Soft Error Rate Reduction Using Redundancy Addition and Removal

Kai-Chiang Wu and Diana Marculescu
Department of Electrical and Computer Engineering
Carnegie Mellon University
{kaichiaw, dianam}@ece.cmu.edu

Abstract – Due to current technology scaling trends such as shrinking feature sizes and reducing supply voltages, circuit reliability has become more susceptible to radiation-induced transient faults (soft errors). Soft errors, which have been a great concern in memories, are now a main factor in reliability degradation of logic circuits. In this paper, we propose a novel framework based on redundancy addition and removal (RAR) for soft error rate (SER) reduction. Several metrics and constraints are introduced to guide our proposed framework towards SER reduction in an efficient manner. Experimental results show that up to 70% reduction in output failure probability can be achieved with relatively low area overhead.

1. Introduction

Circuit reliability has become a critical issue in the deep submicron design era. Crosstalk, voltage drop, and radiation-induced transient errors are currently some of the main factors in reliability degradation. Due to current technology scaling trends, digital designs are becoming more susceptible to radiation-induced particle hits resulting from radioactive decay and cosmic rays, than all other factors [1]. A low-energy particle that before had no effect on a circuit can now flip the output of a gate. Such a bit-flip of a gate is called a *single-event transient* (SET) or a glitch. A *single-event upset* (SEU) or a soft error occurs if the SET is propagated to an output and latched into a memory element. The rate at which soft errors occur is referred to as *soft error rate* (SER).

Both static and dynamic memories have suffered from soft errors because of their dense and vulnerable structures. Unlike SETs in logic circuits, which need to be propagated to outputs before being captured, soft errors happen in memories as long as SETs strike. On the other hand, during the process of propagation in logic circuits, a SET could be masked by three mechanisms: (i) *logical masking* – a SET which is not on a sensitized path from the location where it originates is logically masked; (ii) *electrical masking* – a SET which is attenuated and becomes too small in amplitude or duration to be latched is electrically masked; (iii) *latching-window masking* – a SET which does not arrive "on time" is also masked, depending on the setup and hold times of the target memory element.

These three mechanisms prevent some SETs from being latched and alleviate the effects of soft errors in digital systems. However, continuous scaling trends have negative impact on these masking mechanisms. Decreasing logic depth reduces the probability of logical masking since the path from where a SET originates to a latch is more easily sensitized. The pulse attenuation because of electrical masking, determined by gate delay, also decreases due to smaller contribution of gate delay compared to wire delay. Increasing clock frequencies shorten latching-window intervals and facilitate SET latching. As a result, soft errors in logic become as great of a concern as in memories, where soft errors can be mitigated by conventional error detecting and correcting codes. A recent study [2] has shown that soft errors would significantly degrade the robustness of logic circuits in the 90nm process technology or below. In addition, the SER of combinational circuits is predicted to exceed that of unprotected memory elements by 2011, while the nominal SER of latches tends to be nearly constant or even slightly decreasing [3]. In Figure 1, we demonstrate the SER impact of internal gates and flip-flops in sequential circuits by analyzing output error susceptibility. Since sequential circuits usually have more internal gates (combinational logic) than flip-flops (sequential logic), the impact attributed to combinational logic is 3X-49X larger than the one attributed to sequential logic, as shown in Figure 1, when normalized with respect to all possible locations of particle hits.

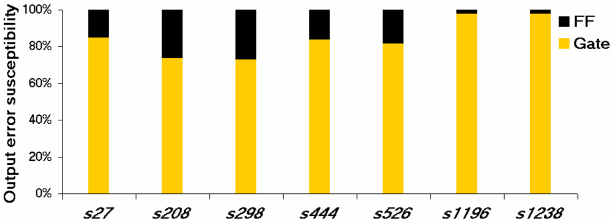


Figure 1. SER impact of internal gates and flip-flops

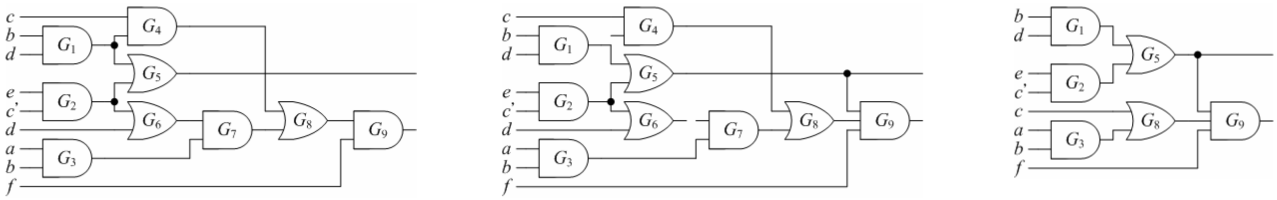
Typically, two types of methods are used for soft error hardening. The first one, *fault avoidance*, consists in minimizing the occurrence of SETs at the most sensitive nodes, which in effect reduces SET generation. The second one, *fault correction*, attempts to maximize the probabilities of the three masking mechanisms, which reduces the likelihood of generated SETs being latched. The first category generally exploits fabrication process (device-level) to reduce charge collection. Therefore, a particle hit harmful for a baseline circuit will cause less or even no damage for the radiation-hardened version. The second category works on circuit-level or higher levels of abstraction to achieve SER improvement. In this paper, we propose an approach which belongs to the second category for combinational logic.

The rest of this paper is organized as follows: Section 2 gives an overview of related work and outlines the contribution of our paper. In Section 3, we introduce the SER analysis engine used in this paper. Section 4 presents our proposed SER reduction framework. In Section 5, several constraints that guide our framework towards SER reduction are discussed. Section 6 describes a few practical considerations for our proposed framework. In Section 7, we report experimental results for a set of benchmarks. Finally, the conclusion is drawn in Section 8.

2. Related Work and Paper Contribution

The most well-known technique for achieving soft error tolerance is *triple modular redundancy* (TMR). TMR consists of three identical copies of an original circuit feeding a majority voter. However, TMR induces more than 200% overhead in terms of area and power. Partial duplication [4] targets only nodes with high soft error susceptibility and ignores nodes with low soft error susceptibility. It also involves at least 50% area penalty according to the specified constraint and additional delay overhead due to the checker circuit. Gate resizing strategy [5] achieves SER reduction by modifying the W/L ratios of transistors in gates. Potentially large overheads in area, delay, and power are introduced in order to obtain a significant improvement in SER. A related method [6] uses optimal assignments of gate sizes,

978-1-4244-1922-7/08/$25.00 ©2008 IEEE



(a) The original circuit (b) The circuit after redundancy addition (c) The circuit after redundancy removal

Figure 2. An example of redundancy addition and removal [12]

supply voltages, threshold voltages, and output capacitive loads to get better results with smaller overheads. Nevertheless, such a method increases design complexity and may make the resulting circuit hard to optimize at physical design stage. Another scheme [7] focuses on flip-flop selection from a given set. This scheme increases the probability of latching-window masking by lengthening latching-window intervals, but does not take into consideration logical masking and electrical masking, which are also dominant factors of circuit SER. A hybrid approach [8] combines gate resizing with flip-flop selection to acquire SER improvement.

Instead of exploring spatial redundancy as described above, several ideas of soft error hardening based on temporal redundancy were presented in [9][10]. However, such techniques which employ time-domain majority voting mechanism are extremely sensitive to delay variation and fail to cope with large duration SETs because sufficient slack time is required.

This paper proposes a SER reduction framework using *redundancy addition and removal* (RAR) [11][12]. RAR has been presented as a successful logic optimization technique which iteratively adds and removes redundant wires to minimize a circuit in terms of literal count. Since during each step of wire addition and removal the soft error rate of a circuit may vary, we rely on estimating the effects of redundancy manipulations and merely accept those with positive impact. The end result of such an approach is a net reduction in soft error rate. The proposed framework has several advantages over other existing techniques:

- First, this RAR-based approach incurs very little or no area overhead since there always exist(s) one or more redundant removable wire(s) after a redundant wire is added into a circuit.
- Second, because of the efficiency in run time and memory usage of the RAR algorithm, our framework can be applied effectively to large circuits.
- Third, our framework relies on a symbolic reliability analyzer MARS-C [13], which provides a unified treatment of three masking mechanisms through decision diagrams. Hence, **all** masking mechanisms rather than one or two of them are considered as criteria for the purpose of SER reduction. Moreover, we introduce a novel metric for masking impact analysis. Using this metric, a systematic algorithm, which can precisely estimate the impact on SER of an added/removed wire and decide whether to accept the given addition/removal step, is developed.
- Finally, such an approach is orthogonal to existing approaches targeting gate resizing and flip-flop selection and thus provides additive savings on top of these SER reduction techniques.

3. SER ANALYSIS ENGINE

Analyzing the soft error rate of a circuit accurately and efficiently is a crucial step for SER reduction. Intensive research has been done so far in the area of SER modeling and analysis. Among various existing modeling frameworks, we choose the symbolic one presented in [13] as the SER analysis engine. We motivate our choice by the fact that, by using this symbolic SER analyzer, we can simultaneously quantify the *error impact* and the *masking impact* of each gate in a combinational circuit. To model a transient glitch originating at gate G to be latched at output F, the following events can be defined:

$\mathcal{A}$: $A > V_{th}$ (if the correct output is "0") or
$A < V_{th}$ (if the correct output is "1")
where A is the amplitude of the glitch and V_{th} is the switching threshold of the latch.

$\mathcal{D}$: $D > t_{setup} + t_{hold}$
where D is the duration of the glitch, and t_{setup} and t_{hold} are the setup and hold times of the latch.

$\mathcal{T}$: $t \in [T_{clk} + t_{hold} - T - D, T_{clk} - t_{setup} - T]$
where t is the time when the initial glitch occurs, T is the delay from gate G to output F, and T_{clk} is the clock period.

In this model, logical and electrical masking are implicitly included in $\mathcal{A}$ and $\mathcal{D}$, while latching-window masking is included in $\mathcal{T}$. The three events are necessary conditions for a soft error to happen. In addition, $\mathcal{D}$ is satisfied only if $\mathcal{A}$ is satisfied (i.e., $\mathcal{D} \subset \mathcal{A}$). Under the assumption that t is uniformly distributed [13], the probability that a soft error occurs can be expressed as:

$$\mathrm{P}(\mathcal{A} \cap \mathcal{D} \cap \mathcal{T}) = \mathrm{P}(\mathcal{D} \cap \mathcal{T}) = \mathrm{P}(\mathcal{T} | \mathcal{D})\cdot\mathrm{P}(\mathcal{D}) \qquad \textbf{(1)}$$

$$= \sum_k \Big(\mathrm{P}(t \in [T_{clk} + t_{hold} - T - D, T_{clk} - t_{setup} - T] \,|\, D = D_k) \cdot \mathrm{P}(D = D_k) \Big)$$

$$= \sum_k \left(\frac{D_k - (t_{setup} + t_{hold})}{T_{clk} - d_{init}} \cdot \mathrm{P}(D = D_k) \right)$$

where $\{D_k\}$ is the set of possible glitch durations and d_{init} is the initial glitch duration.

To find out possible values for duration, $\{D_k\}$, the attenuation model, a function of gate delay, is used. To determine the probability of having a glitch with duration D_k, the authors of [13] use Binary Decision Diagrams (BDDs) and Algebraic Decision Diagrams (ADDs). The detailed methodology of [13] is not described here.

4. RAR-BASED SER REDUCTION FRAMEWORK

Redundancy addition and removal (RAR) is a logic minimization technique which performs a series of wire addition and removal by searching for redundant wires in a circuit. Candidate wires for addition can be identified according to the mandatory assignments made during the process of *automatic test pattern generation* (ATPG). For example, in Figure 2(a) [12], the mandatory assignments for gate G_6 stuck-at-1 fault are $\{f = 1, G_3 = 1, G_4 = 0, G_6 = 0\}$, from which we can get the implications $\{d = 0, G_1 = 0, G_2 = 0, G_5 = 0\}$. If a wire from gate G_5 to gate G_9 is added into the circuit, there will be a conflicting assignment because gate G_5 should be set to be "1" to make gate G_6 stuck-at-1 fault observable at outputs. So the wire $G_5 \rightarrow G_9$ is a candidate for wire addition.

One still needs to check if the candidate wire is indeed redundant. i.e., the wire will not change the functionality of the circuit. In the above example, the wire $G_5 \rightarrow G_9$ is redundant. The newly added wire could cause one or more originally irredundant wires to become redundant (removable). ATPG is again used for redundancy checking of each wire except the one just inserted (e.g., wire $G_5 \rightarrow G_9$ in Figure 2(b)) by finding compatible mandatory assignments. If a set of mandatory assignments for a wire cannot be derived, the wire is said to be redundant and thus can be removed. Consider the same example in Figure 2: after adding wire G_5 → G_9 into the circuit, wires G_1 → G_4 and G_6 → G_7 become redundant as compatible mandatory assignments do not exist for both of them. So they can be removed, as shown in Figure 2(b). Note that gates with only one fanin and gates without fanout can also be deleted. Figure 2(c) shows the resulting circuit after redundancy removal.

For our proposed objective of SER reduction, using redundancy addition and removal in an unsystematic manner may increase SER by reducing the number of gates or the depth of circuits: less gate count will affect the impact of logical masking, while smaller logic depth will reduce the impact of both logical and electrical masking. The basic principle of our proposed approach is to **keep wires/gates with higher masking impact** and to **remove wires/gates with higher error impact**. *Mean masking impact* (MMI) and *mean error impact* (MEI), used as metrics for guiding the RAR-based framework, are defined in the sequel.

4.1. Mean Error Impact

For each internal gate G_i, initial duration d and initial amplitude a, *mean error impact* (MEI) [13] over all primary outputs F_j that are affected by a glitch occurring at the output of gate G_i is defined as:

$$\mathrm{MEI}(G_i^{d,a}) = \frac{\sum_{k=1}^{n_f}\sum_{j=1}^{n_F} \mathrm{P}(F_j\ fails \mid G_i\ fails \cap init\_glitch = (d,a))}{n_F \cdot n_f} \quad \textbf{(2)}$$

where n_F is the cardinality of the set of primary outputs, $\{F_j\}$, and n_f is the cardinality of the set of probability distributions, $\{f_k\}$.

The MEI value of a gate quantifies the probability that at least one primary output is affected by a glitch originating at this gate. **The larger MEI a gate has, the higher the probability that a glitch occurring at this gate will be latched.** This implies that those gates with higher MEI make the circuit more vulnerable to soft errors. Thus, it will be beneficial for SER if **gates with large MEI are removed** from the circuit.

4.2. Mean Masking Impact

D(G_i) is the attenuated duration of a glitch at gate G_i;
C(G_i) is the set of gates in the fanin cone of gate G_i;
F(G_i) is the set of gates in the **immediate** fanin of gate G_i;
p(G_j, G_i) is the set of gates on the paths between gates G_j and G_i.

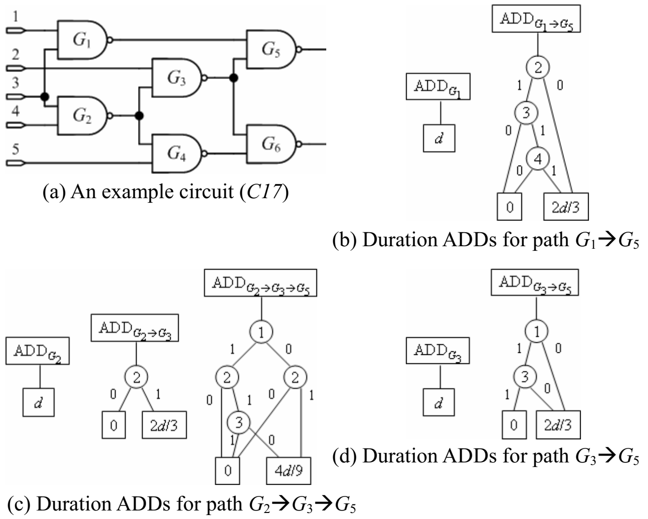
For each internal gate G_i, initial duration d and initial amplitude a, we define *mean masking impact on duration* ($\mathrm{MMI_D}$) as:

$$\mathrm{MMI_D}(G_i^{d,a}) = \frac{\sum_{k=1}^{n_f}\sum_{j=1}^{n_G} \mathrm{MI_D}(G_j^{d,a} \rightarrow G_i)}{n_G \cdot n_f \cdot d} \quad \textbf{(3)}$$

where n_G is the cardinality of C(G_i), n_f is the cardinality of the set of probability distributions, $\{f_k\}$, and $\mathrm{MI_D}(G_j^{d,a} \rightarrow G_i)$, *masking impact on duration* of gate G_i with respect to gate G_j, denotes the absolute duration attenuation contributed by gate G_i on a glitch with duration d and amplitude a originating at gate G_j. More formally, $\mathrm{MI_D}(G_j^{d,a} \rightarrow G_i)$ can be defined as: **(4)**

$$\begin{aligned}&\mathrm{MI_D}(G_j^{d,a} \rightarrow G_i)\\ &= \sum_k \Big(\mathrm{P}(\mathrm{D}(G_i) = D_k \mid G_j\ fails \cap init\_glitch = (d,a)) \cdot (d - D_k)\Big)\\ &- \sum_{G_l \in \mathrm{F}(G_i) \cap \mathrm{p}(G_j, G_i)} \sum_k \Big(\mathrm{P}(\mathrm{D}(G_l) = D_k \mid G_j\ fails \cap init\_glitch = (d,a)) \cdot (d - D_k)\Big)\end{aligned}$$

where $\{D_k\}$ is the set of possible values for glitch duration, the same as that in Equation (1). The second summation represents the total weighted attenuation attributed to gate G_i's immediate fanin gates on the paths between gates G_j and G_i, instead of just gate G_i itself. Intuitively, $\mathrm{MI_D}(G_j^{d,a} \rightarrow G_i)$ quantifies how much attenuation can be contributed to gate G_i **only**, on the duration of glitches originating at gate G_j.


1
2
3
4
5
G_1
G_2
G_3
G_4
G_5
G_6
$\mathrm{ADD}_{G_1 \rightarrow G_5}$
ADD_{G_1}
d
0
$2d/3$
ADD_{G_2}
$\mathrm{ADD}_{G_2 \rightarrow G_3}$
$\mathrm{ADD}_{G_2 \rightarrow G_3 \rightarrow G_5}$
$4d/9$
ADD_{G_3}
$\mathrm{ADD}_{G_3 \rightarrow G_5}$

(a) An example circuit (*C17*)

(b) Duration ADDs for path G_1→G_5

(c) Duration ADDs for path G_2→G_3→G_5

(d) Duration ADDs for path G_3→G_5

Figure 3. An example circuit (*C17*) and its duration ADDs associated with mean masking impact on duration of gate G_5

Example: In Figure 3(a), assume only one set of input probability distribution $\{P_1 = 0.5, P_2 = 0.5, P_3 = 0.5, P_4 = 0.5, P_5 = 0.5\}$, where P_i is the probability of "1" for the i^{th} primary input, is applied to the example circuit. The duration ADDs associated with mean masking impact on duration of gate G_5 are shown in Figure 3(b)(c)(d). Given an initial duration d and initial amplitude a, mean masking impact on duration of gate G_5, $\mathrm{MMI_D}(G_5^{d,a})$, can be computed as follows. Since there are three gates G_1, G_2 and G_3 in gate G_5's fanin cone, there will be three masking impact values for $\mathrm{MMI_D}(G_5^{d,a})$.

Masking impact on duration of gate G_5 w.r.t. gate G_1: (Figure 3(b))

$$\begin{aligned}&\mathrm{MI_D}(G_1^{d,a} \rightarrow G_5) \quad \textbf{(5)}\\ &= \mathrm{P}(ADD_{G_1 \rightarrow G_5} \rightarrow 0) \cdot (d-0) + \mathrm{P}(ADD_{G_1 \rightarrow G_5} \rightarrow \tfrac{2}{3}d) \cdot (d - \tfrac{2}{3}d)\\ &- \mathrm{P}(ADD_{G_1} \rightarrow d) \cdot (d-d) = \tfrac{3}{8}(d-0) + \tfrac{5}{8}(d - \tfrac{2}{3}d) = \tfrac{7}{12}d\end{aligned}$$

Masking impact on duration of gate G_5 w.r.t. gate G_2: (Figure 3(c))

$$\begin{aligned}&\mathrm{MI_D}(G_2^{d,a} \rightarrow G_5) \quad \textbf{(6)}\\ &= \mathrm{P}(ADD_{G_2 \rightarrow G_3 \rightarrow G_5} \rightarrow 0) \cdot (d-0) + \mathrm{P}(ADD_{G_2 \rightarrow G_3 \rightarrow G_5} \rightarrow \tfrac{4}{9}d) \cdot (d - \tfrac{4}{9}d)\\ &- \mathrm{P}(ADD_{G_2 \rightarrow G_3} \rightarrow 0) \cdot (d-0) - \mathrm{P}(ADD_{G_2 \rightarrow G_3} \rightarrow \tfrac{2}{3}d) \cdot (d - \tfrac{2}{3}d)\\ &= \tfrac{5}{8}(d-0) + \tfrac{3}{8}(d - \tfrac{4}{9}d) - \tfrac{1}{2}(d-0) - \tfrac{1}{2}(d - \tfrac{2}{3}d) = \tfrac{5}{6}d - \tfrac{2}{3}d = \tfrac{1}{6}d\end{aligned}$$

Masking impact on duration of gate G_5 w.r.t. gate G_3: (Figure 3(d))

$$\begin{aligned}&\mathrm{MI_D}(G_3^{d,a} \rightarrow G_5) \quad \textbf{(7)}\\ &= \mathrm{P}(ADD_{G_3 \rightarrow G_5} \rightarrow 0) \cdot (d-0) + \mathrm{P}(ADD_{G_3 \rightarrow G_5} \rightarrow \tfrac{2}{3}d) \cdot (d - \tfrac{2}{3}d)\\ &- \mathrm{P}(ADD_{G_3} \rightarrow d) \cdot (d-d) = \tfrac{1}{4}(d-0) + \tfrac{3}{4}(d - \tfrac{2}{3}d) = \tfrac{1}{2}d\end{aligned}$$

One can note that the gate at which a glitch originates has no masking impact on that glitch; in Equation (6), the third and fourth terms are the amount of attenuation attributed to gate G_3 and should be subtracted. Subsequently, we can obtain the mean masking impact on duration of gate G_5:

$$\begin{aligned}&\mathrm{MMI_D}(G_5^{d,a}) \quad \textbf{(8)}\\ &= \frac{\mathrm{MI_D}(G_1^{d,a} \rightarrow G_5) + \mathrm{MI_D}(G_2^{d,a} \rightarrow G_5) + \mathrm{MI_D}(G_3^{d,a} \rightarrow G_5)}{3d}\\ &= \frac{7d/12 + d/6 + d/2}{3d} = \frac{5}{12}\end{aligned}$$

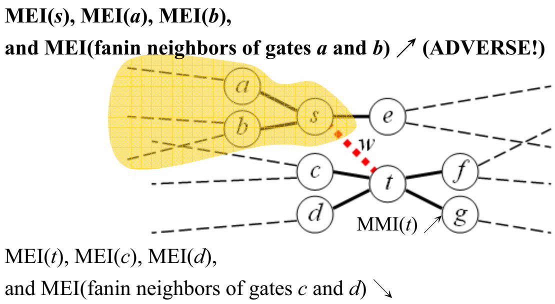

MEI(s), MEI(a), MEI(b),
and MEI(fanin neighbors of gates a and b) ↗ (ADVERSE!)
a
b
s
e
w
c
t
f
d
g
MMI(t) ↗
MEI(t), MEI(c), MEI(d),
and MEI(fanin neighbors of gates c and d) ↘

Figure 4. Changes in MEI and MMI after adding wire w ($s \rightarrow t$)

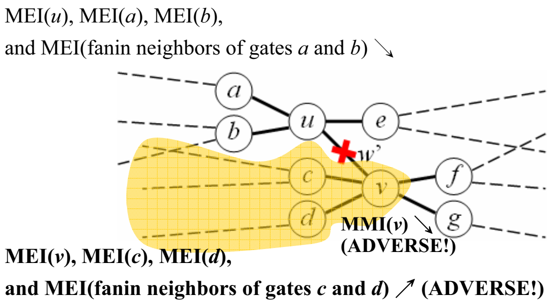

MEI(u), MEI(a), MEI(b),
and MEI(fanin neighbors of gates a and b) ↘
a
b
u
e
w'
c
v
f
d
g
MMI(v) ↘
(ADVERSE!)
MEI(v), MEI(c), MEI(d),
and MEI(fanin neighbors of gates c and d) ↗ (ADVERSE!)

Figure 5. Changes in MEI and MMI after removing wire w' ($u \rightarrow v$)

The MMI value of a gate defined by Equation (3) and shown in the above example denotes the **normalized** expected attenuation on the duration of **all** glitches passing through the gate. The expected value can be computed by traversing ADDs in linear time. Every MMI value ranges from “0” to “1” as a result of normalization. **The larger MMI a gate has, the more capable of masking glitches this gate is.** A gate with MMI equal to “0” will not attenuate any glitch at all; on the contrary, a gate with MMI equal to “1” will entirely mask glitches passing through it. This implies that those gates with higher MMI make the circuit more robust to soft errors. In general, high MMI of a gate is due to its large gate delay or considerable effect of logical masking on it. Thus, it will also be beneficial for SER if **gates with large MMI are kept** in the circuit.

5. CONSTRAINTS ON RAR

The RAR technique has two major parts, wire addition and wire removal. Each wire addition step is followed by a wire removal step, irrespective of whether or not there are any other removable wires available. For logic minimization, where the goal is the total literal count, it is easy to track the change in the number of literals after an iteration of addition and removal by just calculating the number of added and removed wires/gates. However, for SER reduction, it is not efficient to track the change in the soft error rate of a circuit by re-computing it every time. Instead, during each step of wire addition/removal we define criteria or constraints to guide us in the wire addition/removal process and check whether the step is advantageous for SER reduction.

Several constraints on the RAR technique are introduced to ensure that our framework can significantly alleviate the soft error rate of a circuit. In Section 4, we have stated the relationship between MEI/MMI and circuit vulnerability/robustness. Intuitively, one can use MEI and MMI as metrics to guide RAR towards SER reduction.

5.1. Wire Addition Constraint

Let wire w ($s \rightarrow t$) be an addible (redundant) candidate wire whose source node is gate s and destination node is gate t, as shown in Figure 4. The following three effects take place after adding wire w into the circuit:

1) The MEI values of gate s and its fanin neighbors are likely to increase because the new wire w from gate s to gate t provides an additional path for propagating erroneous values to outputs.
2) The MEI values of fanin neighbors of gate t are likely to decrease because, to some extent, the new wire w logically masks glitches from those fanin neighbors. The MEI values of some gates which are fanin neighbors of **both** gates s and t may increase, but these increases are incorporated into effect (1) above.
3) The MMI value of gate t becomes larger due to increased logical masking. The MMI values of fanout neighbors of gate t may also change (increase or decrease), but these changes do not degrade the circuit robustness since fewer glitches (with smaller duration and amplitude) pass through gate t.

Based on the definitions of MEI and MMI in Section 4, the first effect (shown in the highlighted region in Figure 4) is adverse for SER reduction, but the second and third ones are beneficial. Hence, we introduce a constraint to minimize the adverse effect.

<u>**Constraint 1**</u>: Wire w ($s \rightarrow t$) can be added into the circuit only if MEI(t) < T_1 and $\text{MMI}_\text{D}(t) > T_2$ where T_1 and T_2 are pre-specified thresholds. Intuitively, only wires having small MEI and large MMI_D for their destination gates can be added. This constraint will **keep gates with large MMI** in the circuit. To simplify the following discussion, we omit the initial duration d and amplitude a from the notations of MEI and MMI in Equations (2) and (3), but keep in mind that they actually exist.

After adding wire w into the circuit, no matter how small MEI(s) is, a **complete** glitch with initial duration and amplitude is propagated from gate s to gate t once a high-energy particle strikes gate s. That is, the resulting increase in error impact of gate s due to glitches propagated along the new connection w **does not** depend on MEI(s). More precisely, let us assume that the initial duration of a glitch occurring at gate s is d. After passing through gate t, the attenuated duration d' of the glitch can be quantified as:

$$d' = d \cdot \left(1 - \text{MMI}_\text{D}(t)\right) \tag{9}$$

If d' is smaller than or equal to the sum of setup and hold times, the glitch will be masked; otherwise, the increase in MEI(s) due to the added wire w is estimated to be:

$$\begin{aligned} \Delta\text{MEI}(s) &= \text{MEI}(t) \cdot \frac{d'}{d} = \text{MEI}(t) \cdot \frac{d \cdot \left(1 - \text{MMI}_\text{D}(t)\right)}{d} \\ &= \text{MEI}(t) \cdot \left(1 - \text{MMI}_\text{D}(t)\right) \end{aligned} \tag{10}$$

This observation is based on the fact that the duration of a glitch (if large enough) is proportional to the probability of the glitch being latched. From Equation (10), we can minimize the increases in the MEI values of gate s and its fanin neighbors by specifying a sufficiently small T_1 and a sufficiently large T_2 for MEI(t) and $\text{MMI}_\text{D}(t)$, respectively.

5.2. Wire Removal Constraint

Let wire w' ($u \rightarrow v$) be a removable (redundant) candidate wire whose source node is gate u and destination node is gate v, as shown in Figure 5. Three other effects take place after removing wire w' from the circuit.

1) The MEI values of gate u and its fanin neighbors are likely to decrease because erroneous values propagated along the removed wire w' from gate u to gate v are eliminated.
2) The MEI values of fanin neighbors of gate v are likely to increase because logical masking impact of gate v is decreased by the removal of wire w'.
3) The MMI value of gate v becomes smaller due to decreased logical masking. At the same time, the MMI values of fanout neighbors of gate v may also change (increase or decrease).

Based on the definitions of MEI and MMI in Section 4, the first effect is beneficial for SER reduction, but the second and third ones (shown in the highlighted region in Figure 5) are adverse. Hence, we set up two additional constraints: one is to maximize effect (1), the other to minimize effects (2) and (3).

Constraint 2: Wire w' (u → v) can be removed from the circuit only if $\mathrm{MEI}(v) > T_3 \geqq T_1$ and $\mathrm{MMI_D}(v) < T_4 \leqq T_2$ where T_3 and T_4 are pre-specified thresholds. Intuitively, only wires having large MEI and small $\mathrm{MMI_D}$ for their destination gates can be removed. This constraint will try to **remove gates with large MEI** from the circuit. Again, to simplify the following discussion, we omit the initial duration d and amplitude a from the notations of MEI and MMI in Equations (2) and (3).

Similar to the argument from Equation (10), the decrease in MEI(u) due to the removed wire w' is estimated to be:

$$\Delta\mathrm{MEI}(u) = \mathrm{MEI}(v) \cdot \left(1 - \mathrm{MMI}_{\mathrm{D}}(v)\right) \qquad \textbf{(11)}$$

From Equation (11), one can maximize the decreases in the MEI values of gate u and its fanin neighbors by specifying T_3 and T_4 where $T_3 \geqq T_1$ and $T_4 \leqq T_2$. The lower bound for T_3 and the upper bound for T_4 are set such that we can gain more from wire removal (e.g., ΔMEI(u) in Equation (11)) than lose from wire addition (e.g., ΔMEI(s) in Equation (10)).

Constraint 3: Wire w' (u → v) can be removed from the circuit only if $\mathrm{P}(\hat{u} = cv(v)) < T_{cv}$ over all probability distributions where $\hat{u}$ is the output value of gate u, $cv(v)$ is the controlling value of gate v, and T_{cv} is a pre-specified threshold.

The necessary condition of logical masking at gate v is that at least one of the side inputs must be the controlling value of gate v, expressed by ***cv*(*v*)**. Side inputs are those inputs on which no glitch is propagated. For instance, gate v in Figure 5 is assumed to be an OR gate (i.e., $cv(v) = 1$). If a glitch is propagated from gate c to gate v and the output value of gate u is "1", the glitch will be logically masked by the controlling value "1" from gate u. The higher probability of going to $cv(v)$ gate u has, the more likely glitches from gate v's fanin gates (except gate u) will be logically masked at gate v. Therefore, we introduce this constraint to minimize the loss on logical masking as a result of wire removal. When $\mathrm{P}(\hat{u} = cv(v))$ is large, wire w' (u → v) plays an important role in logically masking glitches passing through gate v and should not be removed.

Furthermore, for some added wires, there may be more than one corresponding removable wires which are **mutually irredundant** and cannot be removed together. In other words, removing one redundant wire will cause another one(s) to become irredundant. We sort these removable wires by the MEI values of their source gates in decreasing order. The removable wire with the largest MEI value for its source gate will be removed first. We can thus further maximize the beneficial effect (1) of wire removal and potentially remove gates with large MEI. Our overall algorithm for RAR-based SER reduction is given in Figure 6.

```
RAR-based SER reduction (circuit, $T_1$, $T_2$, $T_3$, $T_4$, $T_{cv}$) {
  Compute MEI and $\mathrm{MMI_D}$ for each internal gate in circuit;
  while (new candidate wire w identified by RAR) {
    // Wire addition procedure
    s ← source gate of wire w;
    t ← destination gate of wire w;
    if (wire w is not redundant) continue;
    if ((MEI(t) >= $T_1$) or ($\mathrm{MMI_D}$(t) <= $T_2$)) continue; // Constraint 1
    Add wire w into circuit;
    // Wire removal procedure
    gain ← 0;
    sorted_wires ← Sort all removable wires due to the addition of wire w
      by the MEI values of their source gates, from the largest to smallest;
    for each (removable wire w' in sorted_wires) {
      u ← source gate of wire w';
      v ← destination gate of wire w';
      if ((MEI(v) <= $T_3$) or ($\mathrm{MMI_D}$(v) >= $T_4$)) continue; // Constraint 2
      if (P(gate u goes to cv(v)) >= $T_{cv}$) continue; // Constraint 3
      Remove wire w' from circuit;
      gain ← gain + 1;
    }
    if ((gain > 0) or (MEI(t) is extremely small)) Keep wire w in circuit;
    else Remove wire w from circuit;
    Update MEI and $\mathrm{MMI_D}$ for affected gates;
  }
}
```

Figure 6. The overall algorithm

6. Practical Considerations

We use *mean error susceptibility* (MES) for evaluating the soft error rate of a circuit. For each primary output F_j, mean error susceptibility (MES) [13] is defined as the probability of output F_j failing due to glitches with initial duration d and initial amplitude a at internal gates, represented by $\mathrm{MES}(F_j^{d,a})$.

Next, we compute MES for all primary outputs in combinational circuits with a discrete set of pairs (d, a) of initial glitch durations and amplitudes. Therefore, the probability of primary output F_j failing due to glitches with various durations and amplitudes at different internal gates is:

$$\mathrm{P}(F_j) = \frac{\Delta d \cdot \Delta a}{(d_{\max} - d_{\min}) \cdot (a_{\max} - a_{\min})} \sum_n \sum_m \mathrm{MES}(F_j^{d_m, a_n}) \qquad \textbf{(12)}$$

Finally, the soft error rate (SER) of output F_j can be derived as:

$$\mathrm{SER}(F_j) = \mathrm{P}(F_j) \cdot \mathrm{R}_{\mathrm{PH}} \cdot \mathrm{R}_{\mathrm{EFF}} \cdot \mathrm{A}_{\mathrm{CIRCUIT}} \qquad \textbf{(13)}$$

where $\mathrm{R_{PH}}$ is the particle hit rate per unit of area, $\mathrm{R_{EFF}}$ is the fraction of particle hits that result in charge disturbance, and $\mathrm{A_{CIRCUIT}}$ is the total silicon area of the circuit.

7. Experimental Results

We have implemented the RAR-based SER reduction framework in C/C++ and conducted experiments on a set of benchmarks from ISCAS'85 and MCNC'91 suites. The technology used is 70nm, Berkeley Predictive Technology Model. The clock period (T_{clk}) used for probability computation by Equation (1) is 250ps, and setup (t_{setup}) and hold (t_{hold}) times for output latches are both assumed to be 10ps. The supply voltage is set to be 1.0V. To calculate SER by Equations (13) and (14), the allowed intervals of initial duration and amplitude are assumed to be (d_{min}, d_{max}) = (60ps, 120ps) and (a_{min}, a_{max}) = (0.8V, 1.0V), with the incremental steps Δd = 20ps and Δa = 0.1V.

For glitches with duration smaller than 60ps, the gates that will influence outputs are mostly the output gates and their fanin gates. For glitches with duration greater than 120ps, there are a considerable number of gates that will almost certainly have negative impact on outputs. This is the reason we choose (60ps, 120ps) as duration sizes for our experiments. The $\mathrm{R_{PH}}$ used is 56.5 $\mathrm{m^{-2}s^{-1}}$ and $\mathrm{R_{EFF}}$ is 2.2×10^{-5}.

Table 1 reports experimental results for SER reduction and area overhead. The area values are found using SIS technology mapping tool with MCNC library (*mcnc.genlib*). For each benchmark listed in Table 1, various glitch sizes and different input distributions are applied. We demonstrate the MES improvements from 60ps to 120ps duration sizes, as shown in columns four and five. For circuit *C432*, which has 36 primary inputs, 7 primary outputs, and 156 internal gates, the average MES of the original circuit attacked by glitches with duration 60ps is 0.00357, while that of the radiation-hardened version (optimized by our framework) is 0.00274. When the initial glitch duration is 120ps, the average MES values of the original and optimized circuits are 0.02954 and 0.02427, respectively. For initial glitches with small duration, the average MEI value is small and the average MMI value is large. In this case, there are more candidate

Table 1
Mean error susceptibility (MES) improvements
and overall soft error rate (SER) reductions

Circuit	(# PIs, # POs, # Gates)	Dur. size (ps)	Ori. Avg. MES	Opt. Avg. MES	# Add wires	# Rem wires	Area over-head	SER reduc-tion
C432	(36, 7, 156)	60	0.00357	0.00274	37	29	3.45%	21.86%
		100	0.01865	0.01345	24	12		
		120	0.02954	0.02427	24	12		
C499	(41, 32, 458)	60	0.00165	0.00140	78	41	4.67%	18.64%
		100	0.00712	0.00577	47	21		
		120	0.01086	0.00899	52	27		
alu2	(10, 6, 339)	60	0.00267	0.00222	58	46	2.67%	18.27%
		100	0.01707	0.01381	36	28		
		120	0.02740	0.02256	28	23		
alu4	(14, 8, 660)	60	0.00093	0.00081	82	60	3.69%	13.54%
		100	0.00870	0.00757	77	42		
		120	0.01464	0.01258	72	42		
t481	(16, 1, 566)	60	0.00105	0.00078	162	76	7.11%	15.91%
		100	0.07297	0.06211	136	57		
		120	0.17785	0.15939	84	23		
ttt2	(24, 21, 166	60	0.00411	0.00342	28	20	1.30%	14.88%
		100	0.01336	0.01125	13	9		
		120	0.02008	0.01712	14	11		
x2	(10, 7, 36)	60	0.01433	0.01213	4	2	3.85%	17.38%
		100	0.04378	0.03580	4	2		
		120	0.06435	0.05341	4	2		
x4	(94, 71, 288)	60	0.00221	0.00180	34	17	1.79%	18.75%
		100	0.00589	0.00476	18	10		
		120	0.00872	0.00718	19	7		
Avg.							**3.57%**	**17.40%**

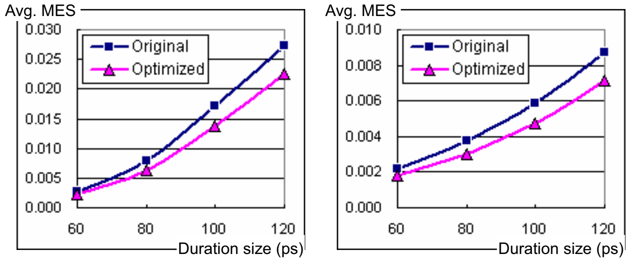
wires satisfying the wire addition constraint (Constraint 1) than the case when the initial duration is large. Hence, more added and removed wires can be expected. When considering all possible glitch sizes, in case of circuit *C432*, total area overhead is 3.45% and overall SER reduction is 21.86%. On average across all benchmarks, **17.4%** SER reduction can be achieved with **3.6%** area overhead.

In Figure 7, the MES improvements for circuits *alu2* and *x4* with various duration sizes (60ps-120ps) are presented. As it can be seen, across various initial duration sizes, a SER reduction of 16%-20% is achieved. We also perform experiments on the probabilities of output failure (by Equation (12)) over all primary outputs before and after optimization by our SER reduction framework, as shown in Figure 8. In order to make the figures more readable, we sort all primary outputs in increasing order of their original output failure probabilities. As it can be seen, in both cases, a maximum reduction of 30%-70% is achieved in output failure probability.

As mentioned earlier, the RAR algorithm used in this paper [11][12] focuses on logic minimization, which does not avoid adding wires on critical paths. Consequently, circuit performance may be degraded. The performance degradation can be suppressed by employing timing-driven RAR [14], which basically replaces a wire on a critical path with one of its alternative wires off a critical path. Finally, the proposed methodology incurs redundancy and thus certain testing schemes need to be considered for circuit testability.

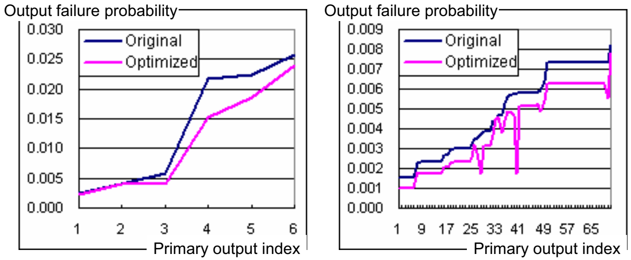
8. Conclusion

In this paper, we propose a RAR-based SER reduction framework for combinational logic. Two metrics, mean error impact (MEI) and mean masking impact (MMI), are used for efficient estimation of SER changes during RAR iterations. According to the estimation through MEI and MMI, we introduce three constraints to guide the RAR technique towards SER reduction. Experiments on a subset of ISCAS'85 and MCNC'91 benchmarks reveal the effectiveness of our framework. This framework is easily applicable to sequential circuits in conjunction with an accurate and efficient SER analyzer for sequential circuits.



(a) *alu2* (b) *x4*

Figure 7. MES improvements with various duration sizes



(a) *alu2* (b) *x4*

Figure 8. Output failure probabilities over all primary outputs before and after optimization by our SER reduction framework

References

[1] R. Baumann, "Soft errors in advanced computer systems," in *IEEE Design and Test of Computers*, pp. 258-266, Vol. 22, No. 3, May 2005.

[2] S. Mitra, N. Seifert, M. Zhang, Q. Shi, and K. S. Kim, "Robust system design with built-in soft-error resilience," in *IEEE Computer Magazine*, pp. 43-52, Vol. 38, No. 2, Feb. 2005.

[3] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," in *Proc. Int'l Conference on Dependable Systems and Networks* (DSN), pp. 389-399, Jun. 2002.

[4] K. Mohanram and N. A. Touba, "Cost-effective for reducing soft error failure rate in logic circuits," in *Proc. Int'l Test Conference* (ITC), pp 893-901, Sep. 2003.

[5] Q. Zhou and K. Mohanram, "Gate sizing to radiation harden combinational logic," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25, No. 1, Jan. 2006.

[6] Y. S. Dhillon, A. U. Diril, A. Chatterjee, and A. D. Singh, "Analysis and optimization of nanometer CMOS circuits for soft-error tolerance," in *IEEE Transactions on Very Large Scale Integration* (VLSI) *Systems*, pp. 514-524, Vol. 14, No. 5, May 2006.

[7] V. Joshi, R. R. Rao, D. Blaauw, and D. Sylvester, "Logic SER reduction through flipflop redesign," in *Proc. Int'l Symposium on Quality Electronic Design* (ISQED), pp. 611-616, Mar. 2006.

[8] R. R. Rao, D. Blaauw, and D. Sylvester, "Soft error reduction in combinational logic using gate resizing and flipflop selection," in *Proc. Int'l Conference on Computer-Aided Design* (ICCAD), pp. 502-509, Nov. 2006.

[9] M. Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," in *Proc. Int'l VLSI Test Symposium* (VTS), pp. 86-94, Apr. 1999.

[10] S. Krishnamohan and N. R. Mahapatra, "A highly-efficient technique for reducing soft errors in static CMOS circuits," in *Proc. Int'l Conference on Computer Design* (ICCD), pp. 126-131, Oct. 2004.

[11] S.-C. Chang, M. Marek-Sadowska, and K.-T. Cheng, "Perturb and simplify: multilevel boolean network optimizer," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 15, No. 12, Dec. 1996.

[12] L. A. Entrena and K.-T. Cheng, "Combinational and sequential logic optimization by redundancy addition and removal," in *IEEE Transactions on Computer-Aided Design of Integration Circuits and Systems*, Vol. 14, No. 7, Jul. 1995.

[13] N. Miskov-Zivanov and D. Marculescu, "MARS-C: modeling and reduction of soft errors in combinational circuits," in *Proc. Design Automation Conference* (DAC), pp. 767-772. Jul. 2006.

[14] Y.-M. Jiang, A. Krstic, K.-T. Cheng, and M. Marek-Sadowska, "Post-layout logic restructuring for performance optimization," in *Proc. Design Automation Conference* (DAC), pp. 662-665, Jun. 1997.