

Modeling and Analysis of SER in Combinational Circuits

Natasa Miskov-Zivanov¹ and Diana Marculescu²

¹University of Pittsburgh, ²Carnegie Mellon University

E-mail: nam66@pitt.edu, dianam@cmu.edu

Abstract - Transient faults in logic circuits are an important reliability concern for future technology nodes. In order to guide the design process and the choice of circuit optimization techniques, it is important to accurately and efficiently model transient faults and their propagation through logic circuits, while evaluating the error rates resulting from transient faults. To this end, we give an overview of the existing methods for modeling and reasoning about transient faults. We describe the main aspects of transient fault propagation that each model needs to include and the advantages and drawbacks of different approaches to modeling them.

1. Introduction

Sensitivity of circuits to radiation faults became an important topic of scientific journals for the first time in the seventies [13]. At that time, this topic drew an attention of military and aerospace memory designers and nuclear physicists only. However, in the middle of nineties, many electronics companies became very interested in effects of alpha particles and neutrons on new technologies, increasing the research efforts in this field [29]. Moreover, in recent years, further technology scaling results in devices and systems that are more sensitive to transient faults [17]: not only that the effect of radiation particles has increased, but also sources of transient faults have increased in their number. For example, current systems can contain millions of gates, working at very high frequency and very small supply voltage levels. This trend has already led to increased cross talk or ground bounce, as well as variability in the behavior of transistors.

Faults that are induced by radiation still receive most of the attention among transient faults and 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). If a transient fault is generated in a memory cell, or in a memory element (flip-flop, latch), an error resulting from this fault can immediately occur. Otherwise, the created pulse propagates through the circuit (Figure 1(a)), and causes an error once it is latched by memory cell or memory element. An error caused by a transient fault is called *soft error*, due to the fact that if a failure results as the end effect of this fault, only data are destroyed. In contrast to this, *hard errors* stem from permanent or intermittent faults that result from the damage in the internal structure of semiconductor material. Another often used term for a radiation-induced error is *single-event upset* (SEU). The most often measure for soft errors is the rate at which soft errors occur, called Soft Error Rate (SER), and its unit of measure is Failure-In-Time (FIT). One FIT is equivalent to one failure in 10⁹ device-hours.

As stated in several recent works [25][24], logic soft errors are increasingly important to the global SER. This stems from the fact that the impact of fault masking is decreasing in logic, making it more susceptible to soft errors. Reduction in feature sizes and supply voltages allows lower energy particles to result in SETs. Reduced logic depth and smaller gate delays decrease attenuation when the glitch propagates through the circuit. Finally, increase in clock frequency decreases latching-window masking.

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. One obvious method is to inject the

fault into the node of the circuit and simulate the circuit for different input vectors and for different fault-originating locations (nodes), in order to find whether the fault propagates [29]. However, this approach quickly becomes intractable for larger circuits and larger number of inputs and thus gives way to formal approaches that use analytical and symbolic methods to evaluate circuit susceptibility to transient faults.

The main goal of formal modeling of transient faults in logic is to allow for efficient estimation of the susceptibility of combinational and sequential circuits to transient faults, with a small estimation error. Although so far there have been several design solutions that address 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 [10]. Reliability analysis is proven to be essential in early design stages for improving system lifetime and for allowing exploration of existing tradeoffs [3]. Therefore, fast and accurate estimation of error rates resulting from transient faults in logic circuits is crucial for identifying the features needed for future reliable circuits. In other words, these models can be used to reduce the cost of applying various techniques for error hardening, detection and correction.

2. Elements of fault modeling in logic

We describe in the following the elements of modeling transient faults in logic that are related to individual nodes in the circuit. We also compare how these aspects are handled in different approaches. First, we discuss modeling decisions related to the pulse shape. Next, we describe a very important aspect of transient fault modeling in logic circuits - the set of *masking factors* (logical, electrical and latching-window masking) that can prevent the fault from propagating to the outputs of the circuit. Finally, we discuss modeling of reconvergent glitches.

2.1. Transient fault shape

When evaluating the impact of transient faults on circuit reliability, before developing a model for transient fault propagation, one first needs to make a decision about the pulse shape. In other words, it is necessary to determine which parameters will be used to describe a pulse that represents a transient fault. One approach to modeling pulses is to use very accurate model, which requires detailed information about the pulse. For example, the current pulse that results from a particle strike (Figure 1(b)) is described as a double exponential function:

$$I_{in}(t) = \frac{Q_{coll}}{\tau_{fall} - \tau_{rise}} \cdot \left(e^{-t/\tau_{fall}} - e^{-t/\tau_{rise}} \right) \quad (1)$$

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. If one is to use simulation (e.g., HSPICE) to model pulse propagation through logic, the description in (1) is sufficient for modeling the pulse shape. However, there is a tradeoff between the accuracy of the glitch model and the time required for a method based on such a model to estimate the impact of the glitch on circuit outputs. Another possibility is to model the pulse as piecewise linear and in that case, it is enough to represent the pulse using a few simple parameters

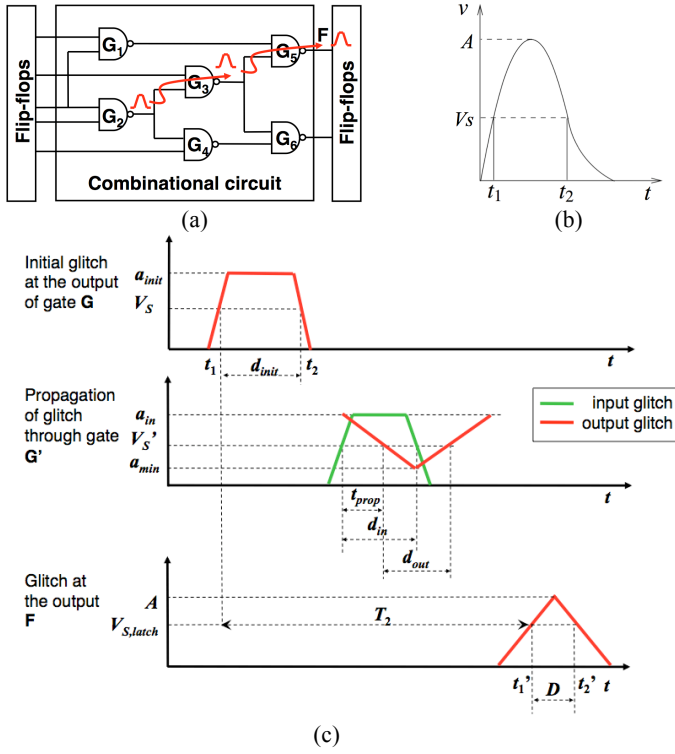


Figure 1. Combinational circuit *C17* with the example glitch propagation path (a), pulse originating from a particle strike (b) and pulse approximation with propagation parameters (c).

(Figure 1(c)). Simpler models, like triangular or trapezoidal [1][6] [20] include information about glitch duration only, amplitude only, duration and amplitude, and some models include slope in addition to duration and amplitude [20].

When describing pulses as piecewise linear, one can use the approach in [7] to determine the interval of pulse sizes of interest in case of particle hits. The authors in [7] present an analytical approach for estimating the width of a pulse described by equation (1). The pulse width is computed for a given gate type, given fanout gates, Q_{coll} , τ_{rise} and τ_{fall} , and the cell library data, drain-source current, I_{DS} , input gate capacitance, and the output node diffusion capacitance.

2.2. Logical masking

When the transient fault arrives at the input of a gate inside the circuit under consideration, if at least one of the other inputs of that gate has a controlling value, the fault is logically masked. In other words, logical masking prevents the fault from propagating through the gate and consequently, prevents the fault from propagating further through the circuit, to the memory element or memory cell at the end of the path. It is important to note here that different gate types have different logical masking “strength.” More precisely, an inverter will always logically propagate a glitch, since there is only one input carrying the glitch, while the probability of propagating the glitch of AND, OR, NAND and NOR gates is the same and depends on the number of inputs. Furthermore, a glitch will always propagate through XOR and XNOR gates due to their logic function. Thus, the logical propagation of the glitch through the circuit will depend on the type of gates used for implementing the circuit, as well as circuit topology and primary input values.

The basic idea behind modeling logical propagation of a glitch through a given gate is finding the probability of propagating a fault from the inputs to the output of a gate. Therefore, it is necessary to measure the sensitivity of the gate output to the value at the fault carrying input. If the output of a given gate is denoted as $f = f(x_1, \dots,$

$x_i, \dots, x_n)$, the sensitivity of a gate output to a given gate input x_i , computed as a Boolean difference, is written as:

$$\frac{\partial f}{\partial x_i} = f_{x_i=1} \oplus f_{x_i=0} \quad (2)$$

When analyzing only a given gate, without taking into account the context of the whole circuit, the sensitivity of a gate to an input fault depends on the gate type and the correct values carried on the other inputs. From a more global (circuit) perspective, values at gate inputs depend on values at primary circuit outputs. Therefore, most modeling approaches keep track of signal values (or probabilities) starting from primary inputs and compute signal values (probabilities) at each gate output. These signal values (probabilities) are then used to compute propagated fault probability in a fanout cone of a gate where the fault originated.

2.3. Electrical masking - pulse attenuation

Due to the relation between electrical properties of the gates and the size of the pulse representing the transient fault, the fault may be attenuated (electrically masked) by the gates through which it propagates. Thus, electrical masking depends on the properties of the gate the fault propagates through, as well as on the properties of the fault (e.g., duration and amplitude) that resulted from the initial characteristics of the fault and the path it propagated through.

The attenuation of the fault may result in its disappearance before it reaches any or some of the outputs of the circuits, or it may decrease the duration and amplitude of the fault such that it is no longer large enough to cause a bit flip in a memory cell or memory element, once it arrives at their inputs. Gates that have larger delays, such as XOR and XNOR gates, will attenuate glitches more, while an inverter usually attenuates glitches much less. If the glitch is “small,” compared to the delays of gates it propagates through, it will always be attenuated. On the other hand, if the glitch is “large,” compared to typical gate delays, it will always propagate to outputs.

There are a number of methods that do not model electrical masking, and instead just focus on logical masking [5][8][9][11] and, in some cases, on latching-window (timing) masking [2][5] [8][9][11]. However, approaches that completely overlook electrical masking and simplify modeling by assuming logical and latching-window masking only, are unable to correctly model fault propagation, and thus provide too pessimistic error rate estimates – in many cases more than 200% larger than actual values. As seen in Figure 2(a), in most cases logical masking has approximately the same impact as the electrical properties in affecting the propagated glitch, thus emphasizing the importance of considering electrical masking impact. Furthermore, as shown in [16][19], the impact of process variations on error rates is increasing. Process variations affect gate delays and thus, significantly impact glitch attenuation and electrical masking.

In those works that tackle the electrical masking effect, there are two main approaches: (i) lookup table ([20][21][22]) and (ii) analytical modeling ([18]). In both cases, there is a tradeoff between accuracy and efficiency, and often, this is the main source of error in any SER modeling approach. In order to find the balance between accuracy and efficiency, it is possible to use the combination of the two, e.g., lookup table for pre-characterizing some library gate parameters and analytical modeling for computing propagated pulse duration and amplitude [20].

Analytical modeling of glitch propagation can be described on the example path $G \rightarrow G' \rightarrow F$ (Figure 1). As described in [18], when the glitch propagates to the input of gate G' , depending on the relation between the duration d_{in} of the glitch and the propagation time of the gate G' , t_{prop} , there are three possible options:

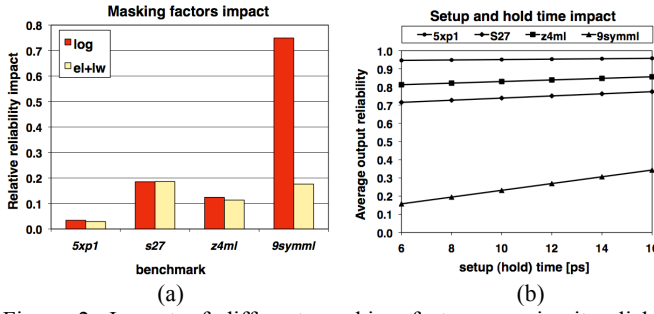


Figure 2. Impact of different masking factors on circuit reliability: logical masking impact vs. electrical and latching-window masking impact, computed as in equation (11) (a), and changes in reliability with the increase in setup and hold time (latching-window) (b).

- If $d_{in} \leq t_{prop}$, then the glitch will not propagate through the gate (it is masked);
- If $t_{prop} < d_{in} \leq 2 \cdot t_{prop}$, then the glitch will propagate, but the amplitude and the duration will be smaller at the output of a gate (it is attenuated);
- If $2 \cdot t_{prop} < d_{in}$, then the glitch will not be attenuated and will be propagated as is.

As it can be seen, the duration of the glitch at the output of the gate through which the glitch propagates depends on the input glitch duration and the propagation delay of the gate. However, if the output glitch amplitude a_{out} is not larger than the switching threshold for the downstream gate, then it can be assumed that the glitch does not propagate at all. One possible approach to approximate the amplitude of the glitch is presented in [18].

2.4. Latching-window (timing) masking

When the transient fault arrives to the input of a memory cell or a memory element, it will be latched only if it arrives on time to satisfy setup and hold time conditions. In other words, if for example, the pulse arriving to the input of a flip-flop represents a 0-1-0 transition, then its rising edge needs to reach the threshold of a flip-flop (or latch), at least a setup time before the clock edge, and its falling edge needs to reach the threshold of a flip-flop at least a hold time after the clock edge. In order to determine the probability of timing masking, one needs to compute the following: (i) the interval where the pulse is *allowed* to occur and (ii) the interval where the pulse *needs* to occur in order to be latched.

The propagation of the glitch and the glitch parameters of interest, when latching-window (timing) masking is considered, are presented in Figure 1(c) [14], on an example path from gate G to output F. More formally, the duration of a glitch at the output of the gate is always measured at switching threshold voltage (V_S) of the downstream gate, and therefore, according to Figure 1(c):

$$d_{init} = t_2 - t_1 \quad (3)$$

At the latched output F, the glitch has amplitude A and duration D . The switching threshold voltage of the latch at which D is measured is $V_{S,latch}$. Since there is a delay from gate G to output F (T_2), the time when the glitch becomes larger than $V_{S,latch}$ is t_1' , and when it becomes lower than $V_{S,latch}$ is t_2' :

$$T_2 = t_1' - t_1 \quad (4)$$

$$D = t_2' - t_1' \quad (5)$$

To satisfy the latching condition, the time at which the glitch reaches $V_{S,latch}$ (t_1') must satisfy:

$$t_1' < T_{clk} - t_{setup} \quad (6)$$

where T_{clk} is the clock period and t_{setup} is the flip-flop setup time. In addition, the time when the glitch becomes less than $V_{S,latch}$ (t_2') must satisfy:

$$t_2' > T_{clk} + t_{hold} \quad (7)$$

where t_{hold} is the flip-flop hold time. The condition that allows for a glitch occurring at gate G to be latched, can be written as [14]:

$$t_1 \in (T_{clk} + t_{hold} - T_2 - D, T_{clk} - t_{setup} - T_2) \quad (8)$$

with duration D of the glitch at output F given by equation (5). Besides the interval in equation (8), representing the interval where t_1 needs to occur in order for glitch to be latched, one also needs to determine the time interval where the glitch is allowed to occur. One possible assumption is that a pulse is equally likely to occur at the output of gate G anytime within a clock cycle period. In other words, t_1 is uniformly distributed within the interval during which the output of gate G is stable. The lower bound of this interval, T_1 , can vary, depending on the primary input vector and on the delay of gates on the path from primary inputs to gate G. The expression for the upper bound of the interval needs to allow for the analysis of the propagation of a glitch with initial duration being equal d_{init} . Thus, the conservative definition of this interval is:

$$t_1 \in (T_1^{\min}, T_1^{\max} + T_{clk} - d_{init}) \quad (9)$$

The two intervals, the interval when t_1 occurs (9) and the interval when it is required to occur (8), can be ordered in time differently, depending on the values at primary inputs. These input values will determine the values of T_1 , T_2 and D . Thus, to be on a safe side, one can find the probability for the worst-case scenario, which occurs when the two intervals completely overlap. In other words, the maximum probability of latching the glitch with a given duration D at primary output is [14]:

$$P(t_1 \in (T_{clk} + t_{hold} - T_2 - D, T_{clk} - t_{setup} - T_2)) = \frac{D - (t_{setup} + t_{hold})}{T_{clk} - d_{init}} \quad (10)$$

The latching of a glitch may also depend on the magnitude of the glitch and the slope of the rising and falling edge of the glitch. The impact of amplitude and slope on latching-window masking is taken into account in [20], by pre-characterizing flip-flops and using a lookup table to determine whether a glitch with given duration, amplitude and slope is latched. If the impact of amplitude and the slope is included in analytical approximation, this decreases the probability of latching, and thus, the probability computed as in (10) is conservative in that sense as well.

As it can be seen from the above discussion, the evaluation of the impact of latching-window masking takes into account both local properties, a setup and hold time of a flip-flop or latch, and global properties, the pulse characteristics (arrival time, duration amplitude and slope) that resulted from its propagation through the circuit. An accurate approach to modeling latching-window masking is important and it needs to include all these parameters. For example, the importance of accurately modeling latching-window masking size can be seen from Figure 2(b) that shows the impact of changes in latching-window on circuit reliability. However, most of the formal approaches that were proposed thus far include only a subset of the above parameters (e.g., setup time, hold time, clock cycle and the time difference between the affected latch and the fault source [12], all parameters except initial glitch size [20], all parameters except glitch slope [14], amplitude and clock cycle only [1], duration and clock cycle only [6]).

2.5. Reconvergent glitches

Once a transient fault occurs at the output of a gate within the circuit, it may propagate through the circuit on more than one path.

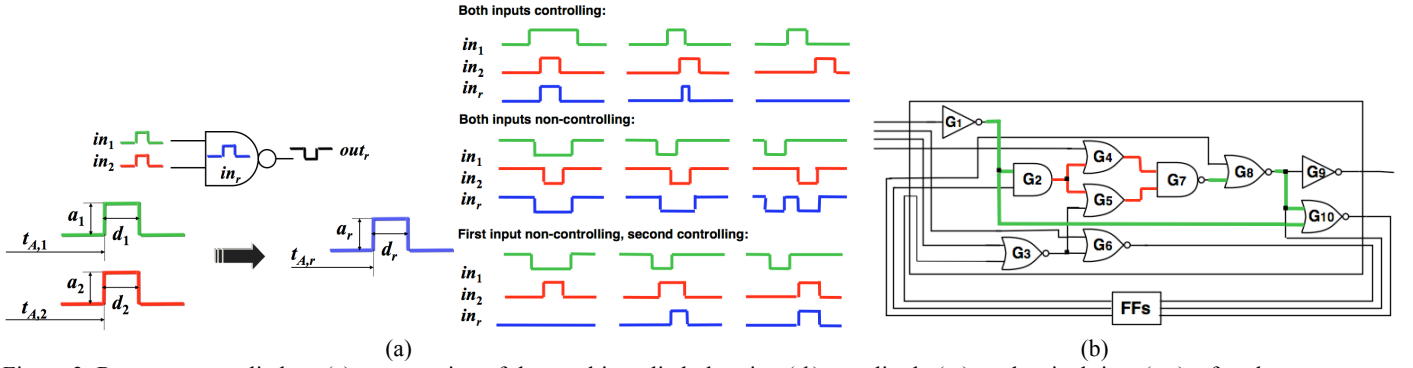


Figure 3. Reconvergent glitches: (a) computation of the resulting glitch duration (d_r), amplitude (a_r), and arrival time ($t_{A,r}$), after the reconvergence of the two input glitches with given durations (d_1, d_2), amplitude (a_1, a_2) and arrival time ($t_{A,1}, t_{A,2}$); (b) reconvergent paths in circuit *S27*.

Besides affecting multiple outputs of the circuit, glitches propagating on different paths can result in *reconvergent glitches* at different inputs of same gate in the fanout cone of the original gate. Figure 3(a) presents the example of reconvergent glitches and parameters that are to be considered when glitches are merged into resulting output glitch. Situations that can occur when two glitches arrive to different inputs of a two input NAND gate are also presented in Figure 3(a).

In Figure 3(b), an example benchmark circuit, *S27*, is shown, with its reconvergent paths highlighted. In circuit *S27*, there are two paths from gate G_2 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. As will be described in Section 3.1, it is important to model reconvergent glitches when modeling glitch propagation. Methods that do not model reconvergence incur a significant error when estimating circuit reliability. However, as will be shown next, only approaches that simultaneously model logical and electrical masking are able to accurately incorporate reconvergent glitch modeling [14][20][26]. Furthermore, only those methods that keep track of the exact signal values at the reconvergence site can model the correlation of gate inputs accurately [14][26], while methods that only propagate signal probabilities can only approximate input correlations [1][5][6].

3. Fault propagation modeling methodology

In the previous section, we gave an overview of important elements that a transient fault modeling approach needs to include. We provide in this section a comparison of different methodologies that were proposed for the evaluation of circuit susceptibility to transient faults.

3.1. Simultaneous treatment of all masking factors

The importance of treating logical, electrical and latching-window masking in a unified manner is emphasized in the following:

- The propagation of a glitch depends on inputs and circuit topology since, for different input vectors, different paths in the circuit are sensitized;
- Glitch attenuation on its way from the originating gate to circuit outputs depends on the gates through which glitch propagates, and thus its impact is affected by logical masking;
- The probability of latching the glitch depends on (i) the glitch size at the output, which in turn is a function of the initial size of the glitch and the attenuation on the sensitized paths; and (ii) the size and relative arrival time of reconvergent glitches, which affects the amplitude and duration of the resulting glitch.

Thus, considering these three factors independently is an incorrect assumption as they all depend on the circuit inputs and sensitized paths from the gate where they occur to outputs. To prove this claim, the two examples are shown in Figure 4 and detailed here. Two *ISCAS'85* benchmark circuits, *C17* and *S27*, are analyzed using two approaches:

1. First, two values are computed:

P_L – the probability of glitch being propagated when only logical masking is taken into account (**LM** column in Figure 4(a));

P_{E+LW} – the probability of glitch being latched when only electrical and latching-window masking are assumed (**ELWM** column in Figure 4(a));

Next, the two probabilities are multiplied to obtain the final error probability (**LM+ELWM** column in Figure 4(a)):

$$P = P_L \cdot P_{E+LW}$$

2. Logical and electrical masking factors are treated in a unified manner, while glitch propagates through the circuit and the probability of the glitch being latched is computed at outputs according to the input vector probability distribution, latching-window size and the glitch arrival time and size at the outputs (**UM** column in Figure 4(a)).

As seen in Figure 4(a), the difference in error probability obtained using a unified approach and any separate approach can be significant. Furthermore, Figure 4(b) represents 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. All results are computed using the framework from [14], which has been validated against HSPICE and found to have 95% accuracy, while being 11000X faster. 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%. For smaller glitch duration (80ps), the average error is not very large, due to the fact that most glitches are masked, and thus, 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.

The reason behind the importance of simultaneous treatment of the three masking factor can be described on the benchmark circuit *S27* (Figure 3(b)) example. The reconvergent paths in circuit *S27* lead to the following scenarios:

1. Only one path exists from a given gate to a given output.
2. More than one path exists from a given gate to a given output.

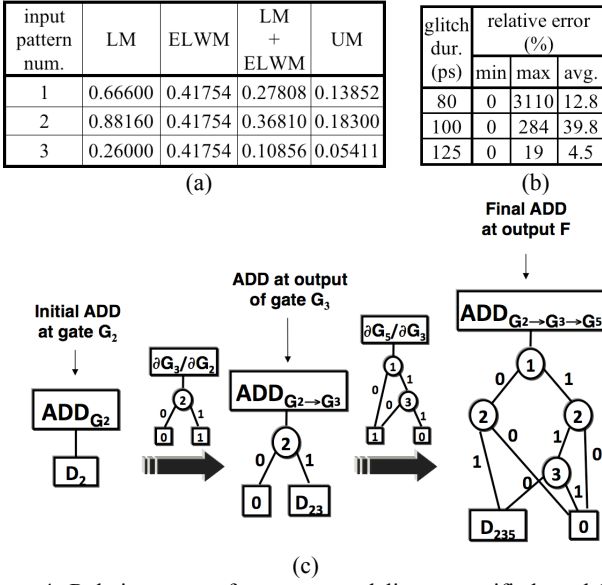


Figure 4. Relative error of separate modeling vs. unified modeling of three masking factors in (a) *C17*, for three different input vector probability distributions, (b) *S27*, for three different initial glitch sizes and (c) symbolic modeling approach with simultaneous modeling of the masking factors.

In case 1, the probability computed using the unified and separate models is the same ($P_L = P_{E+LW}$), since in the unified model the probability of latching the propagated glitch is multiplied by the probability of sensitization on this path.

In case 2, we also need to analyze two possible sub-cases: (a) glitches on some of the paths are attenuated before reaching the reconvergence point and (b) glitches on all paths are propagated to the reconvergence point, where they merge into the resulting glitch(es) with new durations. In these two different sub-cases, 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. Furthermore, when using separate logical, electrical and latching window masking computation, the propagation probability on the sensitized path needs to be multiplied with the latching probability as well.

As seen from the above discussion and the examples in Figure 4(a) and Figure 4(b), a unified treatment of the three masking factors (logical, electrical and latching-window masking) is mandatory for highly accurate estimations. However, most of the previous approaches either treat a subset of masking factors [2][5][9][11][18], or treat and evaluate their impact separately and then merge them into the final reliability measure [6][23][27][28].

One approach that is able to treat the three masking factors in a unified manner is proposed in [14]. The main idea of that approach is that the impact of the three masking factors can be modeled using Binary Decision Diagrams (BDDs) and Algebraic Decision Diagrams (ADDs). This approach is explained in detail in [14]. In Figure 4(c), we give an example of glitch duration ADDs and sensitization BDDs generated for benchmark circuit *C17* (Figure 1(a)), 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 4(c), 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 4(c)) 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 used for modifying the original glitch ADD and a 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_2 to primary output F .

This example shows the propagation of one glitch only. However, the important advantage of the model proposed in [14] is that it *concurrently* computes the propagation and the impact of transient faults originating at any internal gate of the circuit. A similar approach has been proposed in [26], but it uses BDDs only and the algorithm presented is run separately for different polarities at the output of affected gate (fault source location) and separately for each affected gate. The concurrent computation of glitch propagation can account for both single faults and multiple simultaneous faults [15].

3.2. Error rate computation

To find the overall circuit error susceptibility, one can determine the average across all output error probabilities or find the maximum and minimum output error susceptibility. However, *multiple errors* can occur as a result of a single fault being propagated and latched by more than one flip-flop or memory cell. Multiple latched faults are of special concern in sequential circuits where, if latched by state flip-flops, they can continue to propagate through the circuit, causing errors in more than one clock cycle. In addition, averaging across output error probability to determine mean susceptibility of circuit to faults may decrease accuracy, if output error correlations are not accounted for. The reliability of a given circuit when all output correlations are known, can be found using output error probabilities:

$$R = 1 - \left(\sum_j P(F_j) - \sum_{j_1=1}^{n_F-1} \sum_{j_2=j_1+1}^{n_F} P(F_{j_1} \cap F_{j_2}) + \sum_{j_1=1}^{n_F-1} \sum_{j_2=j_1+1}^{n_F} \sum_{j_3=j_2+1}^{n_F} P(F_{j_1} \cap F_{j_2} \cap F_{j_3}) - \dots + (-1)^{n_F-1} \sum_{j_1=1}^{n_F-1} \sum_{j_2=j_1+1}^{n_F} \dots \sum_{j_{n_F}=j_{n_F-1}+1}^{n_F} P\left(\bigcap_{j_i=1}^{n_F} F_{j_i}\right) \right) \quad (11)$$

where n_F is the number of primary outputs, $P(F_j)$ is the probability that outputs $F_{j_1}, \dots, F_{j_i}$ have latched errors in the same cycle, stemming from the same fault source.

However, taking into account all possible output correlations can severely increase complexity. One can instead determine upper and lower bounds for circuit error probability by computing correlations across pairs or triplets of outputs only. The symbolic approach that uses BDDs and ADDs is very convenient for determining these correlations, since ADDs that represent glitches at circuit outputs include the information about all input vectors and finding output error correlations require only multiplication of these ADDs (*i.e.*, AND-ing of their corresponding BDDs).

4. Modeling accuracy and circuit optimization

Once the *gate-output* error probability, that is, the probability that a fault originating at the *gate* results in an error at the *output*, is obtained, it can be further used to obtain more information about the circuit. For each gate, one can find a fanout cone affected by the fault originating at that gate. Next, minimum, maximum, mean and median probability of error at outputs can be determined, given that

the fault occurred at the specified gate. These values describe individual gate error impact and provide guidance when deciding which gates in the circuit need to be hardened. Similarly, for each output, a fanin cone can be found, representing all gates from which faults propagate to the output affecting its correctness. Minimum, maximum, mean and median probability of error at the given output can be computed, in order to better describe output error susceptibility. Input vector probability distribution provides an insight into the way by which input patterns affect gate error impact and output error susceptibility. One can obtain information about circuit's susceptibility to faults by computing the weighted average of error probability across different input probabilities. The weighted average across different initial glitch sizes is of interest as well, and one can assume a distribution of initial glitch duration and amplitude to accurately determine the impact of glitch size on circuit susceptibility.

As shown in Figure 2(b), 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 seen from Figure 2(b), 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 used in experiments. 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 example, in the case of circuit *9symml*, improving electrical masking can lead to significant improvement in error rates.

Thus, in order to guide protection techniques, accurate modeling and evaluation of circuit reliability (error probability) is crucial. With the inclusion of power and performance data, the accurate 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. Underestimation that might occur due to neglecting the impact of variability or, in some cases, inaccurate modeling of reconvergence may result in inadequate protection or hardening choices. On the other hand, overestimation of error rates that can occur due to, for example, ignoring electrical and timing masking, determines the use of too conservative protection and hardening techniques, and consequently, in overdesigning with higher performance penalty, power or area cost.

5. Conclusion

We presented in this paper the aspects of transient fault propagation that need to be accounted for when using formal methods to model and analyze them. In addition, we gave an overview of how these aspects have been tackled by different symbolic and analytical approaches proposed thus far. Finally, we discussed the importance of accurate and efficient modeling for the purpose of guiding the design process.

6. References

[1] H. Asadi and M. B. Tahoori, "Soft Error Derating Computation in Sequential Circuits," in *Proc. of ACM/IEEE International Conference on Computer Aided Design (ICCAD)*, pp. 497-501, November 2006.

[2] H. 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.

[3] 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.

[4] S. Borkar, "Tackling variability and Reliability Challenges," in *IEEE Design and Test of Computers*, Vol. 23, No. 6, pp. 520, June 2006.

[5] 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, pp. 392-405, March 2009.

[6] 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.

[7] R. Garg, C. Nagpal and S. P. Khatri, "A Fast, Analytical Estimator for the SEU-induced Pulse Width in Combinational Designs," in *Proc. of Design Automation Conference (DAC)*, pp. 918-922, June 2008.

[8] C. J. Hescott, D. C. Ness, D. J. Lilja, "Scaling Analytical Models for Soft Error Rate Estimation Under a Multiple-Fault Environment," in *Proc. of Euromicro Conference on Digital System Design Architectures, Methods and Tools*, pp. 641-648, 2007.

[9] D. Holcomb, W. Li and S. A. Sashia, "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.

[10] A. KleinOowski, 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.

[11] S. Krishnaswamy, G. F. Viamonte, 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.

[12] S. Krishnaswamy, I. L. Markov and J. P. Hayes, "On the Role of Timing Masking in Reliable Logic Circuit Design," in *Proc. of Design Automation Conference (DAC)*, pp. 924-929, June 2008.

[13] T. C. May, "Soft Errors in VLSI: Present and Future," in *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, CHMT-2, No. 4, pp. 377-387, 1979.

[14] 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)*, Vol. 25, No. 12, pp. 2638-2639, December 2006.

[15] 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.

[16] N. Miskov-Zivanov, K.-C. Wu and D. Marculescu, "Process Variability-Aware Transient Fault Modeling and Analysis," in *Proc. of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, to appear, November 2008.

[17] S. Mitra, M. Zhang, T. Mak, N. Seifert, V. Zia and K. S. Kim, "Logic soft errors: a major barrier to robust platform design," in *Proc. of International Test Conference (ITS)*, November 2005.

[18] M. Omana, G. Papasso, D. Rossi, and C. Metra, "A Model for Transient Fault Propagation in Combinatorial Logic," in *Proc. of IEEE International On-Line Testing Symposium (IOLT'S)*, pp. 11-115, July 2003.

[19] H.-K. Peng, C. H.-P. Wen and J. Bhadra, "On Soft Error Rate Analysis of Scaled CMOS Designs - A Statistical Perspective," in *Proc. of International Conference on Computer Aided Design (ICCAD)*, pp. 157-163, March 2009.

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

[21] K. Ramakrishnan, R. Rajaraman, N. Vijaykrishnan, Y. Xie, M. J. Irwin and K. Unlu, "Hierarchical Soft Error Estimation Tool (HSEET)," in *Proc. of International Symposium on Quality Electronic Design (ISQED)*, pp. 680-683, March 2008.

[22] 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.

[23] 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.

[24] N. Seifert, P. Slankard, M. Kirsch, B. Narasimham, V. Zia, C. Brookreson, A. Vo, S. Mitra, B. Gill and J. Maiz, "Radiation-Induced Soft Error Rates of Advanced CMOS Bulk Devices," in *Proc. of the IEEE International Reliability Physics Symposium*, pp. 217-225, March 2006.

[25] 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.

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

[27] 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.

[28] 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.

[29] J. F. Ziegler et al., "IBM experiments in Soft Fails in Computer Electronics (1978-1994)," in *IBM Journal of Research and Development*, Vol. 40, No. 1, pp. 3-18, 1996.