Circuit Optimization Techniques to Mitigate the Effects of Soft Errors in Combinational Logic

RAJEEV R. RAO
Magma Design Automation
VIVEK JOSHI, DAVID BLAAUW, and DENNIS SYLVESTER
University of Michigan

Soft errors in combinational logic circuits are emerging as a significant reliability problem for VLSI designs. Technology scaling trends indicate that the soft error rates (SER) of logic circuits will be dominant factor for future technology generations. SER mitigation in logic can be accomplished by optimizing either the gates inside a logic block or the flipflops present on the block boundaries. We present novel circuit optimization techniques that target these elements separately as well as in unison to reduce the SER of combinational logic circuits.

First, we describe the construction of a new class of flip-flop variants that leverage the effect of temporal masking by selectively increasing the length of the latching window thereby preventing faulty transients from being registered. In contrast to previous flip-flop designs that rely on logic duplication and complicated circuit design styles, the new variants are redesigned from the library flip-flop using efficient transistor sizing. We then propose a flip-flop selection method that uses slack information at each primary output node to determine the flip-flop configuration that produces maximum SER savings. Next, we propose a gate sizing algorithm that trades off SER reduction and area overhead. This approach first computes bounds on the maximum achievable SER reduction by resizing a gate. This bound is then used to prune the circuit graph, arriving at a smaller set of candidate gates on which we perform incremental sensitivity computations to determine the gates that are the largest contributors to circuit SER. Third, we propose a unified, co-optimization approach combining flip-flop selection with the gate sizing algorithm. The joint optimization algorithm produces larger SER reductions while incurring smaller circuit overhead than either technique taken in isolation. Experimental results on a variety of benchmarks show average SER reductions of 10.7X with gate sizing, 5.7X with flip-flop assignment, and 30.1X for the combined optimization approach, with no delay penalties and area overheads within 5-6%. The runtimes for the optimization algorithms are on the order of 1-3 minutes.

Categories and Subject Descriptors: B.7.3 [**Integrated Circuits**]: Reliability and Testing

General Terms: Design, Reliability

This work was supported in part by the NSF, SRC, and GSRC/DARPA.

Authors' addresses: V. Joshi, D. Blauw, D. Sylvester, Department of EECS, University of Michigan, Ann Arbor, MI; email: rajeev80@gmail.com.

DOI 10.1145/1640457.1640462 http://doi.acm.org/10.1145/1640457.1640462

Additional Key Words and Phrases: Soft errors, circuit optimization, sequential circuits, combinational logic

ACM Reference Format:

Rao, R. R., Joshi, V., Blauw, D., and Sylvester, D. 2009. Circuit optimization techniques to mitigate the effects of soft errors in combinational logic. ACM Trans. Des. Autom. Electron. Syst. 15, 1, Article 5 (December 2009), 27 pages.
DOI = 10.1145/1640457.1640462 http://doi.acm.org/10.1145/1640457.1640462

1. INTRODUCTION

Energetic cosmic particles interact with the silicon substrate in integrated circuits to produce transient noise events. A radiation particle strike on an SRAM cell or a memory register that can cause a bit flip is called a *single event upset* (SEU). Similarly, a particle strike on a logic gate in a combinational circuit can produce a voltage glitch referred to as a *single event transient* (SET). An SET can potentially propagate to an output node and cause an erroneous signal to be latched into a flip-flop. These types of radiation induced faults are called soft errors and their frequency is referred to as the *soft error rate* (SER). The quantitative metric used to measure SER is *failures-in-time* (FIT), corresponding to the number of errors in one billion device hours.

Continued technology scaling has resulted in the emergence of soft errors as one of the major reliability challenges for current and future digital VLSI designs. The failure rate due to soft errors is expected to exceed the failure rate due to all other reliability mechanisms (such as gate oxide breakdown, electromigration) combined (Baumann [2005]). Several works [Shivakumar et al. 2002; Karnik et al. 2001; Maestro et al. 2008] have surveyed the impact of soft errors on the various components of a typical integrated circuit. A simultaneous reduction in both the critical charge and collection efficiency has resulted in relatively constant SRAM SER over several technology generations. In addition, error correction codes (ECC) enable a high level of soft error protection for memories. Similarly, Mitra et al. [2005] project industrial estimates that show that the nominal SER of latches is nearly constant from 130nm to 65nm technologies and beyond. The use of radiation hardened latches [Karnik et al. 2002; Faccio et al. 1999; Monnier et al. 1998; Omana et al. 2007] further immunizes latches from particle strikes [Ramanarayanan et al. 2003]. In contrast, SER due to particle hits on combinational logic is predicted to increase rapidly and a recent estimate, Shivakumar et al. [2002] show that SETs in logic will significantly influence chip SER at the 45nm node. In large-scale applications such as server farms and communications systems, logic soft errors are predicted to be significant contributors to system-level silent data corruption events [Mitra et al. 2005; Shazli et al. 2008]. It is, therefore, critical to develop analysis and mitigation techniques to combat the effects of soft errors on logic.

Combinational logic circuits can be immunized against the effects of soft errors using two methods. First, the probability of a transient glitch occurring at any sensitive node in the circuit can be minimized. This approach targets the soft error problem at the source by lowering the probability of an erroneous SET pulse from being generated. Selectively hardening the set of susceptible gates can result in the absence of most faulty pulses in the circuit. Second, the probability of an SET being latched into the flip-flop can be minimized. This approach targets the soft error problem at the sink because, although it permits SETs to originate at any node inside the logic, it disallows such erroneous glitches from being registered by the sequential element. By carefully designing a flip-flop to filter a large fraction of the SETs incident at its data port, it is possible to completely suppress a soft error occurring in logic to permeate to the architectural or the system level. Naturally, the selection of one approach over the other is dictated by the amount of overheads that they introduce. Directly modifying the gates inside a circuit incurs, in general, large overheads in power, delay and area that can prohibit design convergence. Conversely, modifying only the flip-flop elements present on the boundary of a logic circuit incurs small cost in terms of power and area but can vastly influence the timing characteristics of the overall design and also place additional constraints on the clock tree network. Hence, it is necessary to consider these gate-based and flip-flop-based SER mitigation approaches separately as well as in unison, along with their associated overheads while optimizing logic circuits for better SER immunity.

This article proposes novel circuit level optimization techniques to minimize SER of combinational logic circuits. First, we illustrate the method for SER reduction using the modification of the latching window associated with the flip-flop. We then present a novel sizing scheme for flip-flops that modulates the sizes of a select few transistors and enables the construction of a variety of SER tolerant flip-flops. This new flip-flop library trades off increased amounts of pulse filtering (and, hence, reduced SET latching susceptibility) with larger amounts of delay overhead. We present a slack-based optimization method where the output flip-flops are selected from the variant library based on the slack available at the node. Next, we present a new gate resizing algorithm that uses accurate sensitivity measurements to guide the optimizer. This approach first prunes the entire circuit to a smaller subset of gates by efficiently computing bounds on the SER reduction achievable by modifying a gate. We then use this subset of gates as possible candidates for resizing and identify gates that provide the maximum SER improvement while incurring the least amount of area overhead. Third, we present a joint optimization algorithm that performs simultaneous gate resizing and slack-based flip-flop assignment. This combined approach produces a near ideal design point by providing significant SER reduction while modifying the original circuit in a minimal fashion. The three techniques incur *zero delay overhead* and instead trade off small amounts of increase in circuit area for SER reduction.

Each proposed optimization technique is exercised on a wide variety of benchmark circuits. Results show that for circuits synthesized with tight delay constraints, we achieve SER reductions of 19.3X while increasing area by 0.6% on average. For circuits synthesized with loose delay constraints, we achieve larger SER reductions of 30.1X while incurring area overhead of up to 4.0% on average.

The article is organized as follows. Section 2 reviews previous work targeted towards logic SER reduction. Section 3 describes the methods by which gate sizing and temporal masking in flip-flops can reduce SER. Section 3 describes the construction of the new flip-flop variants. Section 5 provides a detailed description of the proposed sensitivity-based algorithms. In Section 6, we presents the results, and we conclude in Section 7.

2. PRIOR WORK

Soft error analysis and mitigation is a fairly well-studied topic and a number of methods have been proposed through the years to address this issue. Initial techniques proposed for circuit level radiation hardening are based on classical fault tolerance techniques such as triple modular redundancy. Moharam et al. [2003] propose a more cost-effective approach that duplicates only a portion of the circuit to achieve the target fault coverage. Wu et al. [2008] present an approach based on iterative addition and removal of redundant wires. Garg et al. [2006] propose a method based on duplication using shadow gates. Gate-based SER mitigation methods (as described in the previous section) include techniques that alter some aspects of the circuit structure to selectively harden a small fraction of the susceptible nodes in the circuit. Some examples of these methods use techniques such as transistor sizing [Dhillon et al. 2005; Zhou et al. 2004; Abrishami et al. 2008], ATPG-based rewiring [Almukhaism et al. 2006], dual-Vdd structures [Choudhury et al. 2007] and output remapping (Qian et al. [2008]). On the other hand, flip-flop directed optimization approaches include the dual-sampling latch from Zhang et al. [2005a], flip-flops with delayed data/clock signal sampling from Mavis et al. [2002], dual-ported latches from Zhang et al. [2005b], latches with additional keepers from Karnik et al. [2002], latches enhanced with Schmidt triggers from Sasaki et al. [2008], tunable transient filters from Zhou et al. [2008], and scan flip-flop-based designs from Elakkumanan et al. [2006] and Mitra et al. [2005]. Naturally, the effectiveness of SER protection schemes must be evaluated by the amount of overhead they introduce to the delay, area, and power of the circuit. Standard techniques based on the replicate and recompute design methodology rely on time/space redundancy due to the usage of checkers and logic duplication. Furthermore, the addition of extra gates to a circuit can result in an expansion in the number of vulnerable nodes susceptible to particle strikes, thereby worsening the overall circuit reliability. In contrast, flip-flop-based optimization approaches significantly influence the circuit's delay characteristics.

In this article, we first propose two SER reduction techniques: a node-specific approach using gate resizing and a flip-flop-based approach using a tunable flip-flop latching window. We discuss the utility of each of these methods in the context of achievable SER reduction and the amount of overheads they introduce. We then present a hybrid methodology optimizes the gates and flip-flops simultaneously. The key contribution of our work is this ability to conjoin gate modification with appropriate flip-flop selection to achieve maximum SER reduction while accruing small increases in area and power and zero delay overhead.

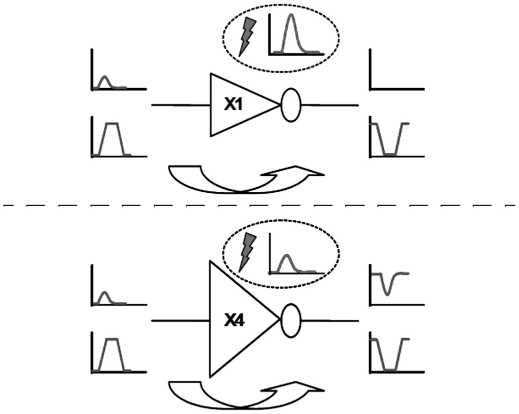


Fig. 1. Qualitative comparison (in terms of electrical properties) between INVX1 and INVX4. Top (Circled) = Injected waveforms and Sides = Propagated waveforms.

3. SER ANALYSIS PRELIMINARIES

This section discusses the mechanisms by which gate sizes and temporal masking impact circuit SER. We describe the efficacy of employing gate resizing and specially designed flip-flops to minimize the circuit SER value. We then provide an overview of the underlying SER computation algorithm.

3.1 Impact of Gate Sizing on SER

The amount of charge generated at a susceptible node in any gate due to a neutron strike is a strong function of its drain area. By sizing up a gate, the effective capacitance of the device is increased thereby making it less likely that the injected transient current will cause a voltage glitch of sufficient magnitude. For instance, consider a single inverter with a fixed output load as shown in Figure 1. Replacing an INVX1 with another inverter INVX4 (with 4X more drive strength) decreases glitch amplitude significantly (see circled waveforms in Figure 1). As a result, upsizing a gate always decreases the probability of transient generation due to direct particle hits. On the other hand, an upsized device has significantly higher drive strength, which allows for better propagation of the input transients at a gate. This is particularly true in cases where the output load of the cell is large. Figure 1 qualitatively shows the two types of input transients at a gate: 1) Nonlinear waveform shapes that can possibly occur due to a strike on the immediately preceding gate and 2) standard trapezoidal shapes that occur when an injected transient has propagated through a few logic stages. In this plot, the INVX1 completely filters the short, nonlinear waveform while allowing the trapezoidal shape to propagate with little or no attenuation. On the other hand, the INVX4 allows the propagation of both types of transients and in fact, produces a slight boost in the signal strength of the nonlinear transient. Transient waveforms with small pulse widths typically correspond to particle hits that inject a small amount of charge but have a larger error rate probability associated with them. Since upsized gates have a higher propensity to propagate these short transients, it is possible that increasing gate sizes unilaterally can actually worsen circuit SER.

In a sensitivity-based timing optimization algorithm (such as TILOS from Fishburn et al. [1985] and its variants), gate sizes are incrementally increased in small steps to determine the size that provides the best delay value. From the previous discussion, we make the key observation in our work that gate sizes can be either increased or decreased to achieve SER reductions. For each gate, it is important to consider the relative significance of the injected and propagated waveforms to the total SER value. This approach is in contrast to Zhou et al. [2004], which considers only the impact of first strike waveforms and Zhau et al. [2004] which only targets the waveforms that are propagated. Further, Zhou et al. [2004] consider only the worst-case injection charge value of 150fC in the analysis, thereby disregarding the vast majority of strikes that inject charge lower than 150fC but contribute a much greater fraction to the total SER value. Considering both injected and propagated waveforms at a gate, across the entire spectrum of neutron strikes, provides a more accurate and realistic assessment of the impact of an individual gate on the total circuit SER. Hence, the proposed algorithms in our work consider gate resizing (upsizing and downsizing) to achieve SER improvement.

In our analysis, we assume that the baseline (unoptimized) circuits are synthesized based on a prescribed set of delay and power constraints. Thus, depending on the available resources, each gate is chosen from a set of sizes so that both upsizing and downsizing can be performed on them.

3.2 Impact of Temporal Masking on SER

Temporal masking (also called timing masking) is the mechanism that determines whether a transient arriving at a sequential element input is latched as an erroneous value [Krishnaswamy et al. 2008a]. Both latch-based and edge-triggered systems are susceptible to registering a spurious voltage pulse in a small region close to the clock edge. This region is referred to as the window of vulnerability in Seifert and Tam [2004] (or the aperture window in Weste and Harris [2005]) is equal to the sum of the setup (T_{setup}) and hold (T_{hold}) times. (We define T_{setup} as the data-to-clock offset T_{DC} that corresponds to a 10% increase in the clock-to-Q delay T_{CQ} from its nominal value as described in Weste and Harris [2005]. T_{hold} is defined in similar fashion). The aperture window is the width of the window around the clock edge during which the data must not transition if the memory element is to produce the correct output. Note that in this definition of an aperture window, we have neglected the effects of clock uncertainty (such as skew and jitter).

For a given transient waveform k with pulse width T_{pw}, let $z(k)$ be the temporal probability that k causes a faulty logical bit to be registered. A flip-flop is susceptible to capturing a spurious bit if the transient pulse completely overlaps the latching ($T_{setup} + T_{hold}$ time) window thereby producing a faulty transition. This effect is shown in Figure 2 using a simple PWL for the waveform incident at the data pin. (Although we use PWLs to illustrate this effect, waveforms generated due to particle strikes have more complex shapes and their effects

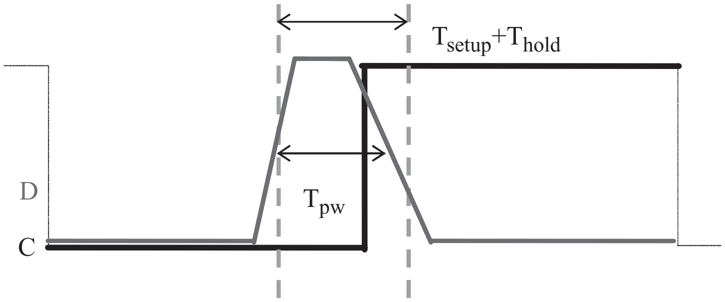


Fig. 2. Timing diagram showing data (D) and clock (C) signals for a flipflop. A soft error event occurs when the transient pulse completely overlaps the latching window.

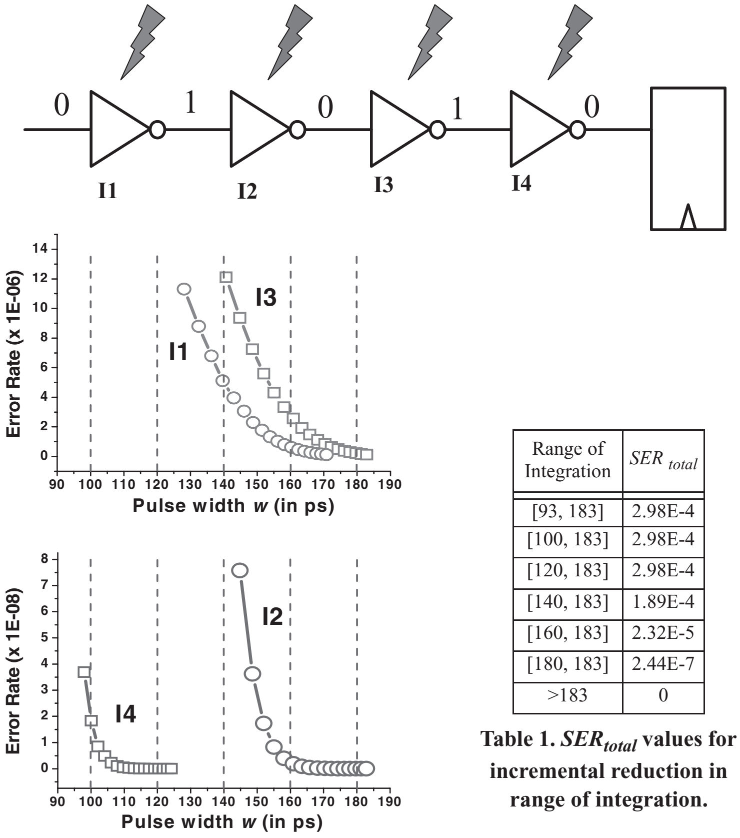
are modeled using the empirical Weibull PDF based methodology we proposed in our SER analysis work [Rao et al. 2006]).

On the other hand, if $T_{pw} < (T_{setup} + T_{hold})$ no logical error is produced. Thus, a very good approximation for $z(k)$ is given by the following expression:

$$z(k) \approx \begin{cases} 0 & T_{pw} < (T_{setup}) + (T_{hold}) \\ \frac{T_{pw}-(T_{setup}+T_{hold})}{T_{Clk}} & T_{pw} \geq (T_{setup} + T_{hold}). \end{cases} \tag{1}$$

Here T_{Clk} is the clock period of the circuit. In this equation, we observe that the probability of a soft error occurring at an output node is inversely proportional to the length of the aperture window. As the value of $(T_{setup} + T_{hold})$ is increased, a larger fraction of the transient pulses will have pulse widths $T_{pw} < (T_{setup} + T_{hold})$, such that the temporal probability $z(k)$ associated with those pulses will become zero.

To illustrate the gate-level effect of temporal masking on radiation-induced waveforms, we constructed a single-input/single-output 4-stage inverter chain connected to a standard D-Flipflop in an industrial 0.13μm technology (Figure 3). We set the clock period for the flip-flop T_{Clk} to be 1ns. By construction, no logical or electrical masking is possible in this circuit. We set the input to this circuit to 0 and determine the logical values of the other nodes in the inverter chain. First, we observe that the susceptible node in each inverter is dependent on the input state: an inverter with input = 1 (0) defines the PMOS (NMOS) drain as the vulnerable region in the device. A large difference (about two orders of magnitude) exists between the strike probabilities associated with NMOS compared to those of PMOS devices (Hazucha and Svensson [2000]). We derive four SET waveforms at the output node: one pair corresponding to the strikes at I1/I3 and one pair corresponding to the strikes at I2/I4. The rate distribution plots and pulse widths corresponding to these waveforms is also shown in Figure 3. Note that the error rate values for I2/I4 are significantly smaller (by about 100X) than the error rate values for I1/I3. For this set of four descriptors, we observe that the pulse widths are in the range [97ps, 183ps].



Range of Integration	SER_{total}
[93, 183]	2.98E-4
[100, 183]	2.98E-4
[120, 183]	2.98E-4
[140, 183]	1.89E-4
[160, 183]	2.32E-5
[180, 183]	2.44E-7
>183	0

Table 1. SER_{total} values for incremental reduction in range of integration.

Fig. 3. Schematic of four inverter chain and the corresponding rate distribution plots. Note that (11, 13) are 100X larger than (12, 14).

In the baseline (unoptimized) case, all SET waveforms incident on the flip-flop can potentially cause a logical error in the register. The total SER value will, therefore, be the sum of the integrals of the four error rate curves across the entire range of widths. However, by widening the aperture window, we effectively reduce the total range of pulse widths that can pass through the flip-flops at the outputs. In Figure 3 we draw dashed vertical lines along the x-axis to indicate the amounts to which the aperture window can be potentially increased to block a portion of the SET pulses. For instance, for the case when the dashed line is at 140ps, the value of $(T_{setup} + T_{hold})$ is set to be 140ps so that all pulses of width $T_{pw} \leq$ 140ps are guaranteed to be temporally masked with the flip-flop performing a low-pass filtering operation. In this case, the numerical integration for SER calculation is performed by setting the range to be [140ps, 183ps] since the temporal probabilities associated with all pulses lower than value will, by definition, be zero. Hence, large SER reductions can be achieved by gradually shifting the minimum width value along the x-axis to decrease the total range of widths over which we are required to integrate.

This key observation allows us to perform incremental measurements to determine the potential reductions in SER values while increasing the length of the aperture window. In the table adjoining the plots in Figure 3, we present the value of SER_{total} for the given inverter chain circuit while considering various filter points on the pulse width axis. From this table, we observe that since I1/I3 are the dominant contributors to the value of SER_{total}, the reductions achieved by increasing the aperture window to 120ps is negligible. However, when $(T_{setup}+T_{hold})$ is increased to 140ps and greater, we observe an exponential decay in SER_{total}. When $(T_{setup} + T_{hold}) \geq$ 184ps, the value of SER_{total} reduces to a negligible amount.

Cosmic particles (particularly neutrons) that strike combinational logic contain a finite amount of energy. For the 0.13μm technology, Zhou and Moharam [2004] note that the energy levels of neutron strikes can be mapped to deposited charge values in the range [10fC, 150fC]. As a result, the pulse widths of transient glitches also occur for a finite duration in a characterizable range. We report a range of [78ps, 206ps] for a 0.13μm cell library (Rao et al. [2006]). This observation of a finite duration for the pulses leads to the possibility of designing flip-flops that filter transients based on the pulse widths. A library flip-flop typically has $(T_{setup} + T_{hold})$ significantly smaller than the pulse widths in this range so that no filtering is possible. However, by sufficiently increasing the setup/hold times associated with the flip-flop, the filtering window is widened so that a subset of the transients are disallowed from being registered by the flip-flop. This effect is specifically targeted towards the fast (short pulse width) transient waveforms. Since fast transients typically correspond to soft errors with high strike rate probabilities, preventing these SETs from latching enables a significant reduction in the circuit SER.

3.3 SER Analysis Engine

Before we describe the SER optimization techniques, we briefly discuss the underlying SER estimation methodology used in our analysis. Recently, a number of logic soft error analysis algorithms have been presented; these include SERA from Zhang and Shanbhag [2004], ASERTA from Dhillon et al. [2005], SEAT-LA from Rajaraman et al. [2006], HSEET from Ramakrishnan et al. [2008], our descriptor approach from Rao et al. [2006], MARS from Zivanov and Marculescu [2006, 2007] and FASER from Zhang and Orshansky [2006]. These tools employ a variety of techniques such as circuit simulation, probability theory and binary decision diagrams to compute the logic SER.

For the analysis presented in this article we chose to use our tool presented in Rao et al. [2006] for the following reasons: (1) It provides a quick and efficient method for SER computation. As we observe in Section 5.1.2, short runtime for the estimation engine is vital to perform fast incremental SER calculations. (2) The Weibull-based waveform descriptor formulation inherently considers the effects of electrical masking on SER. (3) Unlike the other tools, it considers the entire spectrum of neutron strikes (all charge values in the [10fC, 150fC] range) during SER computation. The strike probabilities associated with the individual charge values varies greatly (by about four orders of magnitude).

We therefore believe that, from an optimization perspective, it is important to consider the full range of charge values, instead of just 4-5 discrete values.

We model the transient glitch due to a neutron strike is modeled using the current pulse model presented in Freeman [1996].

$$I(t) = \frac{2Q_0}{\tau\sqrt{\pi}}\sqrt{\frac{t}{\tau}}\exp\left(\frac{-t}{\tau}\right) \tag{2}$$

Here Q_0 is the amount of injected charge, τ is time-dependent pulse shaping parameter and $I(t)$ is the current. Empirical models from Hazucha and Svensson [2000] are then used to map the deposited charge with a strike rate value.

$$R = F \times K \times A \times \left(\frac{1}{Q_s}\right) \times \exp\left(\frac{-Q_0}{Q_s}\right) \tag{3}$$

Here R = rate of SET strikes, F = neutron flux with energy>10MeV, A = area of the circuit susceptible to neutron strikes, K = a technology independent fitting parameter, Q_0 = charge generated by the particle strike and Q_s = charge collection slope. A parametric descriptor object correlates these strike rate values with a corresponding transient waveform. The logic level SER analysis model consists of the injection and propagation of these descriptors through a circuit. The tool accounts for all the three types of masking mechanisms—logical, electrical and temporal—during the estimation flow. We refer the reader to our earlier paper [Rao et al. 2006] for further details about this tool.

4. SLACK-BASED OPTIMIZATION

This section first describes the method by which a library flip-flop (FF) is redesigned to develop the FF variant library. In contrast to Shivakumar and Keckler [2006] that uses architectural level slack in the form of extra cycles available per instruction, we propose to instead leverage block-level timing slack at the register boundaries for SER mitigation. We analyze the electrical characteristics of the different flip-flops to quantitatively gauge the amount of SER reductions that is possible. We then present a slack-based FF assignment method that uses the FF variant library to selectively replace output flip-flops based on the available slack.

4.1 Flip-flop Variant Construction

A standard D-Flip-flop constructed using back-to-back transparent master/slave latches is shown in Figure 4. Each latch consists of a tristate inverter and a cross-coupled inverter pair. The output nodes (with both true and complemented polarities) Q, QB are buffered to isolate the storage nodes from noise on the output. As described previously in Section 3.2, we seek to redesign this library flip-flop by widening the aperture window so that it is more resilient towards the incident transient waveforms.

The most direct method of altering the setup/hold time characteristics of a flip-flop is by the addition of extra transistors inside the memory element such that the input stage is sufficiently slowed down. However, such a scheme

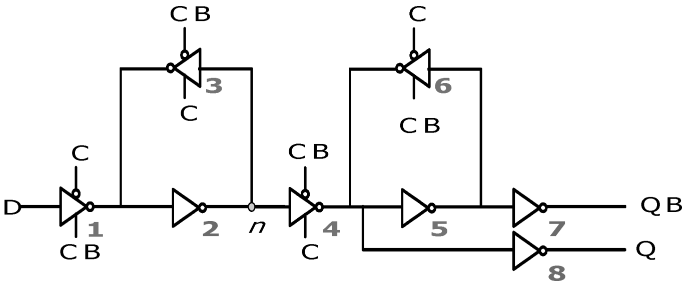


Fig. 4. Circuit schematic of a standard D Flipflop. D = Data, C = Clock, CB = Clock-bar, Q = output QB = Q-bar. The sizes of devices 2, 7, 8 are altered to construct the flipflop variants.

is infeasible since, in addition to the large overheads incurred in power and delay, it increases the effective device area susceptible to direct particle strikes, thereby making the flip-flops more vulnerable to SETs. Cha and Patel [1994] proposed the addition of extra resistors across different pairs of nodes in the flip-flop in order to modify the width of the latching window. This technique is inapplicable for current digital designs due to the high delay and power penalty associated with them (the method in Cha and Patel [1994] for instance incurs a delay penalty of about 300%) as well as the difficulty in including passive elements (such as resistors and capacitors) on the integrated circuit.

Another method to alter the width of the latching window is to use transistor sizing as a design variable. Transistor sizing can be used in two different ways to increase the aperture window of a flip-flop. Reducing the widths of the devices in the master will slow the data signal from reaching the storage node in the latch. Although downsizing transistors is advantageous from a low power perspective, it is not a viable option since it significantly increases the susceptibility of the memory element to direct particle strikes. On the other hand, upsizing has the dual benefit of decreased vulnerability to direct strikes and the ability to mask out temporal glitches due to the transient waveforms. For the analysis presented in this paper we set the flip-flop performance metric as the minimum D-to-Q delay, defined by the sum of setup time T_{setup} and the clock-to-delay T_{CQ}. Since we increase T_{setup} for soft error reduction, we aim to mitigate the performance penalty by decreasing T_{CQ} by a commensurate amount. The reduction in T_{CQ} is also achieved using sizing methods.

We first treat the size of the data input buffer (device 1) as fixed. We avoid resizing this device so that different versions of the flip-flop present the same output load to the combinational circuit. Among Devices 2 and 3, we observe that the forward inverter Device 2 is more suitable from an SER immunity perspective for three reasons: (1) Device 2 presents a larger output load to Device 1,thereby increasing the setup time of the flip-flop. (2) Due to the higher capacitance of Device 2, a larger number of glitches, that can potentially occur at node n, are filtered. Note that before the rising edge of the clock, the master latch is transparent so that a partially overlapping transient pulse can potentially corrupt state node n. However, unlike the case where Device 3 is sized up, increasing the width of Device 2 will help eliminate the possibility of these

Table I. Delay/Area Overheads for the Flip-Flop Variants. A Single FO4 Delay = 40.1ps

Flipflop Variant	T_{pw} Filtering Threshold (in ps)	Overhead Delay (ps)	Overhead Delay (xFO4)	Overhead Area (%)
Lib	27	0	0	0
F100	100	62.4	1.6	< 0.1
F130	130	92.4	2.3	0.1
F160	160	122.5	3.1	0.1
F210	210	153.4	3.8	0.2

glitches. (3) Since Device 2 is not a clocked buffer element, the power overhead during the period when the clock signal is switching is lessened. Concurrently, the most efficient method to decrease the clock-to-Q delay is by sizing down the output drivers 7 and 8. The sizing operation is tuned such that the flip-flop exhibits nearly identical behavior to both rising and falling transitions in terms of the filtering response.

The aforementioned filtering operation is valid only when the flip-flop data bit is not switching from the previous cycle. For the case of switching input data, we observed in Joshi et al. [2006] that the temporal probability is independent of T_{pw} and only depends on the location of the pulse in the overall time interval. The flip-flop variants are inadequate in filtering these types of error events; instead, their circuit response in such cases is identical to that of the standard library flip-flop. However, since the switching probability associated with output nodes is typically a small number (≈0.10-0.20 as stated in Magen et al. [2004]), the contribution of such error events to the total SER value is quite small.

As noted previously, for the industrial 0.13μm cell library that we used for logic SER analysis, all transient pulses have widths in the range [78ps, 206ps]. Since we need to eliminate transient pulses with widths in this range we construct four different variants ($F100$, $F130$, $F160$, $F210$) of flip-flops with the values of the aperture window in this range as shown in Table I. Beginning with a library flip-flop (denoted here as *Lib*), the devices are progressively sized up to obtain four different filtering thresholds. The *Lib* flip-flop does not filter any transient pulses since its filtering threshold of 27ps is much lower than the minimum SET pulse width of 78ps. The $F130$ flip-flop filters all transient pulses with width $T_{pw} \leq 130$ps. The $F210$ flip-flop can potentially eliminate all possible transient pulses from latching into the flip-flop since the maximum transient pulse width is given as 206ps. We observed that the maximum improvement in T_{CQ} that can be achieved by sizing up drivers 7 and 8 was fixed (≈50ps). The delay overhead is then the difference in the sum ($T_{setup} + T_{CQ}$) between the original flip-flop and modified sized-up design. Table I lists the overheads associated with each flip-flop variant. For delay values, we list both the absolute value (in ps) as well as in terms of the standard FO4 value. (For the library used in our analysis, a single FO4 delay was equal to 40.1ps).

In terms of SER tolerance, the qualitative difference of these redesigned flip-flops can be identified by observing the circuit response associated with them. In Figure 5, we plot the noise rejection curves corresponding to the four

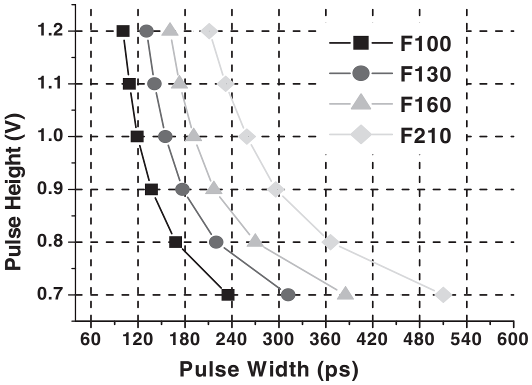


Fig. 5. Noise rejection curves for the flip-flop variants.

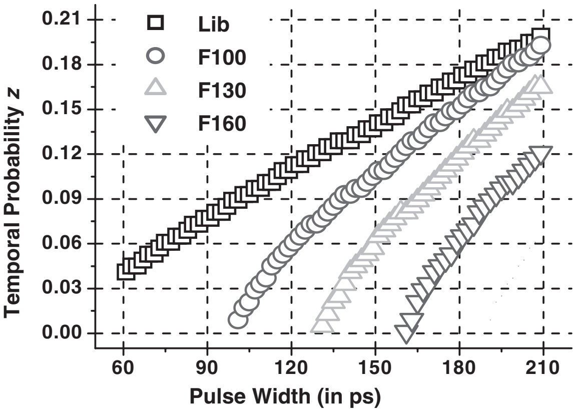


Fig. 6. Temporal probability values for the different flip-flop.

flip-flop variants. First, we confirm that at full V_{dd} (1.2V), the aforementioned filtering operation eliminates pulse widths below the associated flip-flop threshold value. In addition, for lower voltage magnitudes, the shape of the noise rejection curve ensures that an even larger fraction of the pulse widths are filtered by the flip-flop. Figure 5 shows that at pulse height = 1.0V, an *F100* flip-flop eliminates all pulses of widths less than 120ps from latching. In general, transient waveforms originating from deeper inside the combinational logic attain full V_{dd} magnitude before reaching the output. However, SETs that occur close to the output node are likely to consist of waveforms with pulse heights less than V_{dd}. The proposed flip-flops prove to be even more effective in handling these types of SETs.

We also examine the differences in temporal probability (z) values for the newly constructed flip-flops. Figure 6 plots the temporal probabilities for the case of rising pulses with height = 1.2V. First, since we only plot non-zero probability values, we do not show the fact that the z value associated with a flip-flop below its filtering threshold value is zero. Consequently, we exclude *F210* from this plot since the z values associated with them are zero. Next, we observe that for a given width above the filtering threshold associated with each flip-flop, the z value of the modified flip-flop is always lower compared to that of the base case library element. For instance, at pulse width = 120ps, *z(F100)* is about half of $z(Lib)$. The increased sizes in the flip-flops shrink the interval of possible time instances where the faulty bit can be latched into the flip-flop data input. Thus, we observe that in addition to the low-pass filtering mechanism, the upsized flip-flops also lessen the temporal probabilities appreciably, thereby producing a considerable reduction in the total SER of the circuit.

4.2 Slack-Based Flip-flop Assignment

The new flip-flop variants provide an effective option for circuit SER optimization since they do not modify any portion of the logic structure, instead focusing on filtering the faulty transients from being latched. Each variant incurs a certain amount of delay overhead such that FFs with better SER filtering incur larger overhead. In a standard logic circuit, each output node is connected to a standard library flip-flop. A simplistic method to use this FF library for SER mitigation is to replace each library flip-flop in a logic circuit with one of the new variants. However, such a replacement would impose a flat, delay overhead of at least 62.4ps (see Table I) which is not a viable option for most performance sensitive designs. A more effective method is to use the slack available at each output node and assign flip-flops appropriately.

The mathematical formulation of the slack-based FF assignment can be stated as follows: Each output node m is associated with an arrival time value of *AT(m)*. The circuit delay is set by the output node with the maximum value of *AT* so that:

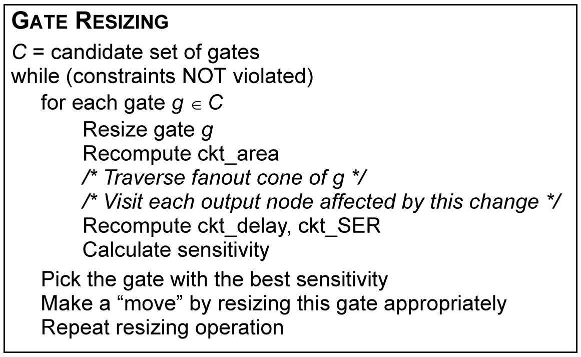
$$Delay = max\{AT(m)\} \quad m = [1, NumOutputs]. \tag{4}$$

The slack available at each output node is the difference between the delay of the circuit and the arrival time at that node.

$$Stack(m) = Delay - AT(m). \tag{5}$$

Depending on the value of slack, one of the flip-flop variants from Table I can now be assigned to each output node. For instance, for $0ps \leq Slack(m) < 62.4ps$, the *Lib* FF is assigned, while for $62.4ps \leq Slack(m) < 92.4ps$ the *F100* FF is assigned, and so on. In each case, the sum of arrival time at the output *AT(m)* and the overhead of the flip-flop variant is always lower than the initial specified value of *Delay* (EQ4). Thus, the worst case delay of the circuit remains unchanged.

This type of flip-flop assignment is best suited to circuits containing several outputs with significant slack. Given a circuit with a small number of critical paths all leading to a single output node, it is possible to assign all other output nodes to one of the flip-flop variants and achieve significant SER reduction. Note that the runtime for this reassignment is negligible since it only requires a single pass through the output nodes of the circuit.



```
GATE RESIZING
C = candidate set of gates
while (constraints NOT violated)
    for each gate $g \in C$
        Resize gate g
        Recompute ckt_area
        /* Traverse fanout cone of g */
        /* Visit each output node affected by this change */
        Recompute ckt_delay, ckt_SER
        Calculate sensitivity
    Pick the gate with the best sensitivity
    Make a "move" by resizing this gate appropriately
    Repeat resizing operation
```

Fig. 7. Pseudo-code for the proposed algorithm for gate resizing.

5. SENSITIVITY-BASED OPTIMIZATION

This section explains the proposed sensitivity-based SER optimization techniques. We first discuss various aspects of the sensitivity-based gate sizing algorithm, including the methods used for gate-specific SER bound calculation and candidate set selection through circuit pruning. We then present a joint approach combining flip-flop (FF) assignment with sizing to provide the best circuit solutions in terms of circuit SER.

5.1 Sensitivity-Based Gate Resizing

A large variety of circuit optimization algorithms in VLSI CAD use sensitivity-driven engines to guide the optimizer towards the best solution. Figure 7 presents pseudo-code for the proposed sensitivity-based gate resizing algorithm for SER minimization. We begin by developing an efficient bounding technique to prune the circuit graph and produce a candidate set of gates C consisting of cells that can potentially be resized for maximum SER improvement. We then define a sensitivity metric to maximize SER gains while limiting area overhead. Efficient sensitivity calculations are a crucial aspect of any circuit optimization algorithm. In our approach, we pick a single gate with the best sensitivity value and make the appropriate sizing move on this gate. Concurrently, it is also possible to use the sensitivity information of all gates in a more complex nonlinear optimizer that performs multiple, simultaneous gate sizing moves to achieve the optimal SER value.

5.1.1 *Candidate gate selection.* The selection of gates for the candidate set C significantly influences the performance of the proposed approach. In a non-ideal case, each gate in the circuit must be considered as a potential candidate for resizing. However, by identifying certain important characteristics related to the optimization metric we efficiently compute bounds on the SER value allowing for a subset of gates to be inserted into C.The SER bounds computation ensures that the circuit graph is pruned sufficiently to keep C relatively small.

The contribution of an individual cell to the total circuit SER is determined by various factors such as cell size, cell output load, input state probabilities,

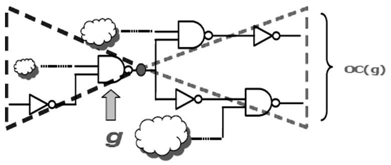


Fig. 8. Fanin and fanout cones associated with a gate g and the definition for $OC(g)$, the output count of g.

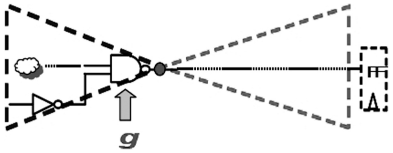


Fig. 9. Calculation of $SER(g)$. Fanout cone of g is disconnected and g is assumed to be directly connected to an output FF.

size of fanin/fanout cones, and depth from the output nodes. Since logic gates across a circuit vary significantly in these parameters, the relative contribution of individual gates to the total circuit SER can vary by as much as three orders of magnitude. This point shows that only a small fraction of the gates affect the circuit SER significantly. Therefore, the candidate set needs to be chosen carefully such that performing resizing on only this smaller set of gates provides the maximum amount of SER improvement.

To perform this selection, we first define new parameters *OC(g)*, *SER(g)* and *RedRatio(g)* for each gate g as follows: Each gate has fanin and fanout cones associated with it. As illustrated in Figure 8, *OC(g)* counts the number of outputs to which g is connected to in its fanout cone. Every gate g contains the set of descriptors due to all SETs that originate in the fanin cone of g and a single SET descriptor due to a strike on g itself. Suppose we disconnect the entire fanout cone of g and treat g as an output node (see Figure 9). *SER(g)* corresponds to this case when g is connected directly to a flip-flop. In the actual circuit, as the transient waveforms propagate in the fanout cone of g, *SER(g)* can only be reduced due to logical and electrical masking mechanisms. For instance, consider a single path from g to an output node that is b levels away from g. Let p_i for $i = [1,b]$ be the logical probabilities associated with each gate in this path. The SER value due to SETs propagating through this path will be $(\prod_{i=1}^{b} p_i)SER\,(g)$. In this expression, since each $p_i \leq 1$, we obtain the following inequality.

$$\left(\prod_{i=1}^{b} p_i\right) SER\,(g) \leq SER\,(g). \tag{6}$$

SER(g) therefore represents an approximate upper bound on the SER contribution of g at a single output node in the fanout cone of g. Note that this relation is independent of the correlation characteristics of the logical probabilities along the path.

Since gate g can affect several output nodes in its fanout cone we calculate *(SER(g)*OC(g))* and see that this product is an upper bound on the relative contribution of the fanin cone of g to the total circuit SER. Given the total circuit SER (*TotalCktSER*), we then define *RedRatio(g)* as:

$$RedRatio\,(g) = \frac{(SER\,(g) \times OC\,(g))}{TotalCktSER}. \tag{7}$$

Any subsequent sizing operation on gate g will, at best, completely eliminate the SER contribution of g and its entire fanin cone and reduce the total circuit SER by at most *(SER(g)*OC(g))*.

In this formulation we do not completely account for the effects of reconvergence on the SER. Our analysis methodology (presented in Section 3.3) is based on a static, block-based, linear-time algorithm that computes circuit SER using parameterized descriptors and a precharacterized cell library. In our prior work, we have shown that when not accounting for the effects of reconvergence, the computed SER value represents a conservative upper bound on the actual circuit SER. Although previous research from Zhang and Shanbhag [2004] has shown that the effects of reconvergence on SER is small, it is nevertheless important to recognize that the computed SER is an upper estimate on the exact circuit SER. Consequently, the results of the SER optimization algorithms also represent a conservative over estimate on the actual circuit SER value. In general, accounting for reconvergence is a computationally intensive proposition for static, block-based algorithms. Recent work from Krishnaswamy et al. [2008b] has proposed an interesting approach to address reconvergence but has the potential to be computationally expensive due to it's usage of input vector selection and path enumeration. A simple approximation to account for reconvergence can be included in the analysis by performing an initial pass on the fanout cone to determine the exact amount of magnification that reconvergent fanouts cause to *SER(g)*.

Next, we specify a minimum reduction ratio (*mrr*) value in order to prune gates and construct the candidate gate set. For each gate g, we add g into C only if *RedRatio(g)* > *mrr*. For instance, with $mrr = 1\%$, we do not add any gates into C that will, at best, give SER improvement of <1%. All gates in C are not guaranteed to give an improvement of at least 1%; Instead the 1% figure represents the minimum *potential* gains and not the actual gains in SER. Since SER values vary dramatically across the gates in a circuit, this pruning operation is very efficient in removing all gates that produce little or no improvement on the circuit SER. For the gates that are added to the candidate gate set, we perform sensitivity computations as explained in the next subsection. In practice, we find that using $mrr = 1\%$ prunes out a large fraction of the gates and only 10–20% of gates are typically considered for sizing.

5.1.2 *Structure of the Algorithm.* In our analysis, we consider three major circuit parameters—delay, SER and area—as the variables during sizing. For each cell, we first extract delay arcs from a standard timing library and define circuit delay as the maximum of the arrival times across all output nodes. The definition for cell SER was specified in the previous sub-section. We define cell area as the sum of device widths of all transistors in the gate and circuit area as the sum of areas of all cells. In our work, we focus only on the overhead aspect when resizing any given gate. While this definition of area is simplistic, we believe that it efficiently characterizes the overheads introduced during a sizing operation. Further, the device widths of the transistors are directly related to the total effective capacitance and hence, the total power dissipated by the cell. Thus, this definition of circuit area correlates fairly accurately with the total power consumed by the circuit.

The algorithm proceeds by picking each gate $g \in C$ in turn, perturbing the circuit by resizing this gate, and then recomputing the circuit delay, area and SER for this perturbed circuit. An important requirement for any sensitivity-based algorithm is the ability to perform incremental recomputation. In other words, by perturbing only a small portion of the circuit, we must not be required to perform a complete recomputation over the entire circuit. The change in circuit area is easy to quantify since local changes in cell area are reflected globally as well. For delay and SER, in our approach when any gate g is resized, we only consider the fanout cone of g while recomputing these parameters. Due to the modified size of g, the output capacitance seen by the immediate fanins of g is affected so that both delay and SER of g are altered.

To recalculate the new circuit delay and SER, we need to propagate the new arrival times and SET descriptors along the fanout cone until we reach an output node. However, during delay recalculation we frequently observe that after a few propagations, we encounter a path with greater arrival time so that further propagation along the cone for the new arrival time is unnecessary. This occurs because, in general, a vast majority of the gates are not critical and have no impact on circuit delay. Similarly, when propagating SET descriptors further along the circuit, a complete recalculation over the entire fanout cone is not required and propagation for at most 4–5 stages is sufficient. To detect cases of zero SER change due to a perturbation, we check that both the waveform shape and SER value of the descriptors are identical since both these factors impact circuit SER.

5.1.3 *Sensitivity Measurement.* After circuit parameters are recalculated, we perform a sensitivity measurement to determine the relative merits of each sizing move. First, we disregard all moves that worsen circuit performance and only consider cases where the circuit delay is equal to (or less than) the initial value. Next, since we seek to minimize area overhead while maximizing SER improvement we define the sensitivity as follows:

$$Sensitivity = \frac{\triangle SER}{\triangle Area} = \frac{SER_{original} - SER_{perturbed}}{Area_{perturbed} - Area_{original}} \tag{8}$$

We only consider cases where SER improves ($\Delta SER > 0$) and prioritize cases where gates are downsized ($\Delta Area < 0$) over those involving upsizing ($\Delta Area > 0$). As a delay constraint, we limit the total circuit delay to the initial delay point of the circuit. Thus, gates on critical paths are resized for SER improvement only if they also result in a delay improvement. We also impose an area constraint to avoid instances where circuit area increases significantly for marginal gains in SER. Note that the sensitivity measurement presented here intrinsically accounts for the dependence of both delay and SER on the capacitive load of the resized gate.

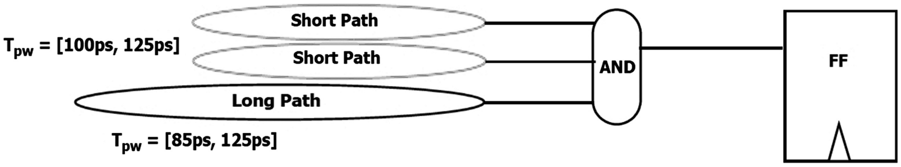


Fig. 10. Multiple short paths and a single long path connected to the output node. The pulse width range for the short paths is [100ps, 125ps] and the range for the long path is [85ps, 125ps]. Joint optimization enables significant SER reduction for this case.

5.1.4 *Algorithm Complexity.* The candidate gate selection mechanism significantly prunes the circuit and typically produces a subset containing at most 10–20% of the gates. The incremental recomputation method for delay and SER decreases the runtime further by eliminating the need for full recalculation over the entire circuit graph. In the worst case, the runtime per iteration can still be O(n^2); however, in practice, we find that it is significantly better than this bound. The total runtime for the algorithm depends mainly on the number of gates j that are resized, and is not directly influenced by the size of the circuit. Further, we impose additional constraints on the area and delay of the circuit so that the number of sizing moves is limited, making j a very small fraction of the total circuit size. The inclusion of such stopping criteria also ensures that the algorithm converges more quickly. The worst case complexity of the entire algorithm is given by *O(jn*^{2}*)*. Runtimes shown in the results section indicate that even for the largest circuit with ∼5000 gates, the total runtime is at most 200 seconds.

5.2 Combined FF Assignment + Gate Sizing

The combined optimization approach uses the electrical masking advantages of gate sizing and the temporal masking properties of the redesigned flip-flops to achieve large SER reductions. In the cooptimization approach, three factors help reduce the total circuit SER. The characteristics of the slack-based FF assignment and simple gate sizing have been described previously in the previous sections. In addition, gate sizing may also create slack at an output, leading to a better choice for the flip-flop variant.

We illustrate this effect using the example shown in Figure 10. Suppose a flip-flop contains multiple short paths and a single long path in its fanin cone. The pulse width ranges for the transient glitches corresponding to these paths are shown in the plot. First, note that simple flip-flop selection as presented in Section 4.2 will not be possible because the presence of the long path imposes only a small amount of slack at the output. Second, although gate sizing (Section 5.1) is possible, it may not produce vast reductions in the circuit SER due to the electrical characteristics associated with this gate. In other words, resizing this AND3 gate could possibly result in only a small filtering of the transients arriving at this gate.

On the other hand, upsizing this cell potentially reduces the delay along the paths through the gate. For instance, suppose the sizing operation modifies the delay of the long path such that the slack at the output node changes from 80ps

```
COMBINED FF ASSIGNMENT AND GATE RESIZING
/* Assign FFs initially based on available slack */
C = candidate set of gates
while (constraints NOT violated)
    for each gate g ∈ C
        Resize gate g
        Recompute ckt_area
        /* Traverse fanout cone of g */
        /* Visit each output node affected by this change */
        /* If slack changed, assign new FF variant */
        Recompute ckt_delay, ckt_SER
        Calculate sensitivity
    Pick the gate with the best sensitivity
    Make a "move" by resizing this gate appropriately
    /* Change FF assignments if necessary */
    Repeat resizing operation
```

Fig. 11. Pseudo-code for the proposed algorithm combining flip-flop assignment with gate resizing.

to 100ps. From Section 4.2 and Table I, we recognize that the flip-flop at this output can be changed from *F100* to *F130* (without affecting delay), thereby filtering all the pulses incident at the output node and obtaining even larger SER improvement. Thus, even if the SET waveforms at an output node are not affected due to the resizing of a candidate, the change in slack value at the gate can result in significant SER reductions due to the ability to reassign the flip-flop. This concept of slack creation amplifies the usefulness of the combined optimization approach. Further, a small set of gates in each circuit enable both slack creation and SET waveform reduction such that the synergy between the two techniques produces considerable reductions in the total SER of the circuit.

The structure of the algorithm (see Figure 11) is similar to the one presented in Section 5.1. We first perform an initial pass on the output nodes and assign flip-flops according to the slack availability. For each output node the set of gates on the critical path to this node can significantly affect the slack produced at the node. We recognize the potential gains offered by these gates by augmenting the candidate set C with the cells on the critical path to each output node. After an incremental recomputation of circuit delay, we visit all output nodes whose arrival times are affected and modify the output flip-flop to the appropriate type. During the sensitivity calculation in Figure 11, changes in SER due to both sizing and flip-flop assignment are reflected in the total SER value. At the end of a single resizing move, we update the flip-flop assignment appropriately. Note that the complexity of this combined algorithm is identical to the complexity gate resizing algorithm (see Section 5.1.4).

The unified optimization method is expected to provide better SER reduction than either FF assignment or gate sizing considered separately. The additional sizing step after FF assignment further targets the gates contributing most to the total circuit SER. Moreover, compared to the sizing-only optimization method, a smaller fraction of gates need to be resized since the flip-flop variants significantly filter out a large portion of the output transient waveforms.

6. RESULTS

The proposed algorithms were implemented in C++ and run on a dual-processor AMD Opteron 2.4GHz machine with 4GB RAM running Linux. We used an industrial 0.13μm standard cell library consisting of four sizes of inverters, NANDs, and NORs. All SER measurements were performed assuming a sea-level neutron flux of $56.5\text{m}^{-2}\text{s}^{-1}$. We employ three sets of benchmark circuits in our analysis: the ISCAS-85 suite (the c* circuits) Brglez and Fujiwara [1985], the MCNC circuit set (the i* circuits) [Yang 1991] and standard multiplier circuits (the m* circuits).

The flip-flop-based optimization approaches proposed in this paper rely on the amount of slack available to optimize the circuit SER. While it is possible to construct circuits with zero slack and completely well balanced paths, we believe that, in practice, such circuits are not commonly found. Warnock et al. [2002] and Curran et al. [2001] report the path delay distribution for a recent IBM microprocessor and show that even after optimization using traditional techniques (e.g., transistor sizing), only a small number of paths have delays within the vicinity of the critical path delay. Moreover, in accounting for other design decisions such as reducing overall power consumption or reducing the susceptibility to the effects of process variability, circuit designers, in general, do not produce designs with equalized path delay for microprocessor pipelines. Thus, leveraging circuit slack to achieve SER protection remains an attractive proposition.

To provide an accurate assessment of the proposed approaches, it is necessary to quantify the SER improvement for circuits with different amounts of slack. We therefore synthesize each benchmark for two separate delay constraint values: a tight delay constraint circuit (TDCC) corresponding to a 5% backoff from the fastest possible circuit implementation and a loose delay constraint circuit (LDCC) corresponding to a 30% backoff point. Current CMOS designs are severely limited by the amount of power that they dissipate so that the usage of circuits with loose delay constraints (20–30% backoff) has become more prevalent to meet the power budget. Moreover, the 5% backoff point is fairly aggressive since it is typically beyond the knee of the power/delay curve and as such represents a highly constrained design.

Circuits with tighter delay constraints will naturally contain a substantial number of sized up gates. However, due to the higher fraction of large gates, the number of locations at which SETs are injected is reduced thereby producing a lower value for the overall circuit SER. In Table II we list the circuit SER (with the FIT rates scaled by 1E-05) and the circuit area (with units in microns since area is defined as the sum of all device widths) for both LDCCs and TDCCs. On average, TDCCs have roughly half the SER while the area is doubled. Table II also includes the number of primary outputs (POs) for each circuit. In the subsequent analysis, we measure overheads in delay, area and SER from this initially specified design point for each type of circuit.

We label the three proposed optimization techniques as (T1) Gate sizing only (Section 5.1), (T2) Slack-based FF assignment (Section 4.2), and (T3) Combined FF assignment and gate sizing (Section 5.2). Table III first demarcates

Table II. Comparison of baseline loose/tight delay constraint circuits. Ckt SER has FITs scaled by 1E-05s

		LDCC			TDCC		
Ckt	POs	Gates	Ckt SER	Ckt Area	Gates	Ckt SER	Ckt Area
i1	13	60	1.5	215.0	85	0.9	692.6
i2	1	222	0.1	1081.2	307	0.1	3548.6
i3	6	132	0.2	770.0	144	0.2	2182.9
i4	6	264	0.2	2254.8	312	0.2	3691.4
i5	66	287	6.9	1542.2	723	3.0	4819.0
i6	67	734	15.0	2575.7	783	2.6	6175.5
i7	67	943	8.0	3683.5	1000	0.8	8986.1
i8	81	1610	15.6	6077.6	1919	5.9	13364.3
i9	63	1026	10.9	3597.2	1172	1.1	9684.1
i10	224	3393	30.5	10730.8	3663	24.3	17928.6
c432	7	247	0.3	1144.2	279	0.1	2211.1
c499	32	750	1.9	4750.4	826	0.1	8554.2
c880	26	608	3.2	2295.3	768	2.1	4901.4
c1355	32	741	1.5	3836.5	774	0.2	7363.3
c1908	25	753	4.0	3720.5	859	1.8	6915.4
c3540	22	1950	2.6	7608.2	2124	1.7	14077.3
c6288	32	5216	4.7	25788.7	6117	4.2	46600.1
m8x8	16	1334	3.3	6856.4	1543	2.1	12841.4
m16x16	32	6217	7.9	33382.4	7234	5.2	57857.8
Avg			6.2	6416.4		3.0	12231.3

the LDCCs from TDCCs. For each type of baseline circuit, we apply the three proposed techniques and quantify the reduction ratio (between the baseline SER and the optimized SER), relative increase in circuit area, number of gates resized, and algorithm runtime. Recall from earlier discussions that there is no delay penalty and the maximum area penalty is set to 20%. Since T2 is a simple FF assignment algorithm that does not involve any modification of the gates, the area increase and number of gates resized is 0, and the runtime related to this reassignment is negligible.

The circuit delay tightness plays an important part in determining the performance of all three optimization techniques. For a TDCC, a larger fraction of gates are on critical or near critical paths, so that a particular resizing move on a specific gate may be disallowed since it violates delay constraints. On the other hand, for LDCCs a large number of gates have no impact on circuit delay and can be resized to achieve SER savings. Thus, comparing SER reductions for the two types of circuits by the application of T1, we observe that circuit SER is reduced on average by 10.7X in a LDCC versus 8.2X in a TDCC. However, since the baseline SER of a TDCC is lower, the final FIT rate of the optimized TDCC, despite the smaller amount of SER reduction, will be less than the final FIT rate of an optimized LDCC. The larger number of critical paths also implies that the arrival times at several output nodes will be nearly identical. Hence, the amount of slack at each output node is small which lowers the gains offered by T2. On average, T2 produces SER reductions of 3.1X in a TDCC compared to 5.8X in a LDCC. However, since T2 is a technique that consumes zero area and delay overhead, it is still an attractive alternative due to its simplicity.

Table III. SER Reduction, area change (%), number of resized gates, and runtimes for the three optimization techniques, T1 = gate sizing only, T2 = slack-based FF assignment, T3 = combined FF assignment and gate sizing

Ckt	Loose Delay Constraint Circuits (LDCC)									Tight Delay Constraint Circuits (TDCC)								
	SER(base)/SER(opt)			% Area change		# Resized gates		Runtime (s)		SER(base)/SER(opt)			% Area change		# Resized gates		Runtime (s)	
	T1	T2	T3	T1	T3	T1	T3	T1	T3	T1	T2	T3	T1	T3	T1	T3	T1	T3
i1	15.5X	2.2X	54.9X	20.4	13.8	21	11	0.3	0.2	16.6X	5.5X	8.9X	10.4	1.6	15	9	0.1	0.1
i2	31.7X	1.0X	31.7X	0.5	0.5	1	1	0.2	0.2	1.0X	1.0X	1.0X	0	0	1	1	0.0	0.0
i3	17.5X	4.6X	34.0X	2.6	2.6	4	4	0.2	0.2	35.5X	1.6X	38.1X	0.9	0.9	4	4	0.0	0.0
i4	17.1X	8.5X	18.6X	0.9	0.2	4	2	0.4	0.2	30.3X	10.6X	35.1X	0.5	0.1	4	2	0.0	0.0
i5	10.8X	1.4X	38.9X	18.3	15.6	58	40	4.2	14.2	5.1X	1.5X	6.7X	3.3	2.5	44	48	3.7	11.5
i6	4.2X	1.6X	13.9X	7.1	7.1	117	116	8.3	5.6	2.0X	1.0X	2.0X	2.2	1.4	44	44	1.0	0.9
i7	4.6X	2.1X	12.8X	9.0	5.9	105	79	20.2	19.9	2.7X	1.2X	3.0X	0.5	0.5	16	16	0.7	0.6
i8	5.7X	2.1X	8.9X	5.3	3.0	156	110	42.7	37.1	4.1X	4.0X	10.8X	0.1	0.0	66	40	8.1	7.7
i9	5.4X	1.2X	14.7X	19.0	15.3	143	123	22.2	22.0	3.5X	1.0X	4.4X	1.1	0.8	26	20	2.8	1.9
i10	3.2X	19.5X	34.0X	8.8	1.2	199	33	131.6	116.4	2.8X	5.5X	30.6X	3.7	0.3	177	31	81.4	77.9
c432	6.0X	1.1X	12.8X	2.3	2.3	5	5	0.8	0.7	5.6X	1.2X	12.1X	0.2	0.2	2	2	0.2	0.2
c499	11.4X	2.0X	17.5X	0.8	0.6	55	44	9.4	8.5	1.0X	1.0X	1.3X	0.0	0.1	0	2	0.1	1.1
c880	7.1X	9.7X	42.5X	9.0	3.2	52	16	19.5	6.9	6.7X	1.2X	35.6X	1.5	0.3	32	6	6.9	2.2
c1355	3.7X	2.0X	9.2X	0.9	0.3	33	27	9.1	7.9	1.1X	1.0X	4.2X	0.1	0.1	2	2	0.1	2.2
c1908	23.7X	2.4X	26.7X	5.3	2.2	73	18	15.0	9.8	10.6X	2.3X	43.4X	1.6	1.1	43	29	4.8	4.8
c3540	8.8X	5.8X	57.3X	1.2	0.3	33	8	51.7	15.2	8.6X	4.7X	27.4X	0.5	0.1	22	4	17.7	4.9
c6288	10.4X	26.3X	30.9X	1.1	0.1	73	4	195.0	17.6	7.6X	6.2X	28.9X	0.6	0.1	74	12	98.3	33.4
m8x8	11.2X	6.5X	47.2X	2.2	0.8	45	11	25.8	8.3	5.2X	3.2X	41.4X	0.7	0.4	27	11	6.8	5.0
m16x16	5.7X	9.4X	66.3X	1.1	0.4	78	18	111.8	48.5	5.2X	6.1X	30.9X	0.3	0.2	58	19	83.4	48.9
Avg	10.7X	5.8X	30.1X	6.1	4.0					8.2X	3.1X	19.3X	1.5	0.6				

The slack creation concept described in Section 5.2 plays an important role in reducing the SER particularly in the TDCCs. We observe here that in general, T3 produces significantly more reductions compared to T1 while resizing a fewer number of gates. This effect is primarily due to the ability for T3 to identify slack-critical gates in the design. The sensitivity metric corresponding to such gates is particularly high given the possibility of achieving even greater gains by reassigning an output flip-flop. Thus, generating enough slack by sizing even a small number of gates produces significant gains in the circuit SER value.

On average, the combined SER optimization method, T3, outperforms both T1 and T2 with average savings of up to 30.1X for LDCCs and 19.3X for TDCCs. The number of gates resized and, consequently, the area overhead using T3 is always lower than for T1. Furthermore, T3 runtime is also smaller than T1. Although we limit area overhead to 20% we observe that in most cases the area increases by a much smaller amount (about 5–6%) and at most 200–250 gates in the entire circuit are resized. The runtimes for both T1 and T3 are quite small and on the order of 1–3 minutes.

Circuit configuration and gate structure also play an important role in determining circuit SER and the effectiveness of the techniques presented in our work. While analyzing circuit SER, we indicate in Rao et al. [2006] that the number of outputs is a critical component in determining SER since a larger output count implies a higher degree of observability for possible transient pulses. In a similar vein, we can now observe that if the number of outputs is small then it is possible to achieve large reductions in SER for small investments in area/power by targeting the select few nodes that influence those outputs. This effect is best illustrated by examining circuit *i2*. The synthesized version of cell *i2* is composed of a multiplexor-like circuit structure where a large fan-in cone (composed of 201 primary inputs) is connected to a chain of inverters that drive a single primary output node. To achieve significant, cost-effective SER reductions, the algorithm modifies one of the inverters in the output chain such that this resizing operation eliminates a large fraction of the transient pulses from reaching the output and the delay is maintained at a value similar to the original circuit. While this process is similar to Dhillon et al. [2005], the large savings are primarily due to the specific gate structure present in this circuit. Due to the presence of only a single primary output, both circuit delay and circuit SER are entirely determined by the characteristics of the gates connected directly to this output node. As a result, resizing a single gate produces substantial reductions in circuit SER. A second example of the effect of output count on SER optimization can be obtained by comparing cells *c1908* and *i6*. We see that they have very similar gate counts *i6* has more than 2.7X more outputs than *c1908*. The SER reduction in *c1908* is as much as 23.7X while we achieve only 4.2X reduction in *i6*.

Although architectural-level SER estimation methods such as Mukherjee et al. [2006] assume a single average value for the SER per output bit while considering the logic in pipeline stages, we have observed in Rao et al. [2006] that this can be an inaccurate assumption by showing a range of more than 100X in the SER values of the different outputs of a single circuit. Zhang and Shanbhag [2004] also report a similar SER peaking phenomenon for specific individual bits in an 8-bit multiplier. Since the techniques presented here inherently consider such SER disparities while taking into account circuit delay/area, they are ideally suited to achieve SER protection. For instance, if cell resizing is undesirable or unavailable as an option, a simple, practical alternative is to identify the output bit(s) contributing most to the circuit SER and apply technique T2 on them based on the slack available in the system.

7. CONCLUSIONS

In this work, we first described the method by which a library flip-flop can be redesigned using transistor sizing to improve its SET resiliency. This type of redesign leverages the effects of temporal masking by widening the flip-flop pulse filtering window. We then presented novel soft error rate optimization techniques for combinational circuits. These involve a slack-based flip-flop assignment method, a sensitivity-based gate sizing algorithm, and a joint optimization approach combining flip-flop assignment with gate sizing into a single algorithm. We explored the effectiveness of these methods for circuits synthesized at different delay constraints. Depending on the amount of slack available in the circuit and the amount of area overhead that is tolerable, we can choose between the three techniques to achieve the best circuit solution. Experimental results show SER reductions of up to 30.1X while accruing an area overhead of $\sim$4–6% and no delay penalties.

REFERENCES

ABRISHAMI, H., HATAMI, S., AMERLIFARD, B., AND PEDRAM, M. 2008. Characterization and design of sequential elements to combat soft errors. In *Proceedings of the International Conference on Computer Design (ICCD)*. 194–199.

ALMUKHAISM, S., MAKRIS, Y., YANG, Y., AND VENERIS, A. 2006. Seamless integration of SER in rewiring-based design space exploration. *International Test Conference (ITC)*. 1–9.

BAUMANN, R. 2005. Soft errors in advanced computer systems. *IEEE Des. Test Comput. 22*, 3, 258–266.

BRGLEZ, F. AND FUJIWARA, H. 1985. A neural netlist of ten combinational benchmark circuits and translator in Fortran. In *Proceedings of the International Symposium on Circuits and Systems (ISCAS)*. 663–698.

CHA, H. AND PATEL, J. 1994. Latch design for transient pulse tolerance. In *Proceedings of the International Conference on Computer Design (ICCD)*. 385–388.

CHOUDHURY, M., ZHOU, Q., AND MOHANRAM, K. 2006. Design optimization for single-event upset robustness using simultaneous dual-VDD and sizing techniques. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 204–209.

CURRAN, B., CAMPORESE, P., CAREY, S., YUEN, C., CHAN, Y., CLEMEN, R., CREA, R., HOFFMAN, D., KOPROWSKI, T., MAYO, M., MCPHERSON, T., NORTHROP, G., SIGAL, L., SMITH, H., TANZI, F. AND WILLIAMS, P. 2001. A 1.1 GHz first 64 b generation Z900 microprocessor. In *Proceedings of the International Solid-State Circuits Conference (ISSCC)*. 238–239.

DHILLON, Y., DIRIL, A. AND CHATTERJEE, A. 2005. Load and logic co-optimization for design of soft-error resilient nanometer CMOS circuits. In *Proceedings of the International Online Testing Symposium (IOLTS)*. 35–40.

ELAKKUMANAN, P., PRASAD, K., AND SRIDHAR, R. 2006. Time redundancy based scan flip-flop reuse to reduce SER of combinational logic. In *Proceedings of the International Symposium. on Quality Electronic Design (ISQED)*. 617–622.

FACCIO, F., KLOUKINAS, K., MARCHIORO, A., CALIN, T., COSCULLUELLA, J., NICOLAIDIS, M., AND VELAZCO, R. 1999. Single event effects in static and dynamic registers in a 0.25μm technology. *IEEE Trans. Nucl. Sci. 46*, 6, 1434–1439.

Fishburn, J. and Dunlop, A. 1985. TILOS: A posynomial programming approach to transistor sizing. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 326–328.

Freeman, L. 1996. Critical charge calculations for a bipolar SRAM array. *IBM J. Resear. Devel. 40*, 1, 77–89.

Garg, R., Jayakumar, N., Khatri, S., and Choi, G. 2006. A design approach for radiation-hard digital electronics. In *Proceedings of the Design Automation Conference (DAC)*. 773–778.

Hazucha, P. and Svensson, C. 2000. Impact of CMOS technology scaling on atmospheric neutron soft error rate. *IEEE Trans. Nucl. Sci. 47*, 6, 2586–2594.

Joshi, V., Rao, R. R., Blaauw, D., and Sylvester, D. 2006. Logic SER reduction through flipflop redesign. In *Proceedings of the International Symposium on Quality Electronic Design (ISQED)*. 611–616.

Karnik, T., Bloechel, B., Soumyanath, K., De, V., and Borkar, S. 2001. Scaling trends of cosmic ray induced soft errors in static latches beyond 0.18μm. In *Proceedings of the Symposium on VLSI Circuits*. 61–62.

Karnik, T., Vangal, S., Veeramachaneni, S., Hazucha, P., Erraguntla, V., and Borkar, S. 2002. Selective node engineering for chip-level soft error rate improvement. In *Proceedings of the Symposium on VLSI Circuits*. 204–205.

Krishnaswamy, S., Markov, I., and Hayes, J. P. 2008a. On the role of timing masking in reliable logic circuit design. In *Proceedings of the Design Automation Conference (DAC)*. 924–929.

Krishnaswamy, S., Viamontes, G., Markov, I., and Hayes, J. 2008b. Probabalistic transfer matrices in symbolic reliability analysis of logic circuits. *Trans. Des. Autom. Electron. Syst. 13*, 1, 8.

Maestro, J.A., and Revieriego, P. 2008. Study of the effects of MBUs on the reliability of a 150 nm SRAM device. In *Proceedings of the Design Automation Conference (DAC)*. 930–935.

Magen, N., Kolodny, A., Weiser, U., and Shamir, N. 2004. Interconnect power dissipation in a microprocessor. In *Proceedings of the International Workshop on System Level Interconnect Prediction (SLIP)*. 7–13.

Mavis, D. and Eaton, P. 2002. Soft error rate mitigation techniques for modern microcircuits. In *Proceedings of the International Reliability Physics Symposium. (IRPS)*. 216–225.

Mitra, S., Karnik, T., Seifert, N., and Zhang, M. 2005. Logic soft errors in sub-65nm technologies design and CAD challenges. In *Proceedings of the Design Automation Conference (DAC)*. 2–3.

Mitra, S., Seifert, N., Zhang, M., Shi, Q., and Kim, K. 2005. Robust system design with built-in soft-error resilience. *IEEE Comput. 38*, 2, 43–52.

Mitra, S., Zhang, N., Waqas, S., Seifer, N., Gill, B., and Kim, K. 2006. Combinational logic soft error correction. In *Proceedings of the International Test Conference (ITC)*. 1–9.

Moharam, K. and Touba, N. 2003. Cost-effective approach for reducing the soft error failure rate in logic circuits. In *Proceedings of the International Test Conference (ITC)*. 893–901.

Monnier, T., Roche, F., and Cathebras, G. 1998. Flipflop hardening for space applications. In *Proceedings of the International Workshop on Memory Technology*. 104–107.

Mukherjee, S., Weaver, C., Emer, J., Reinhardt, S., and Austin, T. 2003. A systematic methodology to compute the architectural vulnerability factors for a high performance microprocessor. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*. 29–40.

Omana, M., Rossi, D., and Metra, C. Latch susceptibility to transient faults and new hardening approach. *IEEE Trans. Comput. 56*, 9, 1255–1268.

Qian, D., Yu, W., Hui, W., Rong, L., and Huazhong, Y. 2008. Output remapping technique for soft-error rate reduction in critical paths. In *Proceedings of the International Symposium on Quality Electronic Design (ISQED)*. 74–77.

Rajaraman, R., Kim, J., Vijaykrishnan, N., Xie, Y., and Irwin, M. 2006. SEAT-LA: A soft error analysis tool for combinational logic. In *Proceedings of the International Conference on VLSI Design (VLSID)*. 499–502.

Ramakrishnan, K., Rajaraman, R., Vijaykrishnan, N., Xie, Y., Irwin, M. J., and Unlu, K. 2008. Hierarchical soft error estimation tool (HSEET). In *Proceedings of the International Symposium on Quality Electronic Design (ISQED)*. 680–683.

Ramanarayanan, R., Degalahal, V., Vijaykrishnan, N., Irwin, M., and Duarte, D. 2003. Analysis of soft error rate in flipflops and scannable latches. In *Proceedings of the International ASIC/SOC Conference*. 2321–234.

RAO, R. R., CHOPRA, K., BLAAUW, D., AND SYLVESTER, D. 2007. Computing the soft error rate of a combinational circuit using parameterized descriptors. *IEEE Trans. Comp.-Aid. Des. 25*, 13, 468–479.

SASAKI, Y., NAMBA, K., AND ITO, H. 2008. Circuit and latch capable of masking soft errors with Schmitt trigger. *J. Electron. Test. 24*, 1, 11–19.

SEIFERT, N. AND TAM, N. 2004. Timing vulnerability factors of sequentials. *IEEE Trans. Dev. Mater. Reliabil. 4*, 3, 516–522.

SHAZLI, S. Z., ABDUL-AZIZ, M., TAHOORI, M. B., AND KAELI, D. R. 2008. A field analysis of system-level effects of soft errors occurring in microprocessors used in information systems. In *Conference on International Test Conference (ITC)*. 1–10.

SHIVAKUMAR, P. AND KECKLER, S. 2006. Exploiting slack for low overhead soft error reliability. *Soft Errors in Logic—System Effects (SELSE)*.

SHIVAKUMAR, P., KISTLER, M., KECKLER, S., BURGER, D., AND ALVIS, L. 2002. Modeling the effect of technology trends on the soft error rate of combinational logic. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN)*. 389–398.

WARNOCK, J., KEATY, J., PETROVICK, J., CLABES. J., KIRCHER, J., KRAUTER, B., RESTLE, P., ZORIC, B., AND ANDERSON, C. 2002. The circuit and physical design of the POWER4 microprocessor. *IBM J. Resea. Develop. 46*, 1, 27–52.

WESTE, N. AND HARRIS, D. 2005. *CMOS VLSI Design: A Circuits and Systems Perspective*, Addison Wesley.

WU, K.C. AND MARCULESCU, D. 2008. Soft error rate reduction using redundancy addition and removal. In *Proceedings of the Asian South Pacific Design Automation Conference (ASPDAC)*. 559–564.

YANG, S. 1991. *Logic Synthesis and Optimization Benchmarks User Guide*, MCNC, Research Triangle Park, North Carolina.

ZHANG, B. AND ORSHANSKY, M. 2006. FASER: Fast analysis of soft error susceptibility for cell based designs. In *Proceedings of the International Symposium on Quality Electronic Design (ISQED)*. 755–760.

ZHANG, M. AND SHANBHAG, N. 2004. A soft error rate analysis (SERA) methodology. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 111–118.

ZHANG, M. AND SHANBHAG, N. 2005. An energy-efficient circuit technique for single event transient noise-tolerance. In *Proceedings of the International Symposium on Circuits and Systems (ISCAS)*. 636–639.

ZHANG, M. AND SHANBHAG, N. 2005. A CMOS design style for logic circuit hardening. In *Proceedings of the International Reliability Physics Symposium (IRPS)*. 223–229.

ZHAO, C., BAI, X., AND DEY, S. 2004. A scalable soft spot analysis methodology for compound noise effects in nano-meter circuits. In *Proceedings of the Design Automation Conference (DAC)*. 894–899.

ZHAO, C. AND DEY, S. 2006. Improving transient error tolerance using robustness compiler (ROCO). In *Proceedings of the International Symposium on Quality Electronic Design (ISQED)*. 133–138.

ZHOU, Q. AND MOHANRAM, K. 2004. Cost effective radiation hardening technique for combinational logic. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*. 100–106.

ZHOU, A., CHOUDHURY, M., AND MOHANRAM, K. 2008. Tunable transient filters for soft error rate reduction in combinational circuits. In *Proceedings of the European Test Symposium (ETS)*. 179–184.

ZIVANOV, N. M. AND MARCULESCU, D. 2006. MARS-C: modeling and reduction of soft errors in combinational circuits. In *Proceedings of the Design Automation Conference (DAC)*. 767–772.

ZIVANOV, N. M. AND MARCULESCU, D. 2006. Soft error rate analysis for sequential circuits. *Conference and Exhibition on Design Automation and Test in Europe (DATE)*. 1436–1441.

Received October 2008; revised March 2009, July 2009; accepted August 2009