142, May 2006.
- [3] R. C. Baumann. *Soft Errors in Advanced Computer Systems*. In IEEE Design and Test of Computers, Vol. 22, Issue 3, 2005.
- [4] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu. *New paradigm of predictive MOSFET and interconnect modeling for early circuit design*. In Proc. of IEEE Custom Integrated Circuits Conference (CICC), pp. 201-204, June 2000.
- [5] P. Dodd. *Basic Mechanisms and Modeling of Single-Event Upset in Digital Microelectronics*. In IEEE Transactions on Nuclear Science, Vol. 50, No. 3, pp. 583-602, June 2003.
- [6] G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi. *Markovian Analysis of Large Finite State Machines*. In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), Vol. 15, No. 12, pp. 1479-1493, December 1996.
- [7] D. Marculescu, R. Marculescu, and M. Pedram. *Trace-Driven Steady-State Probability Estimation in FSMs with Application to Power Estimation*. In Proc. of IEEE Design, Automation and Test in Europe Conf. (DATE), February 1998.

- [8] N. Miskov-Zivanov, D. Marculescu. *MARS-C: Modeling and Reduction of Soft Errors in Combinational Circuits*. In Proc. of Design Automation Conference (DAC), pp. 767-772, July 2006.
- [9] N. Miskov-Zivanov and D. Marculescu. *Circuit Reliability Analysis Using Symbolic Techniques*. In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), pp. 2638-2639, December 2006.
- [10] S. Mitra, N. Seifert, M. Zhang, Q. Shi, and K. S. Kim. *Robust System Design with Built-In Soft-Error Resilience*. In IEEE Computer Magazine, Vol. 28, No. 2, pp. 43-52, February 2005.
- [11] K. Mohanram and N. A. Touba. *Cost-Effective Approach for Reducing Soft Error Failure Rate in Logic Circuits*. In Proc. of International Test Conference (ITC), pp. 893-901, 2003.
- [12] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Inc., pp. 635-640, 1991.
- [13] R. R. Rao, K. Chopra, D. Blaauw and D. Sylvester. *An Efficient Static Algorithm for Computing the Soft Error Rates of Combinational Circuits*. In Proc. of the Conference on Design, Automation and Test in Europe (DATE), pp. 164-169, March 2006.
- [14] R. R. Rao, D. Blaauw and D. Sylvester. *Soft Error Reduction in Combinational Logic Using Gate Resizing and Flipflop Selection*. In Proc. of International Conference on Computer Aided Design (ICCAD), pp. 502-509, November 2006.
- [15] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi. *Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic*. In Proc. of International Conference on Dependable Systems and Networks, pp. 389-398, 2002.
- [16] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, pp. 285-342, 1994.
- [17] I. Sutherland, B. Sproull and D. Harris. *Logical Effort: Designing Fast CMOS Circuits*. Morgan Kaufmann Publishers, Inc., pp.5-15, pp. 63-73, 1999.
- [18] C. Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, and A. M. Despain. *Power Estimation Methods for Sequential Logic Circuits*. In Proc. of the IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 3, No. 3, pp. 404-416, September 1995.
- [19] N. H. E. Weste and D. Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Pearson Education, Inc., pp. 77-84, 103-105, 2005.
- [20] B. Zhang, W. Wang, and M. Orshansky. *FASE: Fast Analysis of Soft Error Susceptibility for Cell-Based Designs*. In Proc. of International Symposium on Quality Electronic Design (ISQED), March 2006.
- [21] M. Zhang and N. R. Shanbhag. *A Soft Error rate Analysis (SERA) Methodology*. In Proc. of ACM/IEEE International Conference on Computer Aided Design (ICCAD), pp. 111-118, 2004.
- [22] Q. Zhou and K. Mohanram. *Transistor Sizing for Radiation Hardening*. In Proc. of the IEEE Annual International Reliability Physics Symposium, pp. 310-315, 2004.
- [23] Q. Zhou and K. Mohanram. *Gate sizing to radiation harden combinational logic*. In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), Vol. 25, No. 1, pp. 155-166, January 2006.
- [24] J. F. Ziegler et al. *IBM experiments in Soft Fails in Computer Electronics (1978-1994)*. In IBM Journal of Research and Development, Vol. 40, pp. 3-18, 1996.
- [25] *Berkeley Predictive Technology Model (BPTM)*: <http://www-device.eecs.berkeley.edu/~ptm>.



Natasa Miskov-Zivanov received the B.S. degree in electrical engineering from the School of Technical Sciences, University of Novi Sad, Serbia, in 2003 and the M.S. degree in computer engineering from Carnegie Mellon University, in 2005.

She is currently pursuing the Ph.D. degree in computer engineering at Carnegie Mellon University. Her research interests include fault-tolerance in nanoscale designs and emerging technologies.



Diana Marculescu is currently an Associate Professor of Electrical and Computer Engineering at Carnegie Mellon University. She received her M.S. in Computer Science from "Politehnica" University of Bucharest, Romania in 1991 and her Ph.D. in Computer Engineering from University of Southern California in 1998. She is the recipient of a National Science Foundation Faculty Career Award (2000-2004), an ACM-SIGDA Technical Leadership Award (2003), the Carnegie Institute of Technology George Tallman Ladd Research Award (2004)

and a Best Paper Award from IEEE Asia South-Pacific Design Automation Conference (ASPDAC 2005). Diana Marculescu was an IEEE-Circuits and Systems Society Distinguished Lecturer (2004-2005) and is currently Chair of the ACM Special Interest Group on Design Automation (SIGDA) and member of IEEE. Her research interests include energy aware computing, CAD tools for low power systems and emerging technologies.