Multiple Transient Faults in Combinational and Sequential Circuits: A Systematic Approach

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

Abstract – Transient faults in logic circuits are becoming an important reliability concern for future technology nodes. Radiation-induced faults have received significant attention in recent years, while multiple transients originating from a single radiation hit are predicted to occur more often. Furthermore, some effects, like reconvergent fanout-induced glitches, are more pronounced in the case of multiple faults. Therefore, to guide the design process and the choice of circuit optimization techniques, it is important to model multiple faults and their propagation through logic circuits, while evaluating the changes in error rates resulting from multiple simultaneous faults. In this paper, we show how output error probabilities change with increasing number of simultaneous faults and we also analyze the impact of multiple errors in state flip-flops, during the cycles following the cycle when fault(s) occurred. The results obtained using the proposed framework show that output error probability resulting from multiple-event transient or multiple-bit upsets can vary across different outputs and different circuits by several orders of magnitude. The results also show that the impact of different masking factors also varies across circuits and this information can be valuable for customizing protection techniques.

Index Terms – logic circuits, reliability, transient faults, symbolic manipulation.

I. INTRODUCTION

The scaling of device feature sizes, operating voltages and design margins raises the susceptibility of circuits to *transient faults* (TFs). Such faults can be caused by different physical phenomena, such as high-energy particle hits originating from cosmic rays, capacitive coupling, electromagnetic interference, or power transients [28].

In recent years, transient faults induced by radiation received much of the attention, as they are claimed to be one of the major challenges for future technology scaling [4]. A transient fault in a logic circuit, resulting from a single particle hit, is often referred to as a *single-event transient* (SET) and an error in a memory element that results either from an SET (glitch or pulse), or from direct radiation hit is called a *soft error* or a *single-event upset* (SEU). The effect of soft errors is measured by the *soft error rate* (*SER*) in FIT (*failure-in-time*), which is defined as one failure in 10^9 hours.

In the past, soft errors used to be a concern only in memories, thus resulting in widely used Error Correcting Codes (ECC) mitigation techniques.

On the other hand, at 90nm and beyond, the reduced device dimensions and operating voltage levels lead to more common radiation-induced faults in logic, with resulting error rates approaching those of memories [31]. Therefore, transient faults in logic circuits are becoming an important reliability concern for future technology nodes. Furthermore, since the distance between junctions decreases with scaling and the critical charge is reduced, the energy of radiation particles that is required to cause multiple transient faults is decreasing. The probability that a single high energetic particle affects the output of more than one circuit node is no longer negligible [26] and if the two (or more) affected nodes belong to different logic gates, multiple transient faults (MTFs) can be generated and propagated to logic circuit outputs.

Future technology nodes will face increased rate of MTFs, stemming from either *single events* (SE-MTF), such as a radiation hit, or different, but simultaneous *multiple events* (ME-MTF), such as crosstalk, ground bounce, IR drop or radiation [2][9][34]. Although so far there have been several design solutions that tackle the transient fault issue, the remaining challenge (emphasized by technology scaling) is finding low cost solutions exhibiting best tradeoffs among power, performance, and cost on one hand, and reliability, on the other hand [13]. Reliability analysis is proven to be essential in early design stages for improving system lifetime and for allowing exploration of existing tradeoffs [2]. Therefore, a fast and accurate estimation of error rates resulting from *single transient faults* (STFs) and *multiple transient faults* (MTFs) in logic circuits is crucial for identifying the features needed for future reliable circuits.

In this work, we present an efficient and accurate methodology for evaluating the impact of single and multiple transient faults in combinational and sequential circuits. The framework described here models all important factors involved in transient fault propagation in logic circuits in a unified manner and allows for comprehensive probabilistic analysis of circuit reliability. The framework supports transient behavior analysis of sequential circuits during the cycle when the transient fault occurs, as well as during subsequent cycles.

The rest of the paper is organized as follows. In Section II, we describe previous work on soft error modeling and analysis and briefly outline the contributions of our work. Section III provides an overview of the preliminaries of a transient fault analysis in logic circuits, with the focus on multiple faults. The proposed multiple fault modeling methodology is described in Section IV. In Section V, we describe the computation of output and circuit error probability. In Section VI, we show the experimental results obtained using the proposed framework and with Section VII we conclude our work.

II. RELATED WORK

The focus of our work is on ***modeling*** of transient faults (both single and multiple faults) and analyzing their effect on ***logic circuits***. Transient faults, and especially, radiation-induced faults, have been extensively studied in recent years. A number of approaches were proposed to tackle the problem of evaluation of logic circuit susceptibility to transient faults. Due to space consideration, we describe in the following only previous work that is closely related to our work (*i.e.*, transient fault modeling in logic circuits), and present the contributions of our work.

A. Single transient fault modeling

One obvious approach to analyze the impact of transient faults is to inject the fault at a given node of the circuit and simulate the circuit for different input vectors so as to find whether the fault propagates. However, this approach becomes intractable for large circuits and large number of inputs, therefore requiring approximate approaches that use analytical and symbolic methods to evaluate circuit susceptibility to transient faults.

A number of methods have been proposed recently to evaluate the susceptibility of combinational logic circuits to soft errors [1][5]-[7][10][11][14][16][22][23]-[25][32]-[34]. Common to these methods is the fact that they model some aspects of glitch propagation analytically or symbolically by avoiding direct circuit (or device) simulation, therefore providing a tradeoff between accuracy and efficiency. However, those approaches usually fail to include at least one of the following:

- modeling of all three masking factors (logical, electrical and latching-window masking) that affect the propagation of glitches through logic circuits and their latching at circuit outputs;
- unified treatment of all three masking factors, necessary due to the fact that their impact on propagated glitch is not independent;
- modeling of reconvergent glitches.

The importance of including all these aspects of transient fault modeling is detailed in Section III.B. Furthermore, another drawback of many of these methods is the fact that they are not capable of keeping track of all possible input vectors and thus, estimate circuit error rates for only a subset of input vectors. This selection of input vectors significantly limits the accuracy of such methods, especially in the case of large circuits. Finally, such an approach does not provide better efficiency since they usually use logic simulation and path tracing to compute output values and the impact of logical masking.

Our previously proposed approach [19] uses symbolic Binary Decision Diagrams (BDDs) and Algebraic Decision Diagrams (ADDs) to model the propagation of a fault in logic circuits. The main difference between this and other non-simulative approaches is its efficiency and accuracy, due to the fact that it models all three masking factors (logical, electrical and latching-window masking) in a unified manner, models glitch reconvergence, and allows for the analysis of circuit error probability for different input patterns by performing only one pass through the topologically ordered circuit [19][20]. We use this approach as a basis for modeling multiple faults and multiple upsets (latter one in the case of sequential circuit flip-

flops), as will be explained in Section IV. Furthermore, the approach proposed in [19] can be used for accurate evaluation of individual and joint output error susceptibilities, since BDDs and ADDs allow for determining correlations among circuit output errors. We present details of output error probability computation in Section V.

A transient fault can affect outputs of a sequential circuit during several clock cycles. Therefore, the sequential circuit approach requires modeling the transient behavior of the circuit affected by TF(s), and not only the steady-state. To consider this effect, the analysis of the propagation of a TF through sequential circuit must be done for more than one clock cycle. There have been just a few approaches that tackle this issue [1][11][20], however, only the work we proposed in [20] models transient faults in sequential circuits, while also including all the other important modeling aspects listed above.

B. Multiple transient fault and multiple upset modeling

The problem of the impact of multiple transient faults occurring simultaneously has been addressed in the past, but with the focus on memories [17][18], in the light of the multiple upsets resulting from a single transient phenomenon. Until recently [26][27], MTFs in logic received very little attention, due to their rare occurrence. These trends are changing [2][15][26][27], making modeling and analysis of MTFs as important as modeling and analysis of STFs.

Similar to STF, previous work that focused on MTFs in logic circuits used, to the best of our knowledge, only simulation. For example, the authors of [27] used simulation to estimate the sizes of multiple transients resulting from a single particle hit, as well as the impact of different gate input combinations on the output transient current. Approaches focusing on the effect of multiple transients on the error rate of the overall circuit used simulation at either device [26] or circuit level [17][27]. Typically, faults were injected at different nodes in the circuit and the impact of those faults on circuit outputs for different input combinations was estimated.

C. Paper contribution

As already stated at the beginning of this Section, the main focus of this work is on:

- **modeling of transient fault propagation**, once the fault(s) occur at the output of a gate inside the logic circuit;
- **accurate and efficient computation of error probabilities** of circuit outputs due to propagated transient fault(s).

Transient faults can have different sources in logic circuits, as mentioned in Section I and in [2][31]. It is thus also important to note that the specific parameters related to the occurrence rate of transients (*e.g.*, in case of radiation-induced faults, these parameters include particle hit rate and the ratio of effective hits) are not directly part of the proposed framework, but instead are included as inputs to the framework. However, once the fault(s) occur, their propagation and latching is affected by same mechanisms (*e.g.*, logical, electrical and latching-window masking). Therefore, with respect to STF and MTF, this work improves state-of-the-art by allowing for:

1. **Accurate and efficient modeling** and analysis of the impact of both STFs and MTFs in logic (combinational and sequential) circuits;

2. Evaluation of **changes in error rates due to MTFs in sequential circuits**, following the cycle when the transient fault occurs within the circuit;

3. Evaluation of the impact of **multiple flip-flop upsets in sequential circuits**;

4. Estimation of **individual** output error susceptibility and the probability of **correlated output errors**;

5. Determining the **outputs that are most susceptible** to errors due to transient faults in logic;

6. Determining the parts of the circuit (**gates or gate clusters**) that have the **largest impact on circuit error** probability;

7. Estimation of **lower and upper bounds of circuit susceptibility** to transient faults;

8. Analysis of the **impact of individual masking factors** on error probability in different circuits.

In Section VI, we present some results with respect to radiation-induced faults, when the hit rate and the ratio of effective particle hits is taken into account. The main purpose of these results is to show how the proposed modeling approach can be used when the fault source is specified. We do not present here the error rate evaluation in case of other fault sources, as this is not the main focus of this work. However, we intend to look at other fault sources in the future and compare individual and combined error rates for such cases.

III. MTF MODELING PRELIMINARIES

In this section, we briefly describe the main principles of modeling STFs and MTFs in logic circuits. Since radiation-induced transient faults are recognized as one of the major concerns for circuit designers, we first give an example of issues and decisions that need to be considered when analyzing radiation-induced transient faults. We then provide an overview of the important aspects of modeling the transient fault propagation, in general.

A. Example: Multiple fault generation due to particle hit

When a high-energy charged particle passes through a semiconductor material, it frees electron-hole pairs along its path, as it loses energy. Charge collection generally occurs within a few microns of the junction. The collected charge for the radiation-induced events in silicon can range from one to several hundreds of fCs [8]. The device sensitivity to this excess charge is defined primarily by the node capacitance, operating voltages, the strength of feedback or fanout transistors, all defining the amount of critical charge required to trigger a change in the data state. Critical charge for technology nodes below 90nm decreases to 10fC [31].

When an energetic particle hits a device at an oblique angle, there is a small, but non-zero probability of disturbing more than one sensitive junction. The larger the particle track and the closer the junctions are, the larger the probability is for upsetting more than one junction [17]. The most probable location of the occurrence of multiple transients, considered at the logic level, is at the outputs of neighboring gates. The spatial 'neighbor' relation among logic gates can be best determined from the layout. However, in the absence of layout information, one possible approach is to assume that a gate and its fanin or fanout neighbors, or gates that have a common fanin or fanout neighbor are possible candidates for SE-MTF generation.

There are several factors that need to be considered when modeling SE-MTFs. First, the exact relationship between two simultaneous transients generated by an energetic particle hitting two junctions, and the same particle affecting only one junction cannot be determined in a straightforward manner. For example, given the particle hit, it is necessary to know how the charge collected by a single junction, Q_{coll}, compares to the charge induced by the same energetic particle, but collected by two or more junctions ($Q_{coll,1}$, $Q_{coll,2}$,…,$Q_{coll,n}$). One possible assumption is that:

$$Q_{coll} \geq \sum_{i=1}^{n} Q_{coll,i} \tag{1}$$

where the inequality stems from the fact that the charge spread across several nodes may result in less overall charge being collected. However, the exact relationship between the charge collected by a single node, Q_{coll}, and the sum of the charges collected by multiple nodes is also affected by the incident particle angle and the collection capacity of nodes.

Next, the MTFs that can result from a single hit are not necessarily uniform and can be of different sizes. Even if we assume *equality* in equation (1), the relationship between resulting glitch sizes is not the same as relationship between the collected charges, *i.e.*:

$$D_{SE-STF} \neq \sum_{i=1}^{n} D_{SE-MTF,i} \tag{2}$$

where D_{SE-MTF} is the duration of the glitch resulting from Q_{coll} and $D_{SE-MTF,i}$ is the duration of the glitch in SE-MTF resulting from $Q_{coll,i}$. We conducted HSPICE simulations of different gates to determine the sizes of glitches resulting from a given collected charge. We show the results for a NAND gate in Figure 1, assuming different gate load values (fan-out-of-1, FO1, and fan-out-of-2, FO2) and different collected charge (from 10 to 200fC). The **original** curve in Figure 1 represents the duration of the glitch resulting from a given collected charge, while the **sum** curve represents the sum of glitch durations, assuming one glitch results from a fixed charge: (a) 10fC; (b) 20fC; (c) 40fC; and (d) 20fC. The collected charge for the second glitch is varied, starting with the same value as for the first glitch and increasing until their sum reaches 200fC. Also, in Figure 1(d), one gate is assumed to be a FO1 and the other one is FO2. As it can be seen from Figure 1, for a FO1 gate, the sum of the glitch sizes exceeds the single glitch size, when they result from the same overall collected charge, while it is smaller than the single pulse size for the FO2 gate for smaller collected charge values. For larger collected charge values, the curve **sum** converges to the curve **original**, as it can be seen in the figure.

Finally, when considering a transient fault induced by a specific event, it is important to define the range of glitch sizes that can occur due to those events. For example, for a radiation-induced transient fault in 130nm technology, it has been shown that the duration of a glitch lies in the interval from 30 to 300ps [8], with most glitches having the duration between 100 and 250ps [27]. In our work, we consider the cases of one, two or three simultaneous glitches and show the tradeoffs between the runtime needed to obtain the results, and the advantage that the
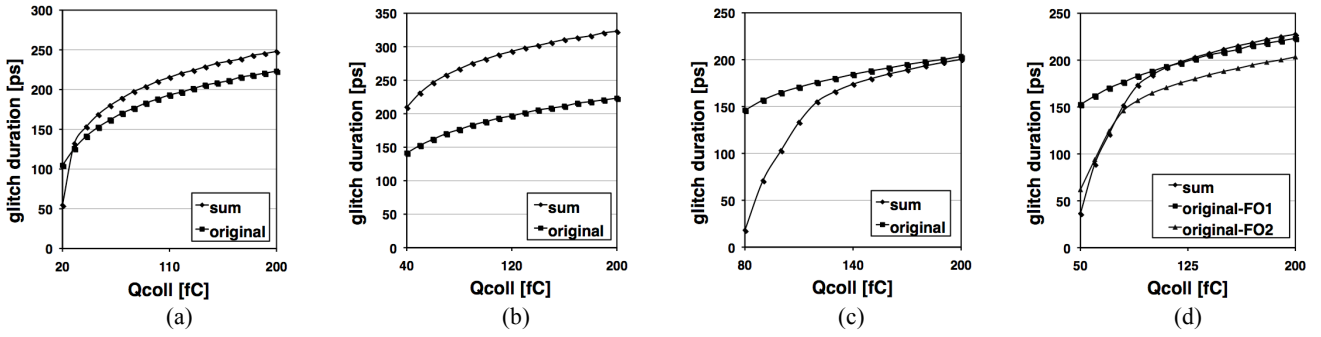
Figure 1. HSPICE simulation results for a NAND gate with different load/collected charge combinations: (a) FO1, first glitch 10fC; (b) FO1, first glitch 20fC; (c) FO2, first glitch 40fC; (d) first gate FO1, first glitch 20fC and second gate FO2.

more detailed analysis (that includes cases of three MTFs) provides.

B. Fault propagation

The three important masking factors affecting the propagation of a glitch through combinational circuit are:

- *logical masking* – occurs if the glitch arrives to the input of a gate when at least one of its other inputs has a controlling value;
- *electrical masking* – can attenuate or even completely mask the glitch that is not large enough compared to the delay of a gate through which the glitch propagates;
- *latching-window masking* – occurs when the glitch does not arrive on time at the input of the flip-flop to satisfy its setup and hold time conditions.

With technology scaling, the impact of these masking factors is decreasing [19][31], thus leading to the increased error rates in logic circuits, emphasizing the importance of modeling transient faults in logic circuits.

We have described in our previous work [19][20] that failing to treat these masking factors in a unified manner often leads to significant errors in the evaluation of circuit susceptibility to transient faults. More specifically, as it can be seen from results presented in [20], multiplying the probability of logical masking with the probability of electrical and latching-window masking (separately computed) leads to the error in the probability of latching the glitch, which can be as large as 3100%. By taking into account the joint dependence among the three masking factors, circuit topology, and input vectors, a unified treatment of the three masking factors also allows for accurate analysis of *reconvergent glitches*. For the purpose of modeling and analysis of MTF in this work, we will refer to glitches as reconvergent if they satisfy the following definition.

Definition 1: Pulses or glitches are called *reconvergent* if they arrive to different inputs of the same gate within the same clock cycle and satisfy one of the following:

- originate from a single gate;
- result from the same particle hit within a single clock cycle;
- result from different fault-induction mechanisms occurring within the same clock cycle.

Separate computation of different masking factors will incur an error, since it sums separately probabilities of sensitization of all reconvergent paths, probabilities of latching on all reconvergent paths, and then multiplies the two terms. This kind of approach cannot take into account the relative arrival time and durations of the glitches at the reconvergence point, as shown in [20].

IV. PROPOSED MTF MODEL

In this section, we describe our approach to modeling of multiple transient fault generation and their propagation through logic circuits. A practical implementation for STF modeling was described in [19] where, for a given circuit, a topologically sorted list of gates is generated first, and then, in one pass through the circuit, all possible glitches that can occur in the circuit are created and propagated to primary outputs. The MTF modeling methodology follows these two steps as well, but in addition to that, it also includes several new features, as presented in the rest of this section.

We first describe our approach to multiple fault generation, then we give an overview of the proposed multiple fault propagation modeling and finally, we provide a detailed evaluation of accuracy of the proposed glitch propagation model. The pseudo code of our algorithm and the glitch merging algorithm are presented in [21].

A. Fault generation implementation

There are several possible approaches to the modeling of an STF in terms of the details included in its model description. For example, simple models, like triangular or trapezoidal, include information about glitch duration and amplitude and, possibly, about the slope. On the other hand, there are approaches that use more accurate models, and consequently need more information about the glitch, like double-exponential current pulse [32]. However, there is a tradeoff between the accuracy of the glitch model and the time needed for a method based on such a model to estimate the impact of the glitch on circuit outputs. Therefore, in this work we use the former approach that assumes only glitch duration and amplitude as parameters. We provide a discussion on the accuracy and the efficiency (scalability) of our approach in Sections IV.C and VI.D, respectively.

Following the example from Section III.A, in case of radiation-induced single event MTFs (SE-MTFs), it is necessary to determine the set of gates that can be affected by a single particle hit. While in general the potential victims of a particle hit can be best determined by using physical design information, at logic level, in the absence of layout information, the two cases described in Section III.A (*i.e.*, a
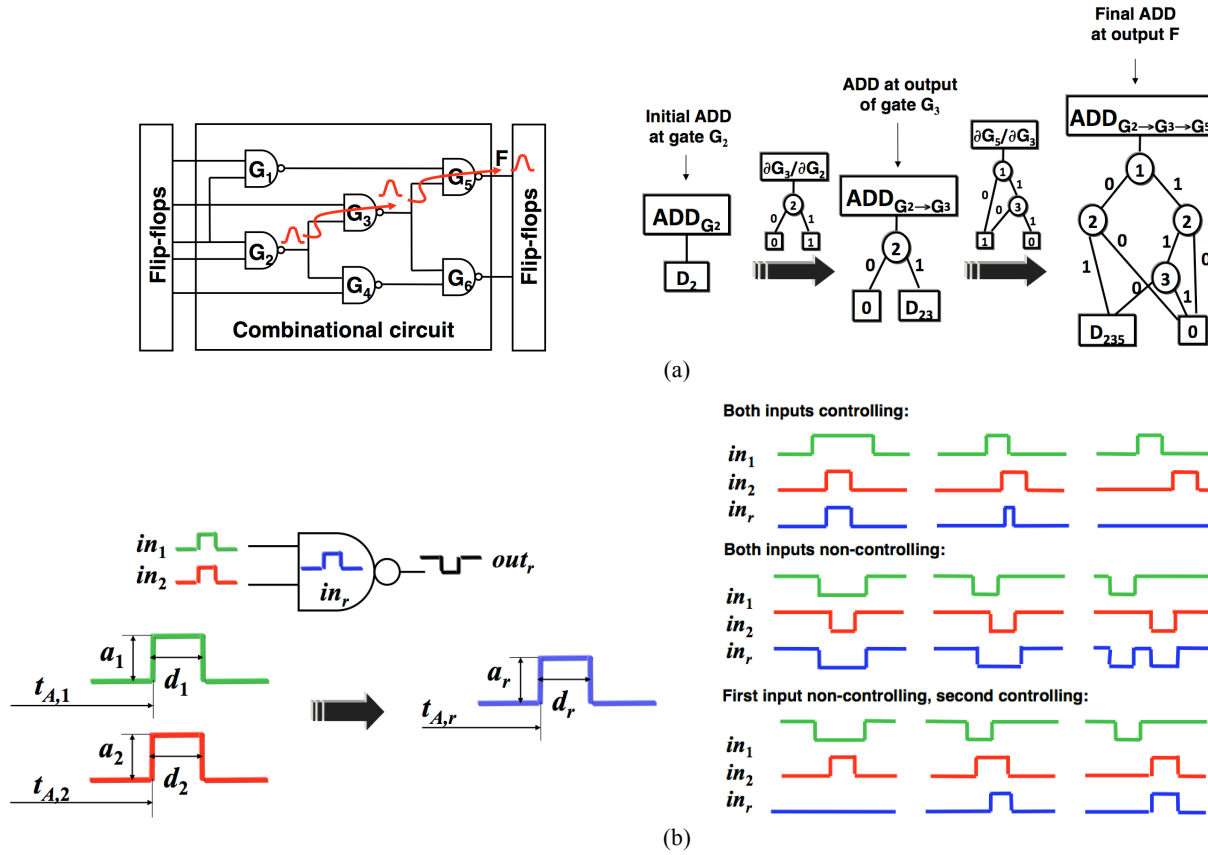
Figure 2. Glitch propagation modeling approach, proposed in [19] and used here as a basis for modeling multiple faults: (a) Symbolic modeling of glitch propagation by means of BDDs and ADDs. (b) Computation of the resulting glitch duration (d_r), amplitude (a_r), and arrival time ($t_{A,r}$), after the reconvergence of the two input glitches with given durations (d_1, d_2), amplitude (a_1, a_2) and arrival time ($t_{A,1}$, $t_{A,2}$).

gate and its fanin or fanout neighbors, or gates with common fanin or fanout neighbors) may be considered as good candidates for jointly being affected by SE-MTFs. Thus, for the radiation-induced MTF example, our framework takes as inputs the gate-level description of the circuit and from this description determines the fanin and fanout 'neighbor' relationship among gates. This does not affect the generality of the main algorithm (Figure 3), since layout information can easily be incorporated into the input circuit description.

Furthermore, if MTFs stemming from multiple events (ME-MTFs) are considered, different relation of interest (*i.e.*, not necessarily a 'neighbor' relation as we defined it in Section III.A) among pairs, triples, *etc.* of gates may be assumed. We assume that glitches are always generated at given internal gate output(s). This is a valid assumption in case of radiation-induced faults. In case of other fault mechanisms [2], the fault may not originate inside the gate, but instead may occur on the interconnect adjacent to the gates. In such cases, the framework can be applied by assuming that the glitch occurs at and is propagated from the gates abutting the victim wire. However, the analysis and the comparison of different transient fault sources is not the focus of this work, but we intend to look into those cases as well in the future.

Besides the shape and the originating location of the fault, one needs to also consider the temporal aspect of the generated fault. In other words, the time within the clock cycle (or, across two clock cycles) when the fault occurs may affect its shape, its

size and finally, its latching probability, once it is propagated to circuit outputs. We model and analyze transient faults occurring within a single clock cycle, assuming that a fault can occur at a given gate output only after the output becomes stable and can last only until the output receives a new value in the next cycle.

B. Fault propagation implementation

Since it has been shown that unified treatment of the three masking factors (logical, electrical and latching-window masking) is mandatory for highly accurate estimations, we use here the unified symbolic model that we proposed earlier in [19] (combinational circuits) and [20] (sequential circuits). The main idea of that approach is that the impact of the three masking factors can be modeled using BDDs and ADDs. This approach is explained in detail in [19][20] and, for the purpose of better understanding of the overall approach, we provide here a brief description. In Figure 2(a), we give an example of glitch duration ADDs and sensitization BDDs generated for benchmark circuit *C17*, assuming that a glitch originates at gate G_2 and propagates through gates G_3 and G_5 to primary output F. First, initial duration and amplitude ADDs (we show only duration ADDs in Figure 2(a), but amplitude ADDs are created similarly) are created, representing a glitch originating at a given gate G_2. Non-zero terminal nodes of ADDs represent duration (amplitude) of the glitch. Paths in ADDs that lead to non-zero terminal nodes represent input vectors that result in

those glitch durations (amplitudes), given initial glitch duration (amplitude) and input circuit parameters that determine the attenuation. Non-terminal nodes of BDDs and ADDs ("1", "2" and "3" in Figure 2(a)) represent primary inputs of the circuit. Next, a sensitization BDD, that represents the sensitization of output of gate G_3, with respect to output of gate G_2, in terms of primary inputs (non-terminal nodes) is computed. This sensitization BDD is applied to modify original glitch ADD and new ADD is created, representing the glitch at the output of gate G_3. Similarly, this new ADD is then modified using the corresponding sensitization BDD ($\partial G_5/\partial G_3$). The ADD computed for the glitch at the output of gate G_5 represents the duration (amplitude) of the glitch propagated from gate G_5 to primary output F.

This example shows the propagation of one glitch only. However, the important advantage of the proposed model is that it *concurrently* computes the propagation and the impact of transient faults originating at different internal gates of the circuit. This is made possible by assigning the originating gate identifier to the duration-amplitude ADD pair associated with each glitch. The concurrent computation of glitch propagation can be modified such that it accounts for multiple glitches occurring as a result of a single event or multiple events occurring within the same clock cycle. To be able to apply the same concurrent computation to multiple glitch propagation, we need to modify the modeling methodology as follows.

First, instead of assuming the occurrence of only one glitch at a time, it is necessary to keep track of several concurrent glitches. This requires that specific information for the gates at which glitches occurred is assigned to all duration-amplitude ADD pairs that correspond to glitches within a given MTF set. Thus, when compared to the STF case, the propagation of MTFs requires more information. The complexity of simple glitch propagation and attenuation is not much affected, due to the fact that the number of glitches that need to be considered during one pass through the circuit is $m \times N$, where N is the number of glitches considered in the STF case, and m is the number of different sizes for simultaneously occurring glitches. In other words, if, for example, we are analyzing the impact of two simultaneous glitches, at each gate we need to assume the occurrence of two different glitch sizes, and keep associated with each glitch a list of all MTF sets, to which a glitch of a given size and occurring at a given gate may belong.

Next, irrespective of the way they are generated, there are several possible cases of reconvergent glitches that can occur for both STFs and MTFs, as shown in Figure 2(b), on the NAND gate example. in_1 and in_2 in Figure 2(b) represent two reconvergent glitches at inputs of a NAND gate. in_r is the result of merging the two input glitches when logical masking is applied and represents the intermediate step when computing the output glitch (out_r). More specifically, two glitches that are to be merged may arrive at gate inputs carrying both controlling or both non-controlling values, or one carrying controlling and the other carrying non-controlling value. These different cases lead to different in_r glitches varying in their size and delay, compared to the original glitches, as shown in Figure 2(b). Finally, the glitch at the output, out_r, is found by applying electrical masking on the resulting input glitch, in_r. With respect to *Definition* 1, the number of cases that need to be considered for merging reconvergent glitches increases in the case of MTF. Besides glitches originating from a single gate and reconverging through different paths to inputs of another gate, it is necessary to consider all glitches belonging to a single MTF set, while their originating location is different. Once glitches are merged, it is necessary to represent the resulting glitch(es) as new glitch(es) with their specific MTF set lists. Therefore, this increases the number of glitches, but at the same time decreases the size of the MTF set of the original glitches that were merged.

C. Method accuracy and efficiency

When modeling transient faults and computing circuit susceptibility to these faults, the most accurate approach would be to simulate the circuit for all possible (single or multiple) fault sites and all possible input vectors. However, this approach fast becomes intractable when the number of inputs and the number of gates increase. Therefore, as already mentioned in Sections I and II, an approximate approach is necessary to provide estimates of output error rates, with a reasonable runtime. In other words, the main issue that the approximate models face is finding the best tradeoff between accuracy and efficiency. Furthermore, we already detailed in Sections II.A, III.B and IV.B, what are the important aspects of transient fault propagation that need to be included in a model, and how the approach we use satisfies these requirements.

To this end, from the perspective of transient fault model properties, our model is:

- approximate in modeling the shape of the glitch;
- exact in modeling logical masking;
- approximate in modeling electrical masking, that is, glitch attenuation when propagated through a logic gate;
- exhaustive in modeling all primary input vectors, in a sense that final ADDs at outputs include information about all possible input vectors;
- exhaustive in modeling all possible cases of reconvergent glitches;
- approximate for trains of glitches obtained as a result of glitch merging, representing them as one single glitch, thus providing conservative results via a worst case scenario;
- exact in the sense that it treats the three masking factors in a unified manner.

Therefore, the main sources of error in the model are the glitch shape assumptions, the attenuation model and some cases of reconvergent glitch merging. The attenuation model we used is explained in detail in our previous work [19]. We have also shown previously [19][20], by comparing with HSPICE simulation results that, for both combinational and sequential circuits the fault propagation method is within 4% accurate with up to 5000X speedup.

V. ERROR PROBABILITY COMPUTATION

The proposed framework allows for the analysis of a number of aspects of combinational or sequential circuit reliability. To estimate the susceptibility of these circuits to transient faults, the first step is to find the impact of an STF or an MTF set on a single output. Since ADDs that are propagated to circuit outputs or state flip-flops contain the information about circuit primary inputs, the symbolic approach provides an opportunity to also model and analyze the correlation among

output errors and state flip-flop errors. Therefore, we also present in this section modeling of multiple output and flip-flop upsets and provide upper and lower bounds for the circuit reliability.

A. Single Bit Upset (SBU) evaluation

To find the probability that an MTF, representing a set of glitches (1,2, …, n), originating at a set of gates (G_1, G_2, …, G_n) is latched at a given output F, for a given original particle hit, all possible values for the duration and amplitude of glitches arriving to the output F are found. There may be several corresponding glitches that have propagated to the output F from different gates. To this end, we define the following event:

E – an event that occurs when any of the glitches originating from one of the gates of the MTF set is latched at the output F.

More specifically, for each output F_j, where $j = 1,…,n_F$ and n_F is the overall number of primary outputs and next-state flip-flops (latter one in the case of sequential circuits), we can find the probability of failing due to an MTF with initial durations $\mathbf{d}_{init} = (d_{init,1}, d_{init,2}, …, d_{init,n})$ and initial amplitudes $\mathbf{a}_{init} = (a_{init,1}, a_{init,2}, …, a_{init,n})$ that originated at a given set of gates $\mathbf{G}^i = (G_1, G_2, …, G_n)$, where $i = 1,…,n_G$ and n_G is the number of all gate sets under consideration, as:

$$P(E_{i,j}^{\mathbf{d}_{init},\mathbf{a}_{init}}) = P(F_j\,fails\,|\,\mathbf{G}^i\,fails\cap glitches = (\mathbf{d}_{init},\mathbf{a}_{init})) \quad (3)$$

Similar to [19], we find the probability of event E by summing over all possible glitch durations, D_k, that occur at a given output and result from the propagation of glitches from a given gate set where MTF originated:

$$P(E_{i,j}^{\mathbf{d}_{init},\mathbf{a}_{init}}) = \sum_k \frac{D_k - (t_{setup} + t_{hold})}{T_{clk} - d_{init}} \cdot P(D = D_k) \quad (4)$$

where T_{clk} is the clock period, t_{setup} and t_{hold} are the setup and hold time of the latch, respectively, and d_{init} is the initial duration of the glitch. The probability that the duration of the glitch at the output is D_k is found from the ADD that represents a glitch belonging to a given MTF set. Depending on the assumptions and the approach taken in computing the overall output error probability, one can distinguish between two options for computing $P(E_{i,j}^{\mathbf{d}_{init},\mathbf{a}_{init}})$:

- *separate* output error probability, representing the probability of error at a given output, without taking into account possible correlated errors at other outputs;
- *exclusive* output error probability, representing the probability that an error due to a given MTF occurred at a given output with no other outputs affected.

One can find the exclusive output error probabilities by using their corresponding output ADDs. Since ADDs store the information about which primary input vectors lead to a glitch with duration D_k at a given output, the primary input vector values can be 'masked' in a given ADD by making the corresponding input variable combinations lead to 'zero' terminal nodes. Once these input vectors that lead to an MTF-induced error at any other circuit output are 'masked' in a given output ADD, for a given MTF, the *exclusive* output error probability can be computed as:

$$P(E_{i,j}^{\mathbf{d}_{init},\mathbf{a}_{init}})^{EX} = P\left(E_{i,j}^{\mathbf{d}_{init},\mathbf{a}_{init}}\,\middle|\,\bigcup_{J \neq j}(\neg E_{i,J}^{\mathbf{d}_{init},\mathbf{a}_{init}})\right) \quad (5)$$

In other words, in equation (5) we compute the probability of an MTF causing an error at a given output (j) only and not causing errors at other outputs ($J \neq j$), where $j, J = 1,…,n_F$.

We can now compute the *Mean Error Susceptibility* (*MES*) of a given output F_j, for a given assumed set of initial glitch durations and amplitudes, ($\mathbf{d}_{init}$, $\mathbf{a}_{init}$), as the average probability of output F_j failing due to all possible MTF sets that can occur in the circuit, given different input probability distributions:

$$MES(F_j^{\mathbf{d}_{init},\mathbf{a}_{init}}) = \frac{\sum_{k=1}^{n_f}\sum_{i=1}^{n_G} P(E_{i,j}^{\mathbf{d}_{init},\mathbf{a}_{init}})}{n_G \cdot n_f} \quad (6)$$

where n_G is the cardinality of the set of MTF gate sets of the circuit, $\{\mathbf{G}^i\}$, and n_f is the cardinality of the set of probability distributions, $\{f_m\}$, associated to the input vector stream.

B. Multiple Bit Upset (MBU) evaluation

While the previous discussion focused mainly on the occurrence of transient faults and their propagation through the combinational part of the circuit, in this section we describe our model for multiple output (or flip-flop) upsets in sequential circuits.

As described in Section V.A, the proposed approach allows for the computation of *separate* and *exclusive* output error probability. To this end, we can compute the probability of error not only for a single but also for two, three, or more outputs with correlated errors. For example, in the case of correlated errors for two given outputs F_{j1} and F_{j2}, we can find the *separate* probability of error stemming from the same MTF and occurring at least at one of the outputs as:

$$P(E_{i,j_1}^{\mathbf{d}_{init},\mathbf{a}_{init}} \cup E_{i,j_2}^{\mathbf{d}_{init},\mathbf{a}_{init}}) \quad (7)$$
$$= P(E_{i,j_1}^{\mathbf{d}_{init},\mathbf{a}_{init}}) + P(E_{i,j_2}^{\mathbf{d}_{init},\mathbf{a}_{init}}) - P(E_{i,j_1}^{\mathbf{d}_{init},\mathbf{a}_{init}} \cap E_{i,j_2}^{\mathbf{d}_{init},\mathbf{a}_{init}})$$

where the probability $P(E_{i,j_1}^{\mathbf{d}_{init},\mathbf{a}_{init}} \cap E_{i,j_2}^{\mathbf{d}_{init},\mathbf{a}_{init}})$ can be found by multiplying the two resulting glitch ADDs from outputs F_{j1} and F_{j2}, which will leave un-zeroed only those cases when both outputs are affected by the given MTF. By setting the un-zeroed terminal nodes to the value 'one', we create a mask to apply on each of the two final ADDs. However, this does not take into account possibly correlated errors at other outputs. To find the probability of error due to a given MTF occurring only at these two outputs (F_{j1} and F_{j2}), it is necessary to modify the corresponding ADDs as described in Section V.A, before applying equation (7).

For the case of state flip-flops, the main idea is to determine their impact on the final computed *error rate* when STFs or MTFs affect the state of more than one flip-flop and thus propagate through the circuit as multiple errors in the cycles following the cycle when the hit occurred.

The analysis of sequential circuits can be split into two main stages [20]: Stage I, representing the cycle when the hit occurs and Stage II, representing all the following cycles. In Stage I, it is necessary to include the impact of all three masking factors, while in Stage II, only logical masking is considered. Therefore, when final glitch duration and amplitude ADDs at primary outputs or next state lines are found in Stage I, it is possible to extract the information about the error correlations between different state lines. In other words, it is possible to find the probability of two or more next state lines failing due
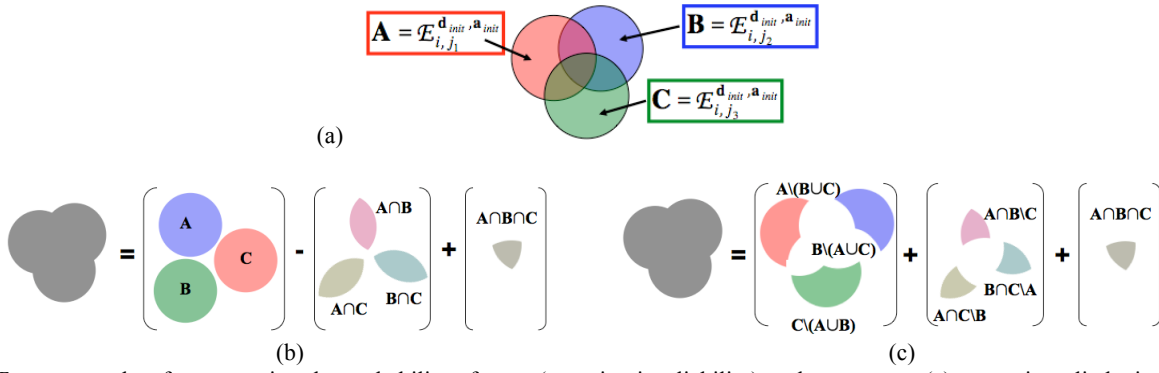
Figure 3. Two approaches for computing the probability of error (*i.e.*, circuit reliability) at three outputs (a), assuming glitch sizes ($\mathbf{d}_{init}$, $\mathbf{a}_{init}$) and faults at gate set $\mathbf{G}^i$, as in equation (11), approaching the estimation from below and using only exclusive output error probabilities (b) or, as in equation (12), oscillating around the correct value, while decreasing the error with each additional element in the sum, and using separate output error probabilities (c). For (c), if an odd number of elements from the sum is considered in equation (12), an upper bound of the exact value is obtained.

to an STF at a given gate (or, an MTF at a given set of gates). The computation of conditional probabilities in Stage II assumes multiple errors, which requires applying the model described above. The error probability at a given output in Stage II can then be found using the conditional probabilities computed in Stage II and state error probabilities in Stage I:

$$P(\mathsf{F}_j^{k,\mathbf{d}_{init},\mathbf{a}_{init}}) = \sum_l P(\mathsf{F}_j^k \mid \mathsf{F}_l^{1,\mathbf{d}_{init},\mathbf{a}_{init}}) \cdot P(\mathsf{F}_l^{1,\mathbf{d}_{init},\mathbf{a}_{init}})$$

$$+ \sum_{l_1} \sum_{l_2} P(\mathsf{F}_j^k \mid \mathsf{F}_{l_1}^{1,\mathbf{d}_{init},\mathbf{a}_{init}} \cap \mathsf{F}_{l_2}^{1,\mathbf{d}_{init},\mathbf{a}_{init}}) \cdot P(\mathsf{F}_{l_1}^{1,\mathbf{d}_{init},\mathbf{a}_{init}} \cap \mathsf{F}_{l_2}^{1,\mathbf{d}_{init},\mathbf{a}_{init}})$$

$$+ ... + \sum_{l_1} \sum_{l_2} ... \sum_{l_{n_s}} P(\mathsf{F}_j^k \mid \bigcap_{l_i} \mathsf{F}_{l_i}^{1,\mathbf{d}_{init},\mathbf{a}_{init}}) \cdot P(\bigcap_{l_i} \mathsf{F}_{l_i}^{1,\mathbf{d}_{init},\mathbf{a}_{init}})$$

(8)

where $P(\mathsf{F}_j^{k,\mathbf{d}_{init},\mathbf{a}_{init}})$ is the probability of output j at the sub-stage k failing, given an initial glitch duration and amplitude sets, $\mathbf{d}_{init}$ and $\mathbf{a}_{init}$. $P(\mathsf{F}_j^k \mid \mathsf{F}_l^{1,\mathbf{d}_{init},\mathbf{a}_{init}})$ is the probability of error at the output j at the stage k, given that an error was latched at the state line l **only** at Stage I, with the probability of error at state line l given by:

$$P(\mathsf{F}_l^{1,\mathbf{d}_{init},\mathbf{a}_{init}}) = \frac{\sum_{i=1}^{n_\mathsf{G}} P(\mathcal{E}_{i,l}^{1,\mathbf{d}_{init},\mathbf{a}_{init}})}{n_\mathsf{G}}$$

(9)

Each probability $P(\mathcal{E}_{i,l}^{1,\mathbf{d}_{init},\mathbf{a}_{init}})$ in equation (9) represents the exclusive error probability for an individual output (*i.e.*, next state line). Similarly, the probability of error for pairs, triples, *etc*. of next state lines in Stage I can be found using the computation of exclusive error probabilities (*e.g.*, exclusive $P(\mathcal{E}_{i,l_1}^{1,\mathbf{d}_{init},\mathbf{a}_{init}} \cap \mathcal{E}_{i,l_2}^{1,\mathbf{d}_{init},\mathbf{a}_{init}})$ for the pair of outputs or next state lines F_{l1} and F_{l2}).

C. Bounds on circuit error probability

The overall probability of output F_j failing, $P(\mathsf{F}_j)$, due to different MTF sets, with different initial glitch durations and amplitudes and for different input vector probability distributions, can be defined using the *MES* metric. Assuming a uniform distribution of duration-amplitude pairs (d,a) along the surface $S = (d_{max} - d_{min}) \cdot (a_{max} - a_{min})$ for individual glitches, by partitioning the surface of each glitch from the MTF set into sub-surfaces (as described in [19]), one can find the probability $P(\mathsf{F}_j)$ as the weighted average of an *MES* across all combinations of MTF element sub-surfaces (S_n):

$$P(\mathsf{F}_j) =$$

$$\frac{1}{S_n} \sum_{l_n=1}^{n_{l_n}} \sum_{m_n=1}^{n_{m_n}} \left(... \frac{1}{S_1} \sum_{l_1=1}^{n_{l_1}} \sum_{m_1=1}^{n_{m_1}} \left(MES(\mathsf{F}_j^{(d_{l_1},d_{l_2},...,d_{l_n})(a_{m_1},a_{m_2},...,a_{m_n})}) \cdot \Delta d_1 \cdot \Delta a_1 \right) \right)$$

$$\cdot ... \cdot \Delta d_n \cdot \Delta a_n$$

(10)

where, for each $i=1,..,n$:

$$d_{l_i} = d_{min} + l_i \cdot \Delta d_i \quad \text{and} \quad d_{max} = d_{min} + n_{l_i} \cdot \Delta d_i$$

$$a_{m_i} = a_{min} + m_i \cdot \Delta a_i \quad \text{and} \quad a_{max} = a_{min} + n_{m_i} \cdot \Delta a_i.$$

Without loss of generality, in equation (10), we assume that all MTF glitch size combinations have equal probability of occurrence. It is, however, straightforward to extend it to the case of different MTF probabilities.

To find the overall circuit error susceptibility, one can average across all output error probabilities and find the maximum and minimum output error susceptibility. In addition, one can determine upper and lower bounds for circuit error probability. To accurately compute the circuit reliability (R) one needs to follow one of the following approaches:

1. Reliability as a function of exclusive probabilities:

$$R = 1 - \left(\sum_j P(\mathsf{F}_j) + + \sum_{j_1=1}^{n_\mathsf{F}-1} \sum_{j_2=j_1+1}^{n_\mathsf{F}} P(\mathsf{F}_{j_1} \cap \mathsf{F}_{j_2}) \right.$$

$$\left. + ... + \sum_{j_1} \sum_{j_2} ... \sum_{j_{n_\mathsf{F}}} P(\bigcap_{j_i,i=1..n_\mathsf{F}} \mathsf{F}_{j_i}) \right)$$

(11)

where n_F is the number of primary circuit outputs. Each probability $P(.)$ on the right hand side of the equation (11) is the *exclusive* probability, that is, it is found using the exclusive ADDs, or

2. Reliability as a function of separate probabilities:

$$R = 1 - \left(\sum_j P(\mathsf{F}_j) - \sum_{j_1=1}^{n_\mathsf{F}-1} \sum_{j_2=j_1+1}^{n_\mathsf{F}} P(\mathsf{F}_{j_1} \cap \mathsf{F}_{j_2}) \right.$$

$$\left. + \sum_{j_1} \sum_{j_2} \sum_{j_3} P(\mathsf{F}_{j_1} \cap \mathsf{F}_{j_2} \cap \mathsf{F}_{j_3}) - ... + (-1)^{n_\mathsf{F}-1} \sum_{j_1} \sum_{j_2} ... \sum_{j_{n_\mathsf{F}}} P(\bigcap_{j_i,i=1..n_\mathsf{F}} \mathsf{F}_{j_i}) \right)$$

(12)

where each probability $P(.)$ on the right hand side of the equation (12) is a *separate* probability, that is, it is computed using original ADDs, without taking into account possible correlated errors at other outputs.

Since the computation of all probability values in equations (11) and (12) may be impractical for circuits with large number

TABLE I

Benchmarks used in experiments, number of gates, inputs, outputs and next state lines.

| Bench. — # of	S27	S208	S298	S444	S526	S1196	S1238	z4ml	5xp1	9symml	C880
gates	10	68	86	153	165	487	540	45	116	145	315
inputs	4	10	3	3	3	14	14	7	7	9	60
outputs	1	1	14	2	21	13	13	4	10	1	26
next-st.	3	8	4	21	21	18	18	0	0	0	0

Figure 4. Individual output *SER* for different benchmarks, when two simultaneous faults (MTF2, MBU1), two correlated output errors (MTF1, MBU2) or an STF are assumed at Stage I.

of outputs, we restrict ourselves to finding the upper and lower bounds for circuit error probability, and hence, for circuit reliability. To do so, we find *exclusive* and *separate* probabilities from equations (11) and (12) for the cases of: (*i*) single outputs; (*ii*) pairs of outputs with highest correlation; and (*iii*) triples of outputs with highest correlation. By doing so, equation (11) will provide an upper bound on circuit reliability, while equation (12) will provide a lower bound. The basic idea behind computing the error probability bounds is presented in Figure 3.

Similar to the overall circuit analysis, it is possible to apply the same reasoning for computing the next state line error probability in Stage I of unrolled sequential circuit (described in Section V.B), and obtain the lower and upper bound on output error probabilities in Stage II of the unrolled sequential circuit.

D. Example: Soft error rate computation

One of the important aspects of the proposed modeling framework is that it is independent of the transient fault source, as well as the circuit implementation, as long as the function of the circuit and the propagation of the glitch can be described using BDDs and ADDs. Furthermore, the final ADDs created for individual gate-output pairs are free of any fault-specific information and include only information about circuit topology and technology node parameters. They can be used to compute output and circuit error probabilities when more specific information about circuit inputs is provided. Finally, the inclusion of transient-fault origin specific parameters as external inputs to the framework allows for computation of error rates.

For example, in case of radiation-induced soft errors, the *SER* can be found by using the output error probabilities from equation (10) as:

TABLE II

Algorithm runtime in seconds, for two benchmarks circuits for the combinations of following cases: a single transient fault (SE-STF), two or three simultaneous transient faults (SE-MTF2 or SE-MTF3), single bit (output or flip-flop) upset (SBU), two or three correlated bit upsets (MBU2 and MBU3).
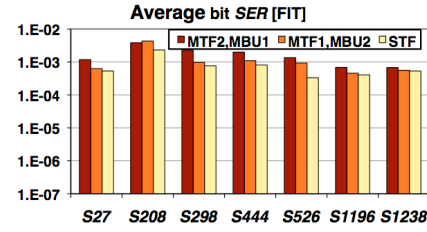
$$SER_{F_j} = P(\mathsf{F}_j) \cdot R_{eff} \cdot R_{PH} \cdot A_{circuit} \qquad (13)$$

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.

As an example of how the proposed framework can be applied to transient faults, we use the soft error specific parameters defined above to compute the error rates, as shown in the next section.

VI. EXPERIMENTAL RESULTS

In this section, we present the results obtained using the proposed framework when using the Berkeley Predictive Technology Model (PTM) [35] in conjunction with benchmark circuits from *ISCAS'89* and *mcnc'91* suites. Number of gates, number of inputs, number of outputs and next-state lines are listed in TABLE I. The proposed framework is implemented in C++, and run on a 3GHz Pentium 4 workstation running Linux.

In TABLE II, we report the runtime of our framework for two benchmarks, *S27* and *S208*, and compare the times for the following cases: single transient fault (STF), two simultaneous transients stemming from a single hit (SE-MTF2) or three simultaneous transients stemming from a single hit (SE-MTF3), when a single bit upset occurs (SBU), two correlated bit upsets (MBU2), or three correlated bit upsets occur (MBU3). The benchmarks are run assuming two unrolled sub-stages, one input vector probability distribution and computation of both separate and exclusive probabilities. While the runtime increases with the increase in number of simultaneous TFs, there is more significant increase in the runtime with the increase in the number of MBUs. This is due to the fact that, while MTFs lead to a larger number of ADDs being propagated through the topologically ordered circuit, MBUs require more operations on final ADDs and a traversal of a larger number of ADDs to compute final probabilities. The runtime can be significantly reduced if separate probabilities are computed or if exclusive probabilities are computed on the pairs (triples) of outputs that have highest probability of being simultaneously affected.

A. Effect of Single vs. Multiple Faults

We show in Figure 4 the *SER* results for double glitches and double flip-flop upsets. All results in Figure 4 are obtained using 65nm technology node. The *SER* values for two simultaneous TFs are obtained by averaging across several glitch size combinations (80ps and 60ps, 80ps and 40ps, 60ps and 40ps). The *SER* values for two MBUs are averaged across different initial STF glitch sizes (100ps, 80ps, 60ps and 40ps)
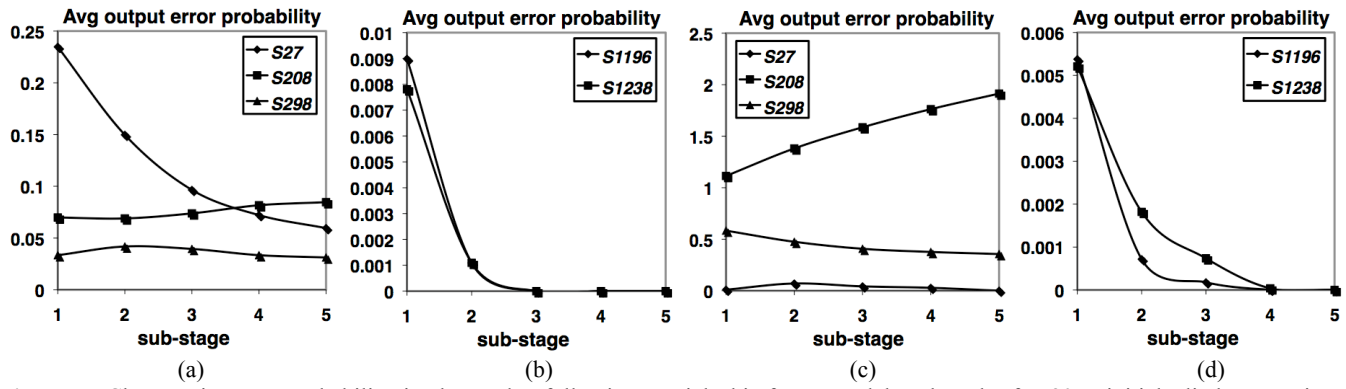
Bench. \ Mode	STF SBU	STF MBU2	STF MBU3	MTF2 SBU	MTF2 MBU2	MTF2 MBU3	MTF3 SBU	MTF3 MBU2	MTF3 MBU3
S27	0.1	0.2	0.3	0.2	0.3	0.4	1.6	1.9	2.1
S208	2.1	46.8	8491	6.3	71	10033	4427	16260	44220

Figure 5. Changes in error probability in the cycles following particle hit for several benchmarks for 80ps initial glitch, assuming two simultaneous TFs (a),(b), or two correlated MBUs (c)(d) at Stage I.

and the output probability values are obtained by summing over all MBU pair combinations that can occur in Stage I. As it can be seen, the impact of multiple-event transients (MTFs) and multiple-state line errors (MBUs) varies across different circuits. As expected, STFs lead to the smallest *SER*. The ratio between the impact of a multiple TF and the impact of multiple flip-flop upsets after the first stage varies. Depending on the circuit topology, in some cases (*S1196* and *S1238*) the impact of multiple flip-flop upsets in Stage II is small, while in other cases (*S208*) is more significant.

In Figure 5, we show the changes in average output error probability for several benchmarks in cycles following the particle hit, assuming two simultaneous MTFs with initial glitch durations of 80ps and 40ps, Figure 5(a)-(b), or two correlated MBUs in Stage I, Figure 5(c)-(d). The results again show that different circuits behave differently with respect to multiple faults or multiple flip-flop upsets. The probability of error at the output, in the cycles after the particle hit, can follow all three trends: decrease rapidly, remain at about the same level, or increase.

1) Impact on circuit outputs

In Figure 6, we show changes in the average, minimum, and maximum probability of error at individual outputs for the benchmark circuit *S1238*, when double (MTF2) or triple (MTF3) simultaneous transient faults are assumed, instead of a single fault (STF) (Figure 6(a)-(c))[1]. For the STF case, we assume a 110ps initial glitch duration, for MTF2 case, we assume a set of 110ps- and 70ps-long glitches, and for the case of MTF3, we assume 110ps-, 70ps- and 30ps-long glitches. We also present the number of TF sets that affect each output (Figure 6(d)). As it can be seen from results presented in Figure 6(a), there is no clear correlation between the increased number of simultaneous faults and the average output error probability. However, maximum error probability increases significantly (Figure 6(c)) from the STF to the MTF3 case. We can also see differences in the error susceptibility across different outputs. For example, the number of transient faults affecting output **6** is smaller than the number of transients affecting other outputs, remaining almost the same in all three cases (STF, MTF2 and MTF3). On the other hand, output **6** has much higher error susceptibility compared to other outputs in those three cases.

[1] This is computed only across TFs affecting output, not all TFs that can occur in the circuit.

This is due to the fact that transient faults affecting this particular output stem only from the nearest input neighbors of the gate driving the output and the gate itself, and thus, when averaged across a small number of TFs with high impact, we obtain a high error probability.

2) Impact on state F/Fs

We also present in Figure 6 (output bits 14-31) state F/Fs and changes in their error probabilities when a double or triple MTF is assumed instead of an STF. As seen in Figure 6, F/Fs in circuit *S1238* are on average more susceptible to transient faults than primary outputs. This fact is especially important when analyzing the circuit response to TFs in cycles following the cycle when the TF(s) occurred. The overall probability of error in those cycles depends on the probability of single or multiple upsets at F/Fs resulting from the fault in the cycle when it occurred, and the probability that these upsets will affect circuit outputs in the following cycles [20].

3) Gate error impact

We present in Figure 7 the impact of individual TF sets on primary outputs and state flip-flops in circuit *S1238*, for the three cases: STF, double MTF (MTF2) and triple MTF (MTF3). In all three cases, most of the TF sets have error impact smaller than 0.1. While in the STF case error probabilities range up to slightly above 0.35, in the MTF2 and MTF3 cases there is a number of TF sets that, if occur, have a high probability of affecting outputs. Similarly, in the latter two cases, there are also a number of TF sets (that is, sets of gates where faults occur simultaneously) that can affect a large number of outputs. By using the proposed framework, one can identify these TF sets (gates) with highest impact, and thus obtain valuable information when choosing circuit optimization techniques.

B. Bounds on circuit reliability

For the circuits with large number of outputs and/or state lines, it is inefficient to compute all terms of the sums in equations (11) and (12). Instead, one can compute separate and exclusive probabilities of output errors for each output first, then for pairs or triples of outputs, and find the lower and upper bound on error probability. We have estimated these bounds, by computing the first three terms of the sum on the right hand side of equations (11) and (12). Results for benchmark circuit *S27* are presented in Figure 8(a).
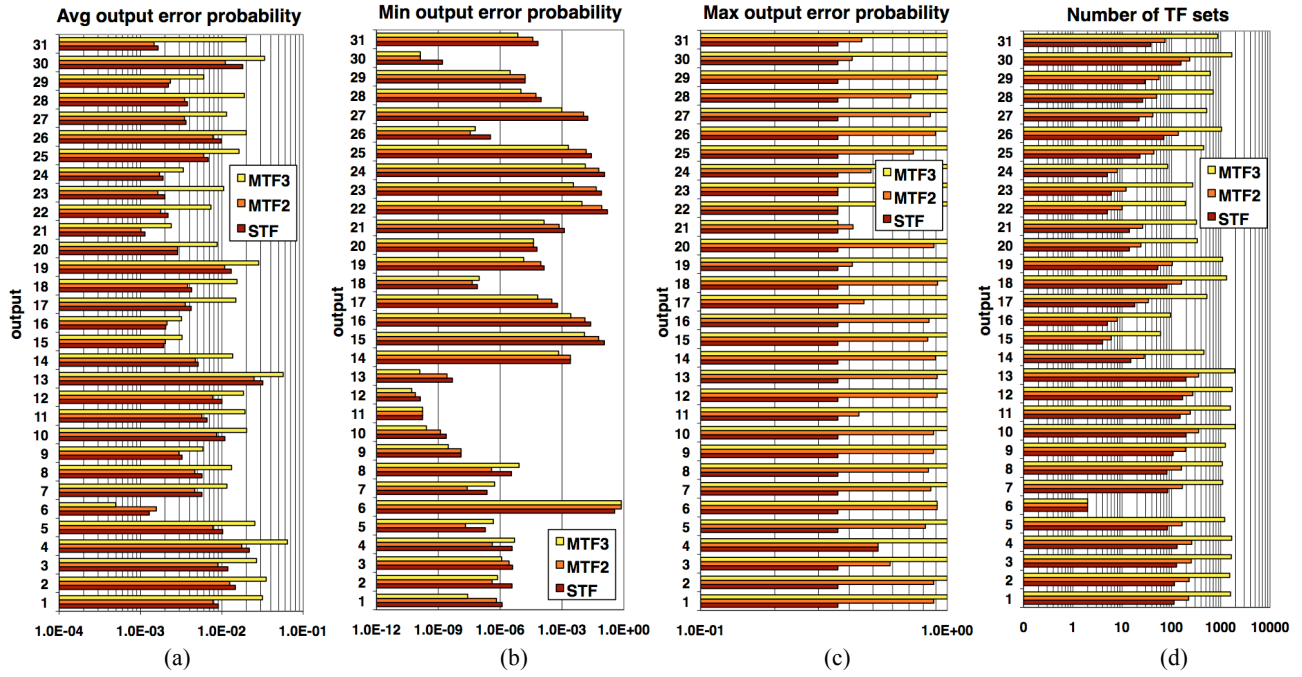
Figure 6. (a) Average, (b) minimum, (c) maximum output error probability, and (d) number of transient faults affecting output, for circuit *S1238*, when single (STF), double (MTF2) or triple (MTF3) transient faults are assumed. Outputs 1-13 are primary outputs, 14-31 are flip-flops.

As it can be seen, by only including the first element in the sum in equation (12), we obtain circuit reliability values that are very close to the lower reliability bound, but save on computation time (as presented in TABLE II, SBU case vs. MBU case). On the other hand, the lower bound estimation is less computationally expensive then the estimation of the upper bound (equation (12) vs. equation (11), MBU case). This is due to the fact that for the upper bound, one needs to determine exclusive probabilities, which require more BDD and ADD handling for masking all cases when other outputs are affected by the same TF set. Lower bounds are conservative, as they overestimate error probability, and thus underestimate circuit reliability.

Figure 8(a). also presents results for the average output reliability (as computed in [19][20]). The average output reliability provides useful information about what one can expect in terms of output robustness to transient faults in a given circuit. On the other hand, in addition to this information, circuit reliability bounds show what is the expected interval of the overall circuit reliability.

C. Impact of individual masking factors

In Figure 8(b), we show results for the impact of individual masking factors. First, in Figure 8(b)(left) the masking factor impact, *i.e.*, the sensitivity of a set of benchmark circuits to the individual masking factors is presented. We also present in Figure 8(b)(right) differences in error probabilities when setup and hold time windows change within same technology node (65nm).

We define *masking impact* (*MI*) of a given masking factor, *factor* $\in$ {logical masking, electrical+latching-window masking}, as:

$$MI_{factor} = \frac{(R_{all} - R_{other})}{R_{all}} \qquad (14)$$

where R_{all} is the reliability computed as in (12), when all masking factors are included, and R_{other} is the reliability computed when all factors except *factor* are included.

The results in Figure 8(b)(left) show that individual masking factors contribute differently to the circuit reliability for different circuits. For example, in case of circuit *9symml*, logical masking (**log** bar) has much more contribution than the other two masking factors (**el+lw** bar). For other circuits in Figure 8(b)(left) (*5xp1*, *S27* and *z4ml*), the impact on the average output reliability of logical masking, on one side, and electrical and latching-window masking, on the other, is almost the same, with logical masking having slightly less impact in circuit *S27*.

As shown in Figure 8(b)(right), the impact of latching-window masking may also vary across different circuits. These results can be affected by the initial size of the glitch, and the logical and electrical masking effect in the circuit. As can be seen from Figure 8(b)(right), the increase in the size of latching-window did not affect much benchmarks *5xp1*, *s27* and *z4ml*. This can be explained by the fact that the reliability of circuits *5xp1*, *s27* and *z4ml* is already high for the initial size we used in experiments (110ps). Since reliability can have values from 0 to 1, it does increase, but slowly for those circuits. However, in case of circuit *9symml*, the reliability is initially very small, and thus latching-window size has more impact on it.

These results provide important insight into the optimization techniques and hardening of circuits. We can draw conclusions about which parts of a specific circuit contribute more to transient fault masking, and which masking factor has more impact on fault propagation for a given circuit. Thus, based on this information, one can decide which techniques, or combination thereof should be used to obtain best results. For
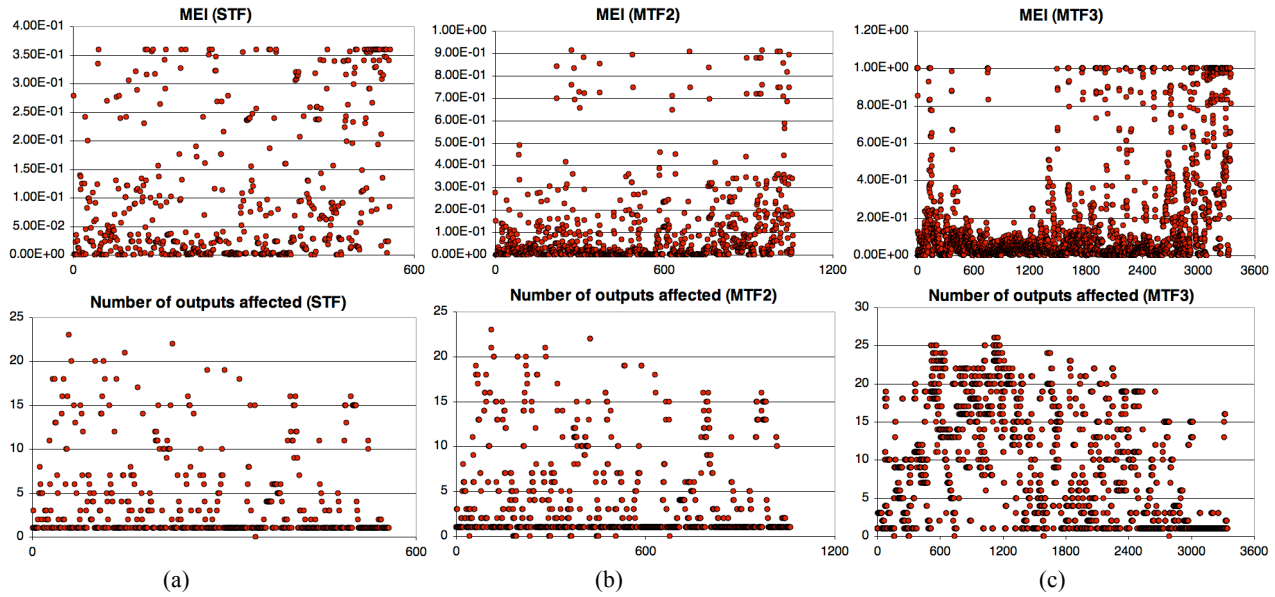
Figure 7. Average error impact (top three charts) and the number of outputs affected (bottom three charts) for each individual TF set (x-axes represent TF set indices) in circuit *S1238* for the following three cases and TF set source location counts: (a) single (STF) case and all 538 gates, (b) double (MTF2) case and 1059 gate pairs under consideration, as described in IV.A, and (c) triple (MTF3) case and 3345 gate triplets under consideration.

example, in case of circuit *9symml*, improving electrical masking can lead to significant improvement in error rates.

With the inclusion of power and performance data, the presented model can be incorporated into circuit design tools and can provide power, performance, cost and reliability tradeoffs for different circuit implementations in earlier design stages.

D. Discussion on method scalability

When BDDs or ADDs are used to represent Boolean functions, there exist some limitations in terms of the complexity of algorithms used and the size of diagrams that are created during program runtime. These limitations can be overcome, to some extent, by finding compact representation of functions with the best variable ordering, and by optimized implementation of algorithms. For many applications, these steps are enough to efficiently obtain necessary results. However, in some cases (*e.g.*, integer multiplication) even the best case is exponential in number of variables, thus requiring a different approach to enable scalability. One possible approach to improve the scalability of the proposed framework is circuit partitioning.

The idea behind circuit partitioning is to partition gates of the circuit among several regions, such that, for example, the number of gates allowed in each partition is below a certain limit and/or the number of nets crossing the cuts between partitions is minimized. For the purpose of this work, it is important that the partitions have the smallest possible number of inputs, since the size and the manipulation of BDDs/ADDs depends on this number. A very efficient partitioning tool that can be used for these purposes is a hypergraph partitioning package, hMetis [12].

Once a given circuit is partitioned, the goal is to apply the framework to each partition separately, instead of the circuit as a whole, in order to avoid the possible explosion of BDD/ADD size that could lead to long run times or memory issues.

However, while allowing for partitions to be run separately, it is necessary to find the mechanism to combine the results from those partitions, without a significant loss in accuracy (for example, by using the output statistics for one partition as input vector statistics for another partition they feed into). We present the flow of applied partitioning algorithm in Figure 9(a).

In order to show the applicability of the partitioning approach on our framework, we used benchmark circuit *C880*. For a given STF, given initial glitch size (110ps) and given input vector distribution, the runtime of our framework for computing all output error probabilities (exclusive and separate) of circuit *C880*, is around 4000s. On the other hand, the runtimes for two partitions of circuit *C880*, obtained using hMetis, are 68s and 7s (after the computation of output vector statistics for each partition, which takes significantly less time than the main algorithm). Average output error probability for a single circuit is 0.219, while output error probability obtained by averaging across all output error probabilities obtained for the two partitions is 0.212. As we present in Figure 9(b), out of 26 outputs, when partitioning is used, the error probability is exactly the same as the one obtained in the case of original (non-partitioned) circuit for 14 outputs, it is overestimated for four outputs (with a single outlier) and underestimated for eight outputs. The average estimation error per output, when partitions are used instead of the original circuit, is 0.5% (overestimation). This estimation error can be further reduced if a weighting mechanism is used for faults assumed to be propagated from other partition(s), depending on whether they stem from more than one gate or not.

We also conducted experiments on benchmark circuit *C6288*, with 2972 gates, 32 inputs and 32 outputs. The circuit *C6288* was partitioned using hMetis [12] into 8 smaller circuits and analyzed in tens of minutes, for the largest partition. On the other hand, HSPICE would require approximately 10^8 minutes or about 200 years for the same partition and the same simulation scenario, i.e., taking into account all possible fault
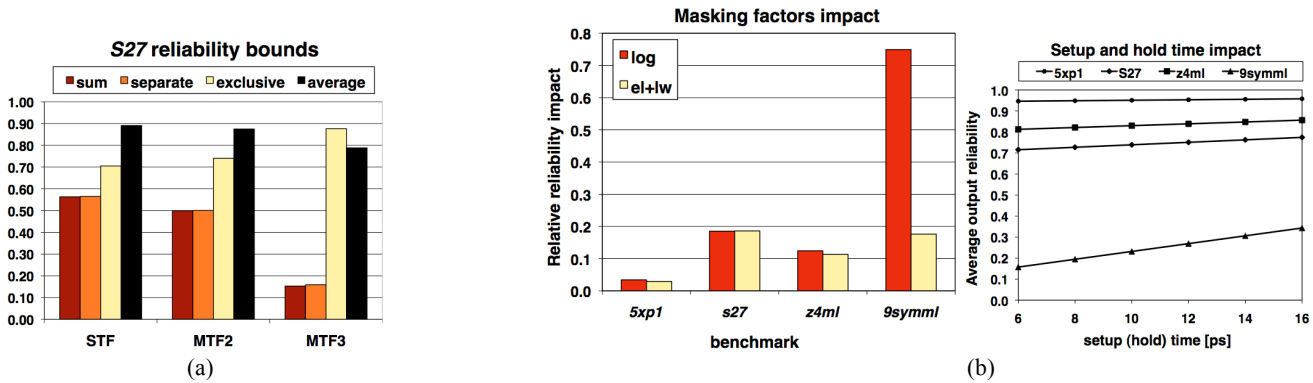
Figure 8. (a) Lower and upper bound for circuit reliability (R), when only the first element of the sum in equation (12) is computed (**sum**), or when the first three elements of the sum in equations (12) (**separate**) and (11) (**exclusive**) are computed. We also present the case when output reliability is averaged across all outputs in the circuit (**average**). (b) Impact of different masking factors on circuit reliability: logical masking impact vs. electrical and latching-window masking impact, computed as in equation (14) (left), and changes in reliability with the increase in setup and hold time (latching-window) (right).

sites and all possible input vectors. Furthermore, a single run of the framework allows for the computation of all final gate-output ADDs, and both separate and exclusive output error probabilities (Section V.A). Computing only separate output probabilities will decrease the runtime, in some cases by half. Furthermore, finding the best partition could decrease the runtime for all partitions to less than a few tens of minutes, but that analysis is beyond the scope of this paper.

VII. CONCLUSION

In this work, a probabilistic symbolic modeling methodology for efficient and accurate estimation of the circuit's reliability is proposed, when single or multiple transient faults occur within the circuit. The main idea behind the proposed work is to enable the analysis of the susceptibility of individual outputs to errors stemming from single and multiple transient faults, as well as finding the parts of the circuit (gates or gate clusters) that have the highest impact on circuit reliability. This kind of analysis can guide circuit designers in early design stages and allow for design tradeoff evaluation. We have demonstrated the efficiency of our method by applying it on a subset of *ISCAS'89* and *mcnc'91* benchmarks of various complexities.

VIII. REFERENCES

[1] H. Asadi and M. B. Tahoori, "Soft Error Derating Computation in Sequential Circuits," in *Proc. of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 497-501, November 2006.
[2] D. Atienza, G. De Micheli, L. Benini, J. L. Ayala, P. G. Del Valle, M. DeBole and V. Narayanan, "Reliability-Aware Design for Nanometer-Scale Devices," in *Proc. of Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 549-554, January 2008.
[3] D. Bhaduri, S. Shukla, P. Graham and M. Gokhale, "Scalable techniques and tools for reliability analysis of large circuits," in Proc. Of International Conference on VLSI Design (VLSID), pp. 705-710,2007.
[4] S. Borkar, "Thousand Core Chips – A Technology Perspective," in *Proc. of Design Automation Conference (DAC)*, pp. 746-749, June 2007.
[5] M. R. Choudhury and K. Mohanram, "Accurate and Scalable Reliability Analysis of Logic Circuits," in *Proc. of Design, Automation and Test in Europe (DATE)*, pp. 1454-1459, 2007.
[6] M. R. Choudhury and K. Mohanram, "Reliability Analysis of Logic Circuits," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, Vol. 28, No. 3, March 2009.

[7] Y. S. Dhillon, A. U. Diril, and A. Chatterjee, "Soft-Error Tolerance Analysis and Optimization of Nanometer Circuits," in *Proc. of Design, Automation and Test in Europe (DATE)*, pp. 288-293, March 2005.
[8] P. E. Dodd, "Physics-Based Simulation of Single-Event Effects," in *IEEE Transactions on Device and Materials Reliability*, Vol. 5, No. 3, pp. 343-357, September 2005.
[9] B. S. Gill, C. Papachristou, F. G. Wolf and N. Seifert, "Node Sensitivity Analysis for Soft Errors in CMOS Logic," in *Proc. of International Test Conference (ITC)*, November 2005.
[10] J. P. Hayes, I. Polian and B. Becker, "An Analysis Framework for Transient-Error Tolerance," in *Proc. of VLSI Test Symposium (VTS)*, pp. 249-255, May 2007.
[11] D. Holcomb, W. Li and S. E. Seshia, "Design as You See FIT: System-Level Soft Error Analysis of Sequential Circuits," in *Proc. of Design, Automation and Test in Europe (DATE)*, pp. 785-790, April 2009.
[12] G. Karypis and V. Kumar, "hMetis: A Hypergraph Partitioning Package," http://www.cs.umn.edu/~karypis.
[13] A. KleinOsowski, E. H. Cannon, P. Oldiges and L. Wissel, "Circuit design and modeling for soft errors," in *IBM Journal of Research and Development*, Vol. 52, No. 3, pp. 255-263, May 2008.
[14] S. Krishnaswamy, G. F. Viamontes, I. L. Markov and J. P. Hayes, "Probabilistic Transfer Matrices in Symbolic Reliability Analysis of Logic Circuits," in *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, Vol. 13, No. 1, January 2008.
[15] S. Krishnaswamy, I. L. Markov and J. P. Hayes, "When are multiple gate errors significant in logic circuits?" in *Proc. of the 2nd Workshop on System Effects of Logic Soft Errors (SELSE-2)*, April 2006.
[16] S. Krishnaswamy, G. F. Viamontes, I. L. Markov and J. P. Hayes, "Accurate Reliability Evaluation and Enhancement via Probabilistic Transfer Matrices," in *Proc. of Design, Automation and Test in Europe (DATE)*, pp. 282 - 287, March 2005.
[17] R. C. Martin and N. M. Ghoniem, "The size effect of ion charge tracks on single-event multiple-bit upset," in *IEEE Transactions on Nuclear Science*, Vol. NS-34, No. 6, pp. 1305-1309, December 1987.
[18] T. Merelle, F. Saigne, B. Sagnes, G. Gasiot, Ph. Roche, T. Carriere, M.-C. Palau, F. Wrobel and J.-M. Palau, "Monte-Carlo Simulations to Quantify Neutron-Induced Multiple Bit Upsets in Advanced SRAMs," in *IEEE Transactions on Nuclear Science*, Vol. 52, No. 5, pp. 1538-1544, October 2005.
[19] 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.
[20] N. Miskov-Zivanov, D. Marculescu, "Soft Error Rate Analysis for Sequential Circuits," in *Proc. of Design, Automation and Test in Europe (DATE)*, pp. 1436-1441, April 2007.
[21] N. Miskov-Zivanov and D. Marculescu, "A Systematic Approach to Modeling and Analysis of Transient Faults in Logic Circuits," in *Proc. of IEEE International Symposium on Quality Electronic Design (ISQED)*, March 2008.
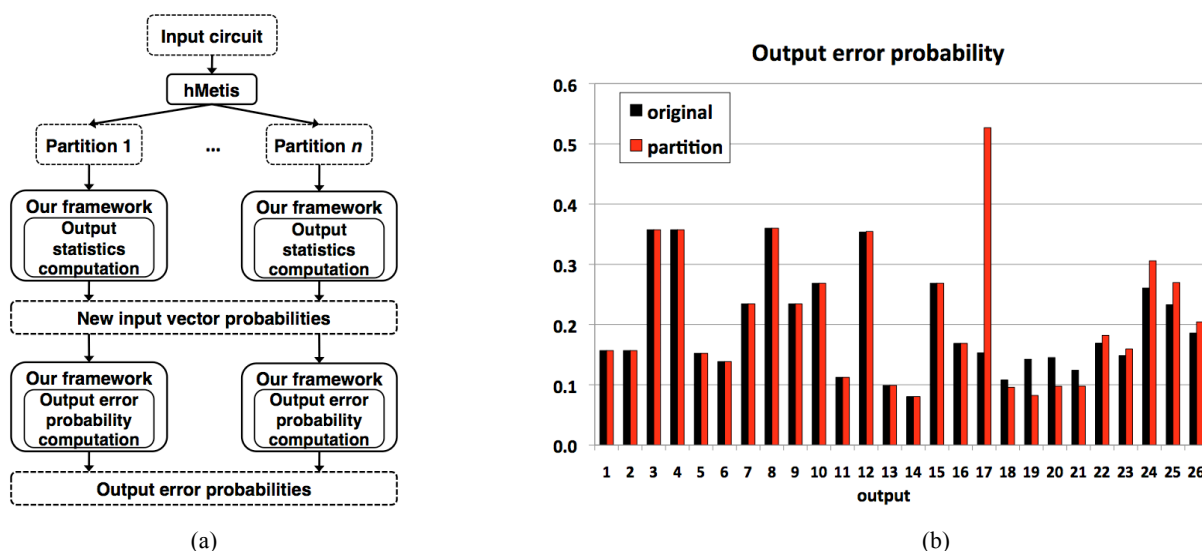
(a)   (b)

Figure 9. (a) The partitioning algorithm flow. (b) Individual output error probabilities for benchmark circuit *C880*, computed using original circuit (**original**), and using partitioning algorithm and the two obtained circuit partitions (**partition**).

[22] I. Polian, S. M. Reddy and B. Becker, "Scalable Calculation of Logical Masking Effects for Selective Hardening Against Soft Errors," in *Proc. of Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 257-262, 2008.

[23] R. Rajaraman, J. S. Kim, N. Vijakrishnan, Y. Xie and M. J. Irwin, "SEAT-LA: A Soft Error Analysis Tool for Combinational Logic," in *Proc. of the International Conference on VLSI Design (VLSID)*, pp. 499-502, January 2006.

[24] 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.

[25] T. Rejimon and S. Bhanja, "Scalable Probabilistic Computing Models using Bayesian Networks", *in Proc. of Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 712-715, 2005.

[26] D. Rossi, M. Omana, F. Toma and C. Metra, "Multiple Transient Faults in Logic: An Issue for Next Generation ICs?," in *Proc. of International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT)*, pp. 352-360, October 2005.

[27] C. Rusu, A. Bougerol, L. Anghel, C. Weulerse, N. Buard, S. Benhammadi, N. Renaud, G. Hubert, F. Wrobel and R. Gaillard, "Multiple Event Transient Induced by Nuclear Reactions in CMOS Logic Cells," in *Proc. of International On-Line Testing Symposium (IOLTS)*, pp. 137-145, July 2007.

[28] G. P. Saggese, N. J. Wang, Z. T. Kalbarczyk, S. J. Patel and R. K. Iyer, "An Experimental Study of Soft Errors in Microprocessors," in *IEEE Micro*, Vol. 25, No. 6, pp. 30-39, November 2005.

[29] N. Seifert, B. Gill, V. Zia, M. Zhang and V. Ambrose, "On the Scalability of Redundancy Based SER Mitigation Schemes," in *Proc. of International Conference on Integrated Circuit Design and Technology (ICICDT)*, pp. 1-9, May 2007.

[30] S. A. Seshia, W. Li and S. Mitra, "Verification-Guided Soft Error Resilience," in *Proc. of the Conference on Design, Automation and Test in Europe (DATE)*, pp. 1442-1447, 2007.

[31] 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.

[32] F. Wang, Y. Xie, R. Rajaraman and B. Vaidyanathan, "Soft Error Rate Analysis for Combinational Logic Using an Accurate Electrical Masking Model," in *Proc. of the International Conference on VLSI Design (VLSID)*, pp. 165-170, January 2007.

[33] B. Zhang, W. Wang and M. Orshansky, "FASER: Fast Analysis of Soft Error Susceptibility for Cell-Based Design," in *Proc. of the International Symposium on Quality Electronic Design (ISQED)*, pp. 755-760, April 2006.

[34] C. Zhao, X. Bai, and S. Dey, "A Scalable Soft Spot Analysis Methodology for Compound Noise Effects in Nano-meter Circuits," in *Proc. of Design Automation Conference (DAC)*, pp. 894-899, June 2004.

[35] *Berkeley Predictive Technology Model (BPTM)*: http://www-device.eecs.berkeley.edu/~ptm.

Natasa Miskov-Zivanov is a Postdoctoral Associate in the Department of Computational and Systems Biology in the School of Medicine at the University of Pittsburgh. She received her Ph.D. and M.S. degrees in Electrical and Computer Engineering at Carnegie Mellon University, in 2008 and 2005, respectively, and a B.S. degree in Electrical Engineering and Computer Science from University of Novi Sad, Serbia, in 2003. Her research interests include computational methods for modeling and analysis of biological systems, reliability and fault-tolerance in nanoscale designs and emerging technologies.

Diana Marculescu is a Professor of Electrical and Computer Engineering at Carnegie Mellon University. She received her Dipl. Ing. degree 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 Best Paper Awards from IEEE Asia South-Pacific Design Automation Conference (ASPDAC 2005), IEEE International Conference on Computer Design (ICCD 2008), and International Symposium on Quality of Electronic Design (ISQED 2009). Diana Marculescu was an IEEE-Circuits and Systems Society Distinguished Lecturer (2004-2005), the Chair of the ACM Special Interest Group on Design Automation (2005-2009) and is a Senior Member of ACM and IEEE. Her research interests include energy- and reliability-aware computing, CAD tools for low power systems and emerging technologies.