Abstract—Vulnerability of combinational logic to soft errors exponentially increases with technology scaling. Reducing soft error susceptibility of logic gates comes with extra area, delay, and power consumption overhead that needs to be balanced in the entire circuit. In this paper, we present efficient soft error hardening techniques based on selective gate resizing to maximize soft error suppression for the entire logic-level design while minimizing area and delay penalties. Experimental results confirm that these techniques are able to highly reduce soft error rate with modest area and delay overhead.

I. INTRODUCTION

Soft errors, also called Single Event Upsets (SEU), are intermittent malfunctions of the hardware that are not reproducible [5]. These errors, which can occur more often than hard (permanent) errors [3], arise from energetic particles, namely neutrons and alpha particles. Soft Error Rate (SER) for a device is defined as the error rate due to SEUs.

Device scaling significantly affects the susceptibility of integrated circuits to soft errors [6]. As the feature size shrinks, the amount of charge per device decreases, enabling a particle strike to be much more likely to cause an error. As a result, particles of lower energy, which are far more plentiful, can generate sufficient charge to cause a soft error. Hence, the vulnerability of VLSI systems to soft errors exponentially increases as an unwanted side effect of Moore’s law [4]. So far, memory elements have been more susceptible to soft errors than the combinational logic. However, analytical models predict that the soft error rate in the combinational logic will be comparable to that of memory elements by 2011 [6].

The ability of a CMOS logic gate to tolerate SEUs is a function of the injected charge by the hitting particle and transistor sizing [7]. By increasing the size of transistors, it is possible to absorb the transient pulse caused by the injected charge, preventing its propagation to the next logic stage. Therefore, transistor up-sizing is a possible solution to reduce soft error susceptibility of logic gates. Nevertheless, this approach imposes area and delay penalties [8]. Therefore, a practical approach is selective gate sizing based on SER reduction requirements as well as cost (area, delay, and power consumption) budgets.

In this paper, We present efficient gate sizing techniques for SER reduction in which the contribution of each logic gate to the system-level SER and its criticality in the in the overall performance are carefully considered. Fast and efficient soft error vulnerability minimization algorithms under different constraints, such as area, delay, or both, are also presented.

The rest of this paper is organized as follows. In Sec. II, the proposed soft error hardening technique are presented. In Sec. III, the experimental results are presented. Finally, Sec. IV concludes the paper.

II. SOFT ERROR HARDENING

It has been demonstrated that the size of a gate driving a node and the amount of capacitance at the node determine the magnitude and duration of the SEU transient [7]. A large transistor can dissipate the charge injected by particle hit so that the effect cannot be propagated to the gate output. As transistor aspect ratio \( \frac{W}{L} \) increases, the transient pulse width due to SEU decreases in the same proportion. As the result, the optimal transistor sizing is a linear function of the injected charge. Based on the energy level of particle and device characteristics, the amount of injected charge can be calculated. Using this information, the size of the logic gate for completely absorbing transient pulse will be determined. However, increasing the size of a gate increases its area and power consumption accordingly. Moreover, the delay of the driving gate (previous stage) will be also increased. Therefore, it is not practical to resize (up-size) all logic gates in the circuit for SER reduction.

Error Propagation Probabilities (EPP) of the gates can be used as a metric to measure the contribution of each gate to the overall soft error rate. A higher EPP means that a bit-flip at the output of the gate will more likely cause an error at the circuit primary outputs. The technique presented in [8] exploits the information regarding EPP of the gates in the circuit to use transistor sizing for gates with highest EPPs. Although this approach can reduce the area overhead associated with gate sizing, it does not particularly address delay penalties since it does not pay attention to the timing slack of the gates chosen for resizing. Note that if a gate in the critical path (i.e. with zero slack) is resized, the overall delay of the circuit will be also increased. However, if a non-critical gate is resized so that its slack remains non-negative after resizing, the overall circuit delay is not affected. Consequently, for cost-effective SER reduction, timing and layout information as well as EPPs have to be considered.

A. Motivation

Here we present some data obtained from timing analysis as well soft error rate estimation of ISCAS’85 circuits regarding timing slack and error propagation probability of individual
logic gates [1]. Figure 1 shows the ratio of non-critical gates in ISCAS’85 benchmark circuits. On average, 75% of gates are non-critical (positive timing slack). This means that resizing of these gates will not impose any performance penalty as long as the slacks of the resized gates are still non-negative.

Figure 2 shows the distribution of error propagation probabilities of the logic gates for some ISCAS’85 benchmark circuits. These values are normalized to the number of gates in each circuit. For instance, in average, 63% of gates in these circuits have EPP of 0.4 or less. In other words, a small subset of gates contributes to the majority of system-level SER. Figure 3 shows the product of average EPP and the number of critical and non-critical gates for ISCAS’85 circuits. It can be seen that the share of non-critical gates in the overall SER is twice more than that for critical gates. This ratio is even bigger in larger circuits of this benchmark set. This shows an opportunity to hide the delay overhead associated with SER reduction. For example, we can expect that a slack-aware approach for gate resizing can reduce the SER to one third (3x reduction) with no performance overhead.

Figure 4 shows the gates whose timing slacks are affected due to resizing one particular gate (changing its input capacitance and its propagation delay). The slack-aware resizing algorithm is outlined below.

1) we first sort the gates based on their EPPs in the descending order.
2) At each step, we choose a gate from the top of this list as a candidate for resizing.
3) If the slack of this gate is large enough such that after resizing its slack remains non-negative, this gate is resized.
   - The delay of this resized gate along with those gates driving and being driven by this are recomputed.
   - The slacks of all gates in the forward and the backward logic cones originated from the resized gate are recalculated and updated (Fig. 4).
4) Otherwise, this candidate gate is discarded and the next gate in the list is examined (goto step 2).
5) this process in repeated until the list becomes empty.

Figure 2 shows the distribution of error propagation probabilities of the logic gates for some ISCAS’85 benchmark circuits. These values are normalized to the number of gates in each circuit. For instance, in average, 63% of gates in these circuits have EPP of 0.4 or less. In other words, a small subset of gates contributes to the majority of system-level SER. Figure 3 shows the product of average EPP and the number of critical and non-critical gates for ISCAS’85 circuits. It can be seen that the share of non-critical gates in the overall SER is twice more than that for critical gates. This ratio is even bigger in larger circuits of this benchmark set. This shows an opportunity to hide the delay overhead associated with SER reduction. For example, we can expect that a slack-aware approach for gate resizing can reduce the SER to one third (3x reduction) with no performance overhead.

**B. Timing-aware SER Reduction**

In our approach, we consider both EPP and timing slack of logic gates in selecting them for resizing. Note that changing the size of one gate affects not only its timing slack, but also the slack of other logic gates. Figure 4, as an example, shows the gates whose timing slacks are affected due to resizing one particular gate (changing its input capacitance and its propagation delay). The slack-aware resizing algorithm is outlined below.

1) we first sort the gates based on their EPPs in the descending order.
2) At each step, we choose a gate from the top of this list as a candidate for resizing.
3) If the slack of this gate is large enough such that after resizing its slack remains non-negative, this gate is resized.
   - The delay of this resized gate along with those gates driving and being driven by this are recomputed.
   - The slacks of all gates in the forward and the backward logic cones originated from the resized gate are recalculated and updated (Fig. 4).
4) Otherwise, this candidate gate is discarded and the next gate in the list is examined (goto step 2).
5) this process in repeated until the list becomes empty.

The above procedure minimizes the SER without introducing any performance penalty. In order to minimize SER under performance constraints, extra timing budget (user-defined) can be also considered in the above procedure. This extra budget can be expressed as a percentage of the original delay of the circuit. To incorporate this, an initial step is required to be added to the above procedure in which the slack of all gates are adjusted based on the additional delay. The remainder of the algorithm remains the same.

We use the example shown in Fig. 5 to show how this procedure is applied. In this figure, G4, G5, G6, G7, and G9 are in the critical path. We first sort the non-critical gates according to their EPPs. \((G_i(x))\) means that the EPP of \(G_i\) is equal to \(x\.)

**Non-critical gate list= \{G3(1), G8(1), G12(1), G11(0.96), G14(0.76), G2(0.53), G10(0.51), G13(0.37), G1(0.36)\}.**

Let’s assume that the particle energy is 0.15 pC. The resizing process is done in such a way that it immunes the gates against SEUs with this energy. To do so, we need to resize the transistors to \(\frac{W}{L} = 4\), i.e. 4x sizing [7]. We start from the gates with the highest EPP. G3 has a considerable slack of 29.5 ps. After resizing this gate, we recompute the slacks of the forward and backward logic cones of this gate. The next candidate in the non-critical list is G8 and its slack
is 18 ps. Although this gate is not on the critical path, once it is resized, the critical path delay will be affected. This is because if G8 is resized, the load capacitance of G6, which is on the critical path, will be affected. So, we discard this gate.\footnote{This gate could be resized if 5% extra delay overhead were considered.}

We can successfully resize the next candidates (G12, G11, G14, G10, G2, G13, and G1) using the available slack budget. In this example, we assumed that the primary inputs provide the required current to the next gate levels. If we buffer the primary inputs (using a BUF gate), we will no longer be able to resize the last three gates of the non-critical list (G10, G13, and G1). In this example, the SER of this circuit is reduced by almost 60% while the circuit performance has not been affected.

![Fig. 5. An example of a circuit](image)

### C. General Optimization Problems for SER Reduction

The problem of maximizing SER reduction with minimum area and delay overhead, as an optimization problem, is addressed here. The heuristic algorithm presented in Sec. II-B tries to maximize SER reduction with bounded delay overhead. Due to the heuristic nature of the presented algorithm, achieving the “maximum” SER reduction is not always guaranteed.

In this section, we look at other variations of this problem and investigate possible heuristic solutions.

1) **Maximizing SER Reduction Under Area Constraints:**

In order to maximize SER reduction with minimum area overhead, both EPP (as a metric for the benefit in SER reduction) and the area (as a metric for cost) of each gate must be taken into account. This optimization problem can be converted into the classical knapsack problem \[2\]. In knapsack problem, there are \( n \) items, each item \( i \) has value \( v_i \) and weight \( w_i \). The objective is to select a subset of these items such that the sum of weights of selected items is not greater than \( W \) (the size of knapsack) while the sum of their values is maximized.

This optimization problem is proven to be NP-hard \[2\]. However, a heuristic solution to this problem is to sort items based on \( \frac{v_i}{w_i} \) (value per unit weight) in the descending order. Then, the items will be chosen from this list as long as the sum of weights of selected items does not exceed \( W \). The time complexity of this heuristic is \( O(n \log n) \) since the sorting part is \( O(n \log n) \) and the selection part is \( O(n) \).

We can use the same heuristic for this SER reduction problem by sorting the logic gates based on \( \frac{EPP_i}{\Delta area_i} \) in the descending order, where \( EPP_i \) and \( \Delta area_i \) are the EPP and area increase (due to sizing) of gate \( g_i \), respectively. Given a user-specified area budget \( A \), gates in this list are resized until the area increase exceeds the area budget \((\sum \Delta area_i > A)\). Similar approach can be used for power-constrained SER minimization in which power increase is used instead of area increase.

2) **Maximizing SER Reduction Under Delay Constraints:**

This problem is basically the original problem addressed in Sec. II-B. Note that unlike the previous problem (Sec. II-C.1), this optimization problem cannot be directly converted to the knapsack problem. This is because resizing one gate (changing its delay) affects the timing of other gates, as well. This is why our presented solution in Sec. II-B is not the same as the straightforward heuristic for the knapsack problem.

3) **Maximizing SER Reduction Under Area and Delay Constraints:**

Given user-specified area and delay budgets, the objective here is to maximize SER reduction such that delay and area overhead does not exceed the specified budget. Since this problem is a combination of the two previous problems, we can use a heuristic based on the algorithms presented in Sec. II-B and Sec. II-C.1. Specifically, we sort logic gates based on \( \frac{EPP_i}{\Delta area_i} \) in the descending order. While choosing each gate from the top of this list for resizing, we consider the timing slack of the circuit after resizing that particular gate, and compare it with the timing budget. If resizing this gate results in violation of the timing budget, this gate is discarded (not resized) and the next gate in the list is considered.

### III. Experimental Results

We have implemented these heuristic slack-aware resizing algorithms and applied them to ISCAS'89 sequential circuits which have larger combinational logic cores compared to pure combinational ISCAS'85 circuits. Figure 6, Figure 7 and Figure 8 show the SER reduction achieved by delay-constrained, area-constrained, and area/delay constrained sizing techniques, respectively. We have considered different values of delay and area budget as the user-defined constraints. The gates chosen for sizing are resized four times (4x) their original sizes to completely block the propagation of the injected charge. As can be seen in these figures, the SER of the circuit can be greatly reduced with modest area and/or delay penalties.

### IV. Conclusions

Soft errors are the main reliability threat of digital systems. In particular, vulnerability of digital systems grows in direct proportion to the Moore’s law. It is predicted that combinational logic will become equality susceptible to soft errors compared to sequential elements in just few years.

In this paper, a gate-level soft error rate reduction technique with bounded area and delay penalties has been presented. We have developed gate resizing algorithms for the entire circuit in which the error propagation probability, area overhead, and timing slack of each gate are carefully considered. Different versions of the SER minimization problem under different area and delay constraints have been addressed. Our results show that more than 6x reduction in the overall SER can be achieved without any performance penalty.
Fig. 6. Soft error vulnerability reduction using slack-aware resizing technique with bounded delay overhead.

Fig. 7. Soft error vulnerability reduction using slack-aware resizing technique with bounded area overhead.

Fig. 8. Soft error vulnerability reduction using slack-aware resizing technique with bounded area/delay overhead.

REFERENCES