

# Logic SER Reduction through Flipflop Redesign

Vivek Joshi\*, Rajeev R. Rao, David Blaauw, Dennis Sylvester

\*Indian Institute of Technology, Kanpur, India 208016

Department of EECS, University of Michigan, Ann Arbor, MI 48109

\*{vivekjj@iitk.ac.in}, {rrrao, blaauw, dennis}@eeecs.umich.edu

## Abstract

In this paper, we present a new flipflop sizing scheme that efficiently immunizes combinational logic circuits from the effects of radiation induced single event transients (SET). The proposed technique leverages the effect of temporal masking by selectively increasing the length of the latching windows associated with the flipflops thereby preventing faulty transients from being registered. We propose an effective flipflop sizing scheme and construct a variety of flipflop variants that function as low-pass filters for SETs and reduce the soft error rates (SER) of combinational circuits. In contrast to previously proposed flipflop designs that rely on logic duplication and complicated circuit design styles, our method provides a simple yet highly effective mechanism for logic SER reduction while incurring very small overheads in both delay (about 5 FO4) and power (about 5%). Experimental results at the circuit level on a wide range of benchmarks show 1000X reductions in SER for small increases in circuit delay and power.

## 1 Introduction

Single event upsets (SEU) arise from the interaction of energetic cosmic particles, such as neutrons, with the semiconductor material in integrated chips. Soft errors resulting from such transient upsets have grown into a critical reliability concern for VLSI circuits. The scaling of device feature sizes combined with shorter pipeline depths have resulted in a significant increase in the susceptibility of CMOS circuits to radiation induced soft errors. In a typical IC, memory arrays, latch elements and combinational logic are all susceptible to soft errors. A number of studies have examined the impact of these transient events on CMOS circuits in modern and future technology nodes [1][2][3]. In [1] the author illustrates that although the SRAM bit SER is expected to stay constant across technologies, the usage of error corrected codes (ECC) can be expected to decrease their impact. On the other hand, it has been predicted that logic SER will grow steadily across technology generations and become the dominant contributor to chip SER at the 45nm technology node [3]. This is particularly relevant for mainstream products (such as medium- and low-end servers) that contain large portions of unprotected logic on the chip. Thus it has become increasingly important to analyze SER for combinational logic circuits.

Logic SER is the cumulative effect of transient events on both sequential (data retention) elements, such as latches and flipflops, and on combinational logic elements that perform actual circuit computations. The impact of cosmic particle strikes that occur directly on flipflops is well understood and a number of methods have been proposed for both the analysis and mitigation of soft errors in these elements. Several approaches investigate the amount of critical charge that must be injected in order to change the state of the memory element. In [4] the authors present a survey of the impact of direct particle strikes on various types of flipflops and scannable latches. The authors of [5] present the effects of SEUs on static and dynamic registers. A number of latch designs have been proposed in the literature to mitigate the impact of soft errors in latches [6][7]. These so-called radiation hardened latches introduce additional circuit components (e.g. keepers, feedback resistors and high impedance MOS transistors) to enable the memory elements to retain state in the event of a particle strike.

On the other hand, the effect of particle strikes on combinational logic is a relatively unexplored topic. While a particle strike on a memory register can permanently change the memory state, a strike on the combinational logic will cause an error only if the transient pulse propagates to the register and is latched by it. For accurate SER analysis of combinational logic, it is important to consider three types of masking mechanisms. Logical masking occurs when a controlling input on a gate filters out all transient pulses propagating through the other inputs. Electrical masking occurs when the characteristics of the gate (such as size, output load) prohibit the propagation of transients further along the circuit. The time window near the clocking edge, when a spurious transition can potentially be latched by the output flipflop, is referred to as the window of vulnerability (WOV) [8]. Temporal masking occurs when the transient that has reached the output is outside the WOV and, hence, is not latched by the memory element. These masking methods serve as derating factors in that they reduce the overall probability that a cosmic particle strike will actually produce an erroneous output. While logical and electrical masking do result in the significant downgrading of combinational logic error rates, it has been observed that temporal masking is the dominant derating factor [2].

A variety of approaches have been proposed to leverage temporal masking to minimize the probability of faulty transients from latching into the registers. These methods are characterized by circuit schemes that use time and space redundancy such as double sampling flipflops [9] and dual ported latches [10]. The authors in [11] develop a technique to reuse existing testing resources for soft error protection. In [12] the authors propose the use of delayed versions of data and clock signals to ensure that the faulty signals are not latched. However, these schemes typically suffer from large overheads in performance, power and area. The use of redundancy introduces significant logic duplication which increases power consumption. Moreover, delayed clock signals incur large delay penalties and also impose stringent constraints for clock tree design. Furthermore, design styles using non-standard cell structures are difficult to implement and are incompatible with standard ASIC flows.

In this paper, we propose an efficient technique to minimize the soft error rates of combinational logic circuits. We first analyze the temporal masking behavior of a standard library flipflop in terms of its timing characteristics (setup/hold times) and the input transient pulse widths. Next, we illustrate the method for SER reduction using the modification of the latching window associated with the flipflop. We then present a novel sizing scheme for flipflops that modulates the sizes of a select few transistors and enables the construction of a variety of SER tolerant flipflops. Previously proposed SER tolerant designs warrant large overheads (as much as 300%) due to logic redundancy. In contrast, our design scheme incurs significantly lower overhead in power (about 5%) and delay (at most 5 FO4). We then utilize the large variety of flipflops to explore the tradeoffs between SER reduction and performance. Experimental results show that in the best case, we achieve 1000X reduction in SER values while incurring a delay overhead of 153ps. Because our method utilizes device sizing as a design variable, it constitutes a simple yet effective technique for soft error rate reduction that is amenable to inclusion in industrial cell library generation.

The remainder of this paper is organized as follows. In Section 2, we discuss various aspects of the temporal masking mechanism. In Section 3, we discuss sizing strategies for flipflops and pulsed latches. In Section 4, we employ the proposed approach in a wide variety of combinational circuits and present the SER reductions that can be achieved using our modified flipflop library. Section 5 concludes the paper.

## 2 Temporal Masking Mechanism

In this section, we first present the relation between the transient pulses and the setup/hold times associated with the flipflop. We then present an overview of the logic SER estimation tool used in our analysis. Finally, we motivate the concept of increasing the setup/hold times of the flipflop to achieve significant amounts of SER reduction.

### 2.1 Aperture Window Analysis

Current digital circuit designs employ a host of memory sequencing elements such as edge-triggered flipflops, transparent latches and pulsed latches. While flipflops are clearly the most prevalent choice due to their simplicity, pulsed latches are used by ASIC designers who desire higher performance. In a static CMOS circuit, the outputs of a standard combinational circuit block are connected to sequencing elements. A particle strike on a particular node in the logic can propagate to a given output based on the input state of the circuit and the output observability of that particular node. Temporal masking analysis seeks to determine the probability of a transient pulse latching a faulty data value into the memory element.

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 [3] (or the aperture window in [13]) and 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 [13].  $T_{hold}$  is defined in similar fashion. The latch propagation delay  $T_{PCQ}$  is equal to the value of  $T_{CQ}$  when  $T_{DC}=T_{setup}$ ). 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 (skew and jitter).

Due to the presence of transient pulses, it is possible that a faulty data transition takes place close to the clock edge. Subsequently, the memory element can latch this erroneous value thereby resulting in a logical error. It is also possible that even if the correct value is latched, the transient pulse can cause a glitch in the memory element so that the  $T_{CQ}$  value, in this case, is greater than the target propagation delay  $T_{PCQ}$ . Such an event results in a delay fault. It has been estimated that delay faults are negligible in current technologies due to the use of safety margins and guardbanding in IC design [2]. We do not consider the impact of delay faults in our analysis and focus only on the cases when logical errors can occur.

We now illustrate the temporal probability calculation for a given waveform using Figure 1. In this plot, we fix the clock edge and vary

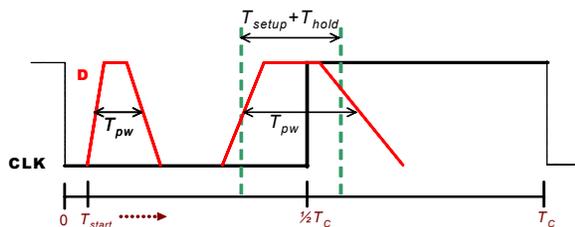


Figure 1. Temporal probability calculation for the case of non-switching input

the value of the start time  $T_{start}$  of two sample waveforms in a set of discrete values in the range  $[0, T_c]$ . For each waveform  $k$ , we tabulate the interval ( $I_k$ ) of  $T_{start}$  values for which a logical error occurs. Clearly, for  $T_{start} > T_c/2$  no error is latched by the flipflop. In the simplest case, if the transient pulse width ( $T_{pw}$ ) is greater than ( $T_{setup} + T_{hold}$ ), then the voltage pulse can completely overlap the aperture window and result in a logical error. If a pulse partially overlaps the aperture window then it is possible that a delay fault may occur at the output of the flipflop. We disregard such cases while computing the interval  $I_k$  of  $T_{start}$  values. Thus, it is evident that no logic error is possible when  $T_{pw} < (T_{setup} + T_{hold})$  and the probability of a soft error occurring due to that pulse is 0. We observe that the interval  $I_k$  over which a pulse is completely filtered is strongly correlated to the length of the aperture window; a good approximation for the  $I_k$  is given by the difference in the interval lengths between the pulse width and the aperture window.

$$I_k \approx T_{pw} - (T_{setup} + T_{hold}) \quad (\text{EQ 1})$$

The temporal probability  $z(k)$  of a flipflop latching a transient pulse can be expressed as follows:

$$z(k) \approx \begin{cases} 0 & T_{pw} < (T_{setup} + T_{hold}) \\ \frac{T_{pw} - (T_{setup} + T_{hold})}{T_c} & T_{pw} \geq (T_{setup} + T_{hold}) \end{cases} \quad (\text{EQ 2})$$

In Figure 1 we implicitly assumed that the flipflop input ( $D$ ) is of the non-switching type i.e.,  $D_{prev}=0$ ,  $Q_{prev}=0$  and because  $D_{next}=0$  the error-free value of  $Q_{next}$  should be 0 with the faulty rising transient possibly corrupting this data value. In the complementary case of switching inputs,  $D_{prev}=1$ ,  $Q_{prev}=1$  and with  $D_{next}=0$  the expected value of  $Q_{next}=0$ . In this instance, the rising transient waveform causes an erroneous value to be latched when it occurs close to the setup edge of the flipflop. Unlike the case for non-switching inputs, the transient pulse is not required to fully overlap the aperture window to cause a logic error. Since the previous state of the flipflop is already at 1, the SET is only required to be close enough to the setup time so that the new value is not registered by the data retention node of the flipflop. From this discussion, we observe that the probability of a logical error depends only on the location of the faulty pulse in the aperture window and is, therefore, independent of the lengths of the aperture window and transient pulse width.

Note that the temporal probabilities associated with the cases for non-switching and switching inputs are different; for each waveform  $k$ , we identify them separately as  $z_{ns}(k)$  and  $z_{sw}(k)$ . To accurately quantify the temporal probabilities for a given waveform, we therefore use the discrete interval based approach and determine the fraction of  $T_{start}$  values for which the waveform is captured at the flipflop output using SPICE measurements.

### 2.2 Logic SER Estimation Engine

We provide a brief overview of our gate level estimation tool used to determine the SER values of combinational logic circuits. In [14], we presented a static linear-time algorithm for SER estimation using the concept of SET descriptors. The current injection model from [15] is used to describe the transient pulse generated by a radiation

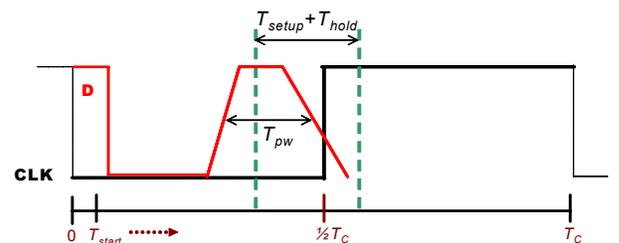


Figure 2. Temporal probability calculation for switching input

particle strike. A Weibull function based 3-tuple is used to describe the voltage glitch due to a single particle hit. While analyzing logic circuits for soft errors, it is important consider the entire spectrum of possible strikes. The authors in [16] show that neutron strikes of varying energy levels can be mapped to an injected charge range equal to [10fC, 150fC]. Each charge value is then mapped to a corresponding strike rate value using the empirical models proposed in [17]. For charge values greater than the 150fC, the corresponding SET strike probability values are negligible. To account for the cumulative effect of the entire range of possible neutron strikes at a node, we introduce the concept of an SET descriptor. Each descriptor consists of an identifying tag for the waveform shapes, a vector  $b$  of the waveform parameters and a vector  $R$  of rate values.

$$\text{SET\_Descriptor} = \{\text{Waveform\_tag, vector } b, \text{ vector } R\}$$

The algorithm proceeds in a bottom-up fashion by injecting transients along the nodes in a circuit. Since each particle strike is assumed to be an independent event, the waveform families and the corresponding SET descriptors are considered as independent instances. We propose efficient techniques for propagation and merging of these SET descriptors as they traverse along the nodes in the circuit. We also present a method to use `wave_tag` to identify near-identical waveform families thereby drastically reducing the total number of SET descriptors and ensuring linearity for the algorithm's complexity. The final output of the algorithm is a set of SET descriptors at each output node. This set of descriptors captures the cumulative effect of particle strikes at each node in the fan-in cone of that particular output bit.

Each output descriptor consists of a vector of values for Weibull parameter  $b$  and strike rate values  $R$ . Note that each discrete point in this set corresponds to an individual transient waveform. We then use `wave_tag` as the index in a pre-characterized lookup table to determine the exact pulse width and height corresponding to each waveform  $k$ . It is important to recognize that a one-to-one monotonic relationship exists between parameter  $b$ , the pulse width  $w$  and the injected charge  $Q$  that generated this waveform so that  $b_{min} \leftrightarrow w_{min} \leftrightarrow Q_{min}$  and  $b_{max} \leftrightarrow w_{max} \leftrightarrow Q_{max}$ . For the given transient pulse we also extract the temporal probabilities  $z_{ns}(k)$  and  $z_{sw}(k)$  from the table. Using these values together with the switching activity  $\alpha$  of the output node, we calculate the scaled strike probability value  $R_{sc}(k)$  as:

$$R_{sc}(k) = R(k) \cdot [(1 - \alpha)z_{ns}(k) + \alpha z_{sw}(k)] \quad (\text{EQ } 3)$$

$R_{sc}(k)$  represents the weighted summation of the temporal probabilities for the switching and non-switching input cases. We perform this computation for each pulse in the descriptor and convert the  $(b, R(b))$  vector into the  $(Q, R_{sc}(Q))$  vector.

From this analysis we observe that for all charge values  $Q$  such that  $Q_{min} \leq Q \leq Q_{max}$  (and correspondingly,  $w_{min} \leq w \leq w_{max}$ ), a soft error will occur in the logic circuit with a probability value indicated by  $R_{sc}$ . The error rate value corresponding to the cumulative effect of all pulses in this SET descriptor  $d$  is determined by calculating the area under this strike probability curve.

$$SER(d) = \int_{Q_{min}}^{\infty} R_{sc}(Q) dQ \quad (\text{EQ } 4)$$

For charge value  $Q > Q_{max}$  (and pulse widths  $w > w_{max}$ ), the strike probability value for the wave  $R(k)$  itself is set to be 0 so that the contribution of pulse widths outside the  $[Q_{min}, Q_{max}]$  (and  $[w_{min}, w_{max}]$ ) range to  $SER(d)$  is zero. Since we use discrete vectors to describe  $R_{sc}$ , we perform numerical integration to calculate  $SER(d)$ . The total circuit SER is the aggregate of the SER due to each individual descriptor at each output node in the circuit.

$$SER_{total} = \sum_{\forall \text{output}} \sum_{\forall \text{descriptor}} SER(d) \quad (\text{EQ } 5)$$

## 2.3 SER vs. Aperture Window

To study 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 0.13 $\mu\text{m}$  technology (Figure 3). We set the clock period for the flipflop  $T_C$  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 [17]. The algorithm produces four SET descriptors at the output node: one pair corresponding to the strikes at I1/I3 and one pair corresponding to the strikes at I2/I4. For the preliminary analysis presented in this section, we assume that the switching activity factor  $\alpha$  is set to 0 so that the entire contribution to SER is due to the case of non-switching inputs. The rate distribution plots of  $R_{sc}$  and pulse widths is also shown in Figure 3. (Recall from the previous sub-section, the one-to-one relationship between various parameters so that  $b \leftrightarrow w \leftrightarrow Q$ ). Note that the  $R_{sc}$  values for I2/I4 are significantly smaller (by about 100X) than the  $R_{sc}$  values for I1/I3. For this set of four descriptors, we observe that the pulse widths are in the range [97ps, 183ps].

From EQ2, 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  $z_{ns}$  associated with those pulses will become zero. Note that for a library flipflop,  $(T_{setup} + T_{hold})$  is significantly smaller than the pulse widths corresponding to the SETs so that no filtering is possible. Consequently, we integrate over the entire range of widths to determine the circuit SER.

By widening the aperture window, we effectively reduce the total range of pulse widths that can pass through the flipflops 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 poten-

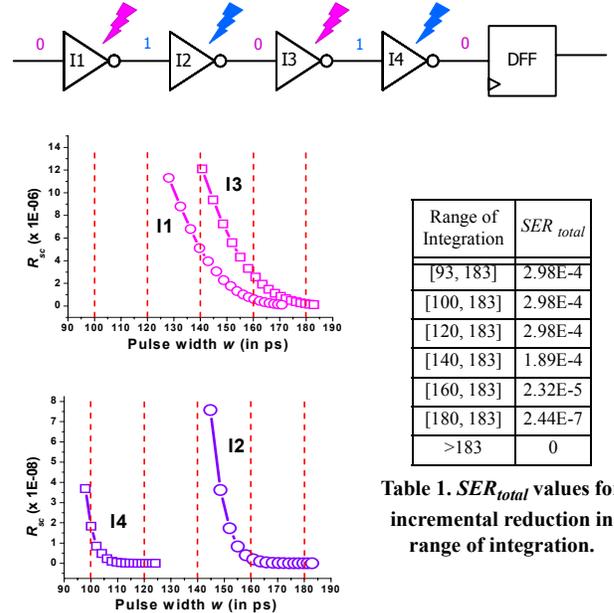


Figure 3. Schematic of four inverter chain and the corresponding rate distribution plots. Note that (I1, I3) are 100X larger than (I2, I4).

tially 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 exactly 140ps so that all pulses of width  $T_{pw} \leq 140ps$  are guaranteed to be temporally masked with the flipflop performing a low-pass filtering operation. In this case, the numerical integration for SER calculation for all the descriptors is performed by setting the  $w_{min}$  to be 140ps since the temporal probabilities  $z_{ns}$  associated with all pulses lower than value will, by definition, be zero. Moreover, since, for a given descriptor, narrow pulses have significantly greater  $R_{sc}$  values compared to wider pulses, large SER reductions can be achieved by gradually shifting the  $w_{min}$  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 11/13 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.

The setup time of a sequential element can be increased in a number of ways. The most direct method is by the addition of extra transistors inside the memory element such that the input stage is slowed in order to filter fast transient pulses. However, such a scheme 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 flipflops more vulnerable to SETs. In [18], the authors proposed the addition of extra resistors across different pairs of nodes in the flipflop in order to reduce 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 (E.g. The method in [18] 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 increase the setup time while constructing the flipflops is to use transistor sizing as a design variable. In the next section, we observe that this method of flipflop sizing is a simple, yet highly effective, method to increase the aperture window to achieve excellent SER immunity. The objective of our sizing scheme is to design a variety of flipflop variants with different sizes such that they filter a large fraction of the transient waveforms.

### 3 Flipflop Sizing Strategies

The positive edge triggered flipflop is the memory element used in a large majority of modern digital circuit designs. A standard D-Flipflop 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.

Transistor sizing can be used in two different ways to increase the aperture window of a flipflop. 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 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 flipflop 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 flipflop. (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 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 flipflop exhibits nearly identical behavior to both rising and falling transitions in terms of the filtering response.

It is important to recognize that the filtering mechanism described here is only applicable to the case of non-switching inputs (Section 2.1). For the case of switching inputs, irrespective of the length of the aperture window, a transient pulse can cause an error if it is close to the setup time of the flipflop and disallows the storage node from charging (or discharging) to the correct value. A sizing solution to combat the effects of transients for switching inputs is rendered difficult by the fact that sizing for rising pulses is antagonistic (in terms of SET filtering) towards falling pulses. We determined through circuit simulations that by carefully sizing the input tristate Device 1 it is possible to reduce the temporal probability  $z_{sw}$  (EQ3) and, hence, improve the filtering response of the flipflop to switching inputs. Since the switching activity  $\alpha$  associated with output nodes is typically a small number such as 0.1, the contribution of this component to the overall SER rate (in EQ3) is also small.

For the industrial 0.13 $\mu$ m cell library that we used for the logic block SER analysis, we determined through circuit simulations that all the transient pulses have widths in the range [78ps, 206ps]. Since

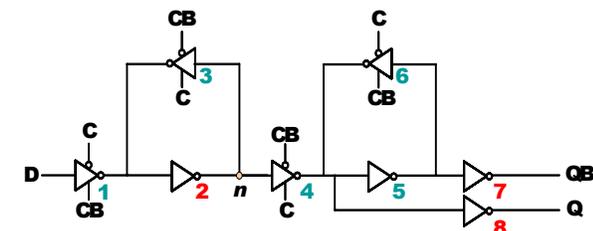


Figure 4. Circuit schematic of a standard D Flipflop. D=Data, C=Clock, CB=Clock-bar, Q=output QB=Q-bar. The sizes of devices 2, 7, 8 are altered to construct the flipflop variants.

Table 1. Delay/power overheads for the flipflop variants.

Flipflop Variant	Overhead		
	$\Delta Del$ (ps)	$\Delta Del$ (xFO4)	$\Delta Pow$ (%)
Base	0	0	0
F100	62.4	1.6	2.9
F130	92.4	2.3	3.4
F160	122.5	3.1	3.8
F190	148.6	3.7	4.3
F210	153.4	3.8	4.6
Fswi	191.1	4.8	7.0

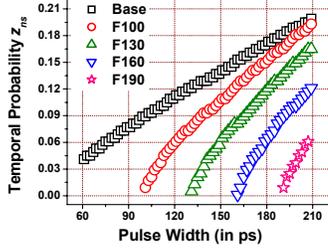


Figure 6. Temporal probability values for the different flipflops

we need to eliminate transient pulses with widths in this range we construct five different variants of flipflops with the values of the aperture window in this range as shown in Table 1. We denote the library flipflop as “Base”. An F130 flipflop has  $(T_{setup}+T_{hold})=130$ ps and filters all transient pulses (for the case of non-switching inputs) with width  $T_{pw} \leq 130$ ps. The F210 flipflop can potentially eliminate any possible transient pulse from latching into the flipflop since the maximum transient pulse width is given as 206ps. In addition, we marginally modify the F210 and construct the Fswi variant to handle the case of switching inputs. We observed that the maximum improvement in  $T_{CQ}$  that can be achieved by sizing up drivers 7 and 8 was fixed ( $\approx 50$ ps). The delay overhead is then the difference in the sum  $(T_{setup}+T_{CQ})$  between the original flipflop and modified sized-up design. In Table 1 we list the overheads associated with each flipflop 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 flipflops can be identified by observing the circuit response associated with them. In Figure 5, we plot the noise rejection curves corresponding to the six flipflop variants. First, we confirm that at full  $V_{dd}$  (1.2V), the aforementioned filtering operation eliminates pulse widths below the associated flipflop 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 flipflop. Figure 5 shows that at pulse height = 1.0V, an F100 flipflop 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 flipflops prove to be even more effective in handling these types of SETs.

We also examine the differences in temporal probability values for the newly constructed flipflops. Figure 6 plots  $z_{ns}$  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  $z_{ns}$  value associated with a flipflop below its filtering threshold value is zero. Consequently, we exclude F210 and Fswi from this plot since the  $z_{ns}$  values associated with them are zero. Next, we observe that for a

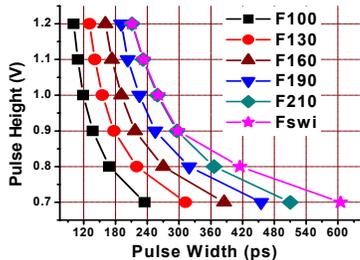


Figure 5. Noise rejection curves for the flipflop variants.

given width above the filtering threshold associated with each flipflop, the  $z_{ns}$  value of the modified flipflop is always lower compared to that of the base case library element. For instance, at pulse width = 120ps,  $z_{ns}$ (F100) is about half of  $z_{ns}$ (base). The increased sizes in the flipflops shrink the interval  $I_k$  of possible time instances where the faulty bit is latched into the input (Section 2.1). Thus, we observe that in addition to the low-pass filtering mechanism, the upsized flipflops lessen the temporal probabilities appreciably, thereby producing a considerable reduction in the total SER of the circuit.

## 4 Results

The proposed flipflop sizing scheme was implemented based on an industrial 0.13 $\mu$ m cell library. We considered the standard D-flipflop from this library as the base case and built upon this initial design to construct the flipflop variants. Based on industrial estimates [19] of the switching activity of local signal nets, we set the value of  $\alpha$  in EQ3 to be 0.1. We used our SER estimation tool proposed in [14] to determine the strike probabilities at the output nodes of each logic block.

We first compare the total SER value while using the different types of flipflops in Table 2. In our analysis, we use benchmark circuits from the ISCAS set [20] and MCNC suite [21]. For the sake of simplicity, we assume that the slacks at all output nodes are identical. Thus, a fixed amount of delay overhead is introduced when we use any given flipflop variant in place of the library flipflop. For each circuit listed here, we apply 1000 random input vectors and calculate the value of  $SER_{total}$  associated with the circuit. We normalize the SER value for the base case for each circuit to be 1.000 and cal-

Table 2. Comparison of SER across flipflop variants

Circuit	Gates	O/P		Base	F100	F130	F160	F210	Fswi
i1	101	12	SER	1000	0.497	0.085	0.022	0.022	0.003
			Pow	100	<101	<101	<101	<101	<101
i2	222	1	SER	1000	0.487	0.060	0.008	0.007	0.000
			Pow	100	<101	<101	<101	<101	<101
i3	258	6	SER	1000	0.558	0.120	0.045	0.045	0.005
			Pow	100	<101	<101	<101	<101	<101
i4	236	6	SER	1000	0.569	0.169	0.049	0.046	0.005
			Pow	100	<101	<101	<101	<101	<101
i5	670	66	SER	1000	0.494	0.093	0.024	0.023	0.001
			Pow	100	<101	<101	101	102	103
i6	875	67	SER	1000	0.524	0.107	0.036	0.035	0.002
			Pow	100	<101	<101	101	102	103
i7	1128	67	SER	1000	0.536	0.116	0.044	0.042	0.004
			Pow	100	<101	101	101	102	103
i8	1822	81	SER	1000	0.536	0.106	0.044	0.043	0.004
			Pow	100	101	102	103	103	104
i9	1086	63	SER	1000	0.531	0.107	0.043	0.041	0.003
			Pow	100	<101	101	101	102	103
i10	3994	224	SER	1000	0.520	0.102	0.043	0.041	0.003
			Pow	100	<101	101	101	102	103
c432	365	7	SER	1000	0.485	0.100	0.019	0.016	0.001
			Pow	100	<101	<101	<101	<101	<101
c499	552	32	SER	1000	0.496	0.086	0.043	0.043	0.003
			Pow	100	<101	<101	<101	101	102
c880	740	26	SER	1000	0.557	0.150	0.040	0.036	0.004
			Pow	100	<101	<101	<101	<101	<101
c1355	857	32	SER	1000	0.493	0.085	0.044	0.044	0.003
			Pow	100	<101	<101	<101	101	101
c1908	959	25	SER	1000	0.513	0.105	0.041	0.040	0.004
			Pow	100	<101	<101	<101	<101	101
c3540	2161	22	SER	1000	0.530	0.124	0.037	0.034	0.004
			Pow	100	<101	<101	<101	<101	<101
c6288	5970	32	SER	1000	0.553	0.158	0.045	0.038	0.006
			Pow	100	<101	<101	<101	<101	<101
Avg			SER	1000	0.526	0.114	0.035	0.034	0.003

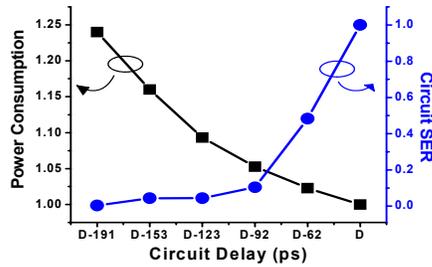


Figure 7. Tradeoff curves for circuit delay/power/SER values.

culate the average relative SER values in the presence of the other flipflops accordingly.

On average we obtain a 9X reduction in SER while using the F130 flipflop and a 300X reduction while using the Fswi variant (maximum improvement is 1000X). Unlike the case presented in Section 2.3, the SER value does not reduce to 0 for the case of F210 because the temporal probability for the case of switching inputs is never equal to 0. The power overhead incurred is less than 1% in most cases since the flipflops contribute only a small amount to the overall power consumption of the entire circuit. The maximum power overhead occurs for a medium-sized circuit (i8) with a large number of output nodes. The delay overheads for these circuits are identical to the delay values presented in Table 1. Note that in the worst case the delay overhead is about 5 FO4, which is typically small compared to the worst-case path delay in circuits other than high-performance microprocessors.

In the previous table we maintain a fixed logic implementation and use different sized flipflops to measure the amount of SER reduction. In the next set of experiments, we explore the use of different logic implementations (that consume larger power) to tradeoff SER reduction with performance. Consider the tradeoff curves shown in Figure 7 for benchmark circuit *i6*. We first set the initial specified delay point to D and normalize the SER and power consumption corresponding to this circuit implementation to 1.000. Moving along the delay axis, we resynthesize the same circuit for the new delay values of {(D-62)ps, (D-92)ps, ...}. The tighter delay constraint values are chosen such that the differences are exactly equal to the delay overheads exhibited by the flipflop variants in Table 1. In other words, given a circuit with the operating delay point set at (D-62)ps, the inclusion of the F100 flipflops at the outputs will introduce a fixed amount of delay overhead (of exactly +62ps) resulting in the modified circuit operating at the initial delay value D. In a similar fashion, we introduce the F130 flipflop for the circuit synthesized to operate at (D-92)ps and so on for the five delay points depicted in the plot. From this discussion, we observe that all the circuits in this figure have the same total delay value of D.

The tightened delay constraints require the use of larger devices resulting in larger power consumption (left vertical axis). However, as we observe on the right vertical axis, the SER value of the circuit decreases in an exponential manner due to the use of the proposed flipflop variants. From this plot, we observe that an ideal tradeoff is achieved for the (D-92)ps delay point where the power overhead is 5.3%, while the total circuit SER has reduced by about 9X.

## 5 Conclusions

In this paper, we presented a flipflop sizing scheme to combat the effects of single event transients on combinational logic circuits. Our method uses the concept of increasing the aperture windows to

enhance the effects of temporal masking and reduce the overall SER of the circuit. We constructed a library of flipflop variants that provide significant reductions in circuit SER (up to 1000X) while imposing small delay and power overheads. Experimental results in exploring the tradeoffs in circuit delay/power and SER show that for a 9X reduction in SER, the power overhead is about 5%, while operating at a fixed delay point.

## 6 Acknowledgments

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

## 7 References

- [1] R. Baumann, "Soft errors in advanced computer systems," *IEEE Design and Test of Computers*, 22 (3), May 2005.
- [2] P. Shivakumar, M. Kistler, S. Keckler, D. Burger, L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," *DSN 2002*.
- [3] S. Mitra, T. Karnik, N. Seifert, M. Zhang, "Logic soft errors in sub-65nm technologies design and CAD challenges," *DAC 2005*.
- [4] R. Ramanarayanan, V. Degalhal, N. Vijaykrishnan, M. Irwin, D. Duarte, "Analysis of soft error rate in flipflops and scannable latches," *Intl. SoC Conference*, 2003.
- [5] F. Faccio, K. Kloukinas, A. Marchioro, T. Calin, J. Cosculluela, M. Nicolaidis, R. Velazco, "Single event effects in static and dynamic registers in a 0.25μm technology," *IEEE Trans. on Nuclear Science*, 46(6), 1999.
- [6] T. Karnik, S. Vangal, V. Veeramachaneni, P. Hazucha, V. Erraguntla, S. Borkar "Selective node engineering for chip-level soft error rate improvement," *VLSI Symposium*, 2002.
- [7] T. Monnier, F. Roche, G. Cathebras, "Flipflop hardening for space applications," *Intl. Workshop on Memory Technology*, 1998.
- [8] N. Seifert, N. Tam, "Timing vulnerability factors of sequentials," *IEEE Trans. on Device and Materials Reliability*, 4(3), 2004.
- [9] M. Zhang, N. Shanbhag, "A CMOS design style for logic circuit hardening," *IRPS*, 2005.
- [10] M. Zhang, N. Shanbhag, "An energy-efficient circuit technique for single event transient noise-tolerance," *ISCAS*, 2005.
- [11] S. Mitra, N. Seifert, M. Zhang, Q. Shi, K. Kim, "Robust system design with built-in soft-error resilience," *IEEE Computer*, 38(2), 2005.
- [12] D. Mavis, P. Eaton, "Soft error rate mitigation techniques for modern microcircuits," *IRPS*, 2002.
- [13] N. Weste, D. Harris, "CMOS VLSI Design: A Circuits and Systems Perspective," Addison Wesley, 2005.
- [14] R. R. Rao, K. Chopra, D. Blaauw, D. Sylvester, "An efficient static algorithm for computing the soft error rates of combinational circuits," *DATE 2006*.
- [15] L. Freeman, "Critical charge calculations for a bipolar SRAM array," *IBM Journal of R & D*, 40(1), 1996.
- [16] Q. Zhou, K. Mohanram, "Cost effective radiation hardening technique for combinational logic," *ICCAD 2004*.
- [17] M. Zhang, N. Shanbhag, "A soft error rate (SERA) analysis methodology," *ICCAD 2004*.
- [18] H. Cha, J. Patel, "Latch design for transient pulse tolerance," *ICCD*, 1994.
- [19] N. Magen, A. Kolodny, U. Weiser, N. Shamir, "Interconnect power dissipation in a microprocessor," *SLIP 2004*.
- [20] F. Brglez, H. Fujiwara, "A neural netlist of ten combinational benchmark circuits and translator in Fortran," *ISCAS 1985*.
- [21] S. Yang, Logic synthesis and optimization benchmarks user guide, Microelectronics Research Center of North Carolina, 1991.