Reducing Circuit Soft Error Rate (SER): From Combinational to Sequential Circuits

Submitted to **Track 4** as 1st choice, **Track 6** as 2nd choice, and **Track 2** as 3rd choice
(Work not presented at any Ph.D. forum)

Kai-Chiang Wu (**Advisor:** Prof. Diana Marculescu)
Department of Electrical and Computer Engineering, Carnegie Mellon University
kaichiaw@ece.cmu.edu

Expected Graduation: Summer 2011
Supporting Paper: K.-C. Wu and D. Marculescu, "Clock Skew Scheduling for Soft-Error-Tolerant Sequential Circuits," in *Proc. of Design, Automation, and Test in Europe* (DATE), pp. 717-722, March 2010.

I. INTRODUCTION

Soft errors, process variations, and device aging phenomena are currently some of the main factors in reliability degradation. With the continuous scaling of transistor dimensions, soft errors, which cause unpredictable transient circuit failure, are becoming increasingly dominant for functional reliability concerns [1]. A radiation-induced charged particle passing through a microelectronic device ionizes the material along its path and generates free pairs of electrons and holes. The free (ionized) carriers deposited around the particle track can be attracted or repelled by an internal electric field of the device and lead to an electrical pulse, referred to as a *single-event transient* (SET) or a glitch. A *single-event upset* (SEU) or a *soft error* refers to transient bit corruption that occurs when a single-event transient is large enough to flip the state of a storage node. The rate at which soft errors occur is called *soft error rate* (SER).

During SEU propagation in combinational logic, three mechanisms used to provide logic circuits with effective protection against soft errors: (*i*) logical masking, (*ii*) electrical masking, and (*iii*) latching-window (timing) masking. However, as technology scaling proceeds aggressively (*e.g.*, decreasing node capacitance and increasing clock frequency), the impact of these three masking mechanisms is lessened. On the other hand, error detecting and correcting codes have been mature enough to successfully mitigate soft error susceptibility of memory elements. A recent study [2] showed that soft errors significantly degrade the robustness of logic circuits, while the nominal SER of SRAMs tends to be nearly constant from 130nm to 65nm technologies. As a result, unless explicitly dealt with, the SER of logic will become as great of a concern and is expected to be comparable to that of unprotected memories by 2011 [3].

When the combinational block of a sequential circuit can propagate SEUs freely, the sequential circuit may become very sensitive to such events. This is because, once latched, soft errors can circulate through the circuit in subsequent clock cycles and affect more than one output, more than once. The untraceable propagation of soft errors greatly affects the circuit operation for consecutive cycles and thus, necessitates design methods for soft error tolerance of sequential circuits, in a similar manner to classic design constraints such as performance and power consumption.

Having demonstrated the importance of soft errors in both combinational and sequential circuits which motivates our work, the main goal of this dissertation research is to develop a low-cost, integrated framework that can reduce the overall SER of a logic circuit. Several approaches are included to target different parts of logic circuits and poised to provide additive improvements in SER when applied in a particular order.

II. PROPOSED FRAMEWORK

Intensive research has been done in the area of SER reduction or soft error tolerance for logic circuits. The well-known *triple modular redundancy* induces excessive overhead and is unnecessary for transient (soft) errors. To reduce the overall cost for realizing soft error tolerance, partial duplication and gate resizing strategies target only nodes with high error susceptibility and ignore nodes with low error susceptibility. A potentially large overhead in area and power is still needed for a higher degree of soft error tolerance.

In this thesis, we propose three approaches for SER reduction based on (*A*) *redundancy addition and removal* [4], (*B*) *selective voltage scaling* [5], and (*C*) *clock skew scheduling* [6]. These three approaches are described further described in the sequel.

A. Redundancy Addition and Removal (RAR)

Redundancy addition and removal 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 change, we rely on estimating the effects of redundancy manipulations and accept only those with positive impact on circuit SER. Several metrics and constraints are introduced to guide the RAR algorithm toward SER reduction in a systematic and cost-effective manner.

B. Selective Voltage Scaling (SVS)

Voltage scaling is also a possible solution for SER reduction because it can mitigate SET generation. More specifically, the same amount of charge disturbance produces a smaller (less harmful) SET at gates with high supply voltage than at gates with low supply voltage. Accordingly, we assign a higher supply voltage (V_{DD}^H) selectively to gates that have large error impact and contribute most to the overall SER, and leave the remaining gates with the nominal supply voltage (V_{DD}^L). The number of gates operating at the higher voltage level, positively correlated with the power overhead, can be bounded by the appropriate use of level converters on the connections from V_{DD}^L-gates to V_{DD}^H-gates for preventing short-circuit leakage current.

C. Clock Skew Scheduling (CSS)

To address the issue of *multiple-bit upsets* (MBUs) in sequential circuits which manifest themselves as multiple errors during multiple clock cycles, affecting more than one output, more than once, we propose to exploit clock skew scheduling for MBU-aware soft error tolerance. The CSS-based approach adjusts the arrival times of clock signals to memory elements (latches or flip-flops) such that the probability of capturing unwanted transient pulses is significantly decreased, as a result of more latching-window masking. For our concern of MBU awareness, instead of using all flip-flops in a sequential circuit as candidates for CSS, flip-flops that are capable of mitigating potential MBU effects need to be extracted before applying CSS.

III. THESIS CONTRIBUTION

These three techniques (RAR, SVS, and CSS) target different parts of logic circuits. Given a logic circuit, the RAR-based approach focuses on restructuring its combinational block, while the approaches using SVS and CSS involve modifications on the power distribution and clock network, respectively. All of these proposed approaches, when integrated and applied in a particular order (*i.e.*, RAR → SVS → CSS), can thus provide additive improvements in SER. In addition, our framework as a whole has the following major and unique contributions:

- ***Symbolic unified treatment:*** The proposed framework relies on a symbolic SER analyzer [7] which provides a unified treatment of three masking mechanisms through decision diagrams. Therefore, **all** masking mechanisms, rather than one or two of them, are considered **jointly** as criteria for our objective of SER reduction. To the best of our knowledge, this is the first work reducing circuit SER with all three masking mechanisms jointly considered. Also, two novel metrics are introduced for characterizing each gate/wire in terms of masking impact and error impact. Using these two metrics, we can precisely estimate the impact on SER of a redundancy manipulation or a voltage assignment (scaling in supply voltage from V_{DD}^{L} to V_{DD}^{H}), and then decide whether to accept the given optimization step for SER reduction.
- ***Insignificant area overhead:*** Unlike some of existing SER reduction techniques based on duplication or resizing, which monotonically increase hardware resources without eliminating any, our RAR-based approach incurs very little area overhead since there usually exists one or more redundant removable wires after a redundant wire is added into a circuit. On average, only **4%** area overhead can be observed in order for a SER reduction of about **23%**.
- ***Favorable power overhead:*** The proposed approach using SVS minimizes SER while keeping the power overhead below a specified limit. To this end, level converters (LCs) are placed such that the number of up-scaled gates is bounded. It has been verified by our experiments that the appropriate use of LCs is beneficial for **power-aware** SER reduction. On average, circuit SER can be reduced by **33%** with less than **12%** energy increase, which is much smaller than those induced by other existing frameworks applying voltage scaling/assignment where LCs are avoided. Moreover, we optimize the number and distribution of required LCs for minimal design penalty and error impact due to inserted LCs. At the same time, the nets with terminal nodes operating at different voltages implicitly become fewer, which can alleviate the common layout issues coming with dual-V_{DD} or multiple-V_{DD} design style. As a fraction of total gate count, less than **4%** LCs are inserted across all benchmarks considered.
- ***Minor modification on clock network:*** The overall methodology using CSS for **MBU-aware** soft error tolerance is formulated as a piecewise linear programming problem and its optimal solution can be found by any mixed integer linear programming solver. CSS itself involves only modifications of clock tree synthesis during the physical design stage. In other words, the difference between original and optimized designs lies in their clock trees, whereas the combinational network remains identical. Hence, our CSS-based approach, when applied as a post-processing procedure, can provide additive SER reduction without destroying existing SER improvements. On average, an extra **30-40%** reduction in SER can be achieved with a drastic decline of MBU effects, while the clock network suffers a minor degree of modification ranging from **1%** up to **7%**.

IV. SUMMARY AND FUTURE WORK

In this thesis, we present three SER reduction approaches based on RAR, SVS, and CSS. All of them rely on the symbolic SER analyzer which provides a unified treatment of three masking mechanisms. However, each of them targets a different part of logic circuits, leading to orthogonal relationships and compounding results. Various experiments on a set of standard benchmarks reveal the effectiveness of our framework and demonstrate that the normalized joint cost per unit of SER reduction is relatively low when compared to other state-of-the-art techniques.

As a future direction, we plan to consider the impact of process variability on clock skew scheduling for soft error tolerance. In the presence of process variations where the analyses of masking impact and error impact are no longer fixed values but distributions (Gaussian or non-Gaussian), probability density functions are required to be modeled and the problem will be formulated in a more complex mathematical form.

REFERENCE

[1] R. Baumann, “Soft errors in advanced computer systems,” *IEEE Design and Test of Computers*, May 2005.

[2] S. Mitra *et al.*, “Robust system design with built-in soft-error resilience,” *IEEE Computer Magazine*, Feb. 2005.

[3] P. Shivakumar *et al.*, “Modeling the effect of technology trends on the soft error rate of combinational logic,” in *Proc. of Int’l Conf. on Dependable Systems and Networks*, June 2002.

[4] K.-C. Wu and D. Marculescu, “Soft error rate reduction using redundancy addition and removal,” in *Proc. of ASP-DAC*, Jan. 2008.

[5] K.-C. Wu and D. Marculescu, “Power-aware soft error hardening via selective voltage scaling,” in *Proc. of ICCD*, Oct. 2008.

[6] K.-C. Wu and D. Marculescu, “Clock skew scheduling for soft-error-tolerant sequential circuits,” in *Proc. of DATE*, March 2010.

[7] N. Miskov-Zivanov, K.-C. Wu, and D. Marculescu, “Process variability-aware transient fault modeling and analysis,” in *Proc. of ICCAD*, Nov. 2008.

Clock Skew Scheduling for Soft-Error-Tolerant Sequential Circuits*

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

Abstract

Soft errors have been a critical reliability concern in nanoscale integrated circuits, especially in sequential circuits where a latched error can be propagated for multiple clock cycles and affect more than one output, more than once. This paper presents an analytical methodology for enhancing the soft error tolerance of sequential circuits. By using clock skew scheduling, we propose to minimize the probability of unwanted transient pulses being latched and also prevent latched errors from propagating through sequential circuits repeatedly. The overall methodology is formulated as a piecewise linear programming problem whose optimal solution can be found by existing mixed integer linear programming solvers. Experiments reveal that ***30-40%*** *reduction in the soft error rate for a wide range of benchmarks can be achieved.*

1. Introduction

Soft errors, process variations, and device aging phenomena are currently some of the main factors in reliability degradation. With the continuous scaling of transistor dimensions, soft errors, which cause unpredictable transient circuit failure, are becoming increasingly dominant for functional reliability concerns [1]. A radiation-induced charged particle passing through a microelectronic device ionizes the material along its path and generates free pairs of electrons and holes. The free (ionized) carriers deposited around the particle track can be attracted or repelled by an internal electric field of the device and lead to an electrical pulse, referred to as a *single-event transient* (SET) or a glitch. A *single-event upset* (SEU) or a *soft error* refers to transient bit corruption that occurs when a single-event transient is large enough to flip the state of a storage node. The rate at which soft errors occur is called *soft error rate* (SER).

During SEU propagation in logic, three mechanisms used to provide logic circuits with effective protection against soft errors: (*i*) logical masking, (*ii*) electrical masking, and (*iii*) latching-window (timing) masking [2]. However, as technology scaling proceeds aggressively (*e.g.*, decreasing node capacitance and increasing clock frequency), the impact of these three masking mechanisms is lessened. On the other hand, error detecting and correcting codes have been mature enough to successfully mitigate soft error susceptibility of memory elements. A recent study [3] showed that soft errors significantly degrade the robustness of logic circuits, while the nominal SER of SRAMs tends to be nearly constant from 130nm to 65nm technologies. As a result, unless explicitly dealt with, the SER of logic will become as great of a concern and is expected to be comparable to that of unprotected memories by 2011 [4].

When the combinational block of a sequential circuit can propagate SETs freely, the sequential circuit may become very sensitive to such events. This is because, once latched, soft errors can circulate through the circuit in subsequent clock cycles and affect more than one output, more than once. The untraceable propagation of soft errors greatly affects the circuit operation for consecutive cycles and thus, necessitates design methods for soft error tolerance of sequential circuits, in a similar manner to classic design constraints such as performance and power consumption.

In this paper, we present an analytical methodology for soft error tolerance of sequential circuits. Our work proposes to adjust the arrival times of clock signals to memory elements (latches or flip-flops) such that the probability of capturing unwanted transient pulses is significantly decreased. The technique, called *clock skew scheduling* (CSS), is formulated in our methodology as a *piecewise linear programming* (PLP) problem, and its optimal solution can be found by existing mixed integer linear programming (MILP) solvers. The proposed framework involves only minor modifications of the clock tree synthesis step and does not touch the combinational logic of sequential circuits. Hence, this CSS-based approach can also act as a post-processing procedure for additional SER improvement on top of techniques targeting only combinational logic, which typically change the circuit timing and topology (*e.g.*, resizing [5] and rewiring [6]).

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 illustrate an example motivating clock skew scheduling for soft error tolerance. Section 4 introduces several metrics associated with SER analysis. In Section 5, our proposed framework, using clock skew scheduling and based on a piecewise linear programming formulation, is presented. Section 6 reports the experimental results for a set of standard benchmarks. Finally, we conclude our work in Section 7.

2. Related Work and Paper Contribution

2.1. Previous Work on Soft Error Tolerance

Intensive research has been done in the area of soft error tolerance for combinational circuits. To reduce the overall cost for realizing soft error tolerance, gate resizing [5] and partial duplication [7] strategies target only nodes with high error susceptibility and ignore nodes with low error susceptibility. A potentially large overhead in area and power is still needed for a higher degree of soft error tolerance. In [8] and [9], voltage scaling/assignment is used to enhance the circuit robustness to soft errors. These methods trade power penalty for SER reduction by applying higher supply voltage(s) to a certain portion of gates. Approaches based on rewiring or resynthesis [6][10] can achieve relatively smaller SER improvement while incurring little overhead.

Sequential circuits, as opposed to combinational circuits, have received less attention in terms of soft error tolerance. Since a sequential circuit has a feedback loop leading back to state inputs of the circuit, it is possible that errors latched at state lines propa-

* This research was supported in part by NSF Grant CNS-07020653.

978-3-9810801-6-2/DATE10 © 2010 EDAA

gate through the circuit for multiple clock cycles. The intuitive way to address this problem is by replacing sequential elements with hardened latches or flop-flips that are less sensitive to soft errors, as developed in [11]. A flip-flop sizing scheme [12] increases the probability of timing masking by lengthening the latching window intervals of vulnerable flip-flops. Nevertheless, this scheme does not take into account logical masking and electrical masking, which are also important factors in determining circuit SER. In [13], gates are locally relocated such that, for each gate, delays to different outputs are balanced as much as possible. In effect, this strategy minimizes the probability that an error originating at a gate is registered by any of the flip-flops. The error, however, may reach more than one output simultaneously due to balanced path delays and be registered by multiple flip-flops, resulting in so-called *multiple-bit upsets* (MBUs). For sequential circuits, MBUs imply that there will be multiple errors propagating in subsequent cycles, further degrading circuit reliability. This is a crucial reliability concern in sequential circuits that has not been addressed so far.

2.2. Paper Contribution

This paper presents a SER mitigation framework where the MBU impact is explicitly considered and alleviated. To the best of our knowledge, this is the first work addressing MBU-aware soft error tolerance in sequential circuits. On one hand, for an original error (SEU) in the clock cycle when a particle strikes, we maximize the probability of timing masking via *clock skew scheduling* (CSS). On the other hand, during clock cycles following the particle hit, we avoid multiple errors (MBU) from propagating repeatedly by exploring the effects of (*i*) *implication-based masking* and (*ii*) *mutually-exclusive propagation*, as explained later in Section 3.1 and Section 3.2, respectively. In this paper, we take advantage of intentionally induced skews to increase the probability of timing masking via CSS, while accounting for the MBU impact to further enhance soft error robustness. The contributions and advantages of our framework are twofold:

- ***Optimality*/*Complexity***: The overall methodology for MBU-aware soft error tolerance is formulated as a *piecewise linear programming* (PLP) problem and its optimal solution can be found by existing mixed integer linear programming solvers. The worst-case problem size of our PLP formulation is $\mathbf{O(n^2)}$ where n is the number of flip-flops in a sequential circuit. Therefore, the runtime spent on solving the PLP-based SER mitigation problem is quite reasonable.
- ***Compounding results***: CSS itself involves only modifications of clock tree synthesis during the physical design stage. In other words, the difference between original and optimized designs lies in their clock trees, whereas the combinational network remains identical. Hence, our CSS-based framework, when applied as a post-processing procedure, can provide additive SER reduction without destroying existing SER improvements. On average, an extra 30-40% reduction in SER can be achieved with a drastic decline of MBU effects.

3. A Motivating Example

To motivate the use of clock skew scheduling for soft error tolerance, we use benchmark *s27* (see Figure 1) from the ISCAS'89 suite, where flip-flops (FFs) are positive-edge-triggered. Without loss of generality, we assume that the delay of each gate is 1 (unit delay model) and wires do not contribute to the circuit delay. The assumption can be relaxed for a non-uniform delay model, with consideration of wire loads. In this example, we focus on a SEU which occurs at gate G_8 and may be captured by flip-flops FF_2 and/or FF_3.

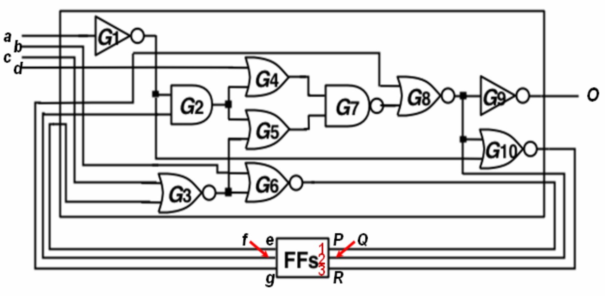


Figure 1. Example circuit *s27*

Definition 1 (*error-latching window*): The *error-latching window* [13] of a flip-flop is a time interval, $[t–t_{su}, t+t_h]$, where t is the moment when a clock edge happens, t_{su} and t_h are the setup and hold times of the flip-flop. An error must be present during this interval to be latched; otherwise, it is filtered by latching-window (timing) masking. The error-latching window associated with a flip-flop can be backward propagated to internal gates (according to respective propagation delays) to determine when an error has to occur to be latched by that flip-flop.

Under unit delay model, the delays from G_8 to FF_2 and to FF_3 are 0 and 1, respectively. Our goal is to overlap the error-latching windows of FF_2 and FF_3 at G_8 by adjusting the arrival times of clock signals to FF_2 and/or FF_3, which in effect decreases the probability that an error at G_8 is latched with increased impact of timing masking. The idea of overlapping error-latching windows, first proposed in [13], is based on the fact that the probability of timing masking is inversely proportional to the sum of sizes of disjointed error-latching windows. For example, in Figure 2(a), there are two ***separate*** error-latching windows at G_8 (one at time t-1 and the other at t) before skewing any flip-flop. If we lengthen the arrival time of clock signals to FF_3 by 1 and its new error-latching window is shown as the upper right diagram in Figure 2(b), there will be only one ***joint*** error-latching window at G_8 (at time t) due to complete overlapping. This implies that, after skewing FF_3, only errors occurring at G_8 during the error-latching window at time t will be latched, while errors occurring during the already-non-existing window at time t-1 will be filtered by timing masking, leading to a significant reduction in SER. Since the overlapped error-latching window (at time t) can be backward propagated to primary inputs, the positive impact on circuit SER is also valid for those gates in G_8's fanin cone.

However, in the case where FF_3 has been skewed, MBUs may become more frequent because an error occurring at G_8 during the joint error-latching window at time t will be latched by both FF_2 and FF_3 simultaneously. Instead of using all flip-flops in a sequential circuit as candidates for clock skew scheduling, we carefully pick pairs of flip-flops that are beneficial for MBU elimination. In the sequel, we demonstrate how to identify pairs of flip-flops that are capable of alleviating MBU effects (during clock cycles subsequent to particle hits) and suitable to be managed by CSS for ***MBU-aware*** soft error tolerance.

3.1. Implication-Based Masking

We consider the following example to illustrate the concept of implication-based masking required for our methodology. The function of primary output O of circuit *s27* is:

$$O = (a + f' + g)(c + d' + e + g) \qquad (1)$$

The complement of Boolean difference of O with respect to (w.r.t.) FF_2's present-state line f is:

$$F = (\partial O/\partial f)' = a + c'de' + g \qquad (2)$$

Equation (2) represents the Boolean expression of logical masking patterns for errors propagated from f to O.

Similarly, the complement of Boolean difference of O w.r.t. FF_3's present-state line g is:

$$G = (\partial O/\partial g)' = (a + f')(c + d' + e) \qquad (3)$$

Note that F is a function of g and G is a function of f, where f and g are present-state lines of FF_2 and FF_3 and may be corrupt due to the presumed SEU at G_8. To remove f and g while keeping the logical masking patterns, we apply *universal quantification*.

The universal quantification of F w.r.t. g is:

$$\forall_g F = F_{g=1} \cdot F_{g=0} = a + c'de' \qquad (4)$$

Equation (4) describes the patterns for logical masking of errors from f to O, for all possible values of g (0 and 1). Since we do not know whether g is corrupt, applying universal quantification makes sense and will correctly reflect logical masking of errors from f to O, irrespective of g.

Similarly, the universal quantification of G w.r.t. f is:

$$\forall_f G = G_{f=1} \cdot G_{f=0} = a \cdot (c + d' + e) \qquad (5)$$

Up to now, (4) and (5), which no longer include f or g, have been functions of inputs a, c, d, and e. In addition, one can find that (5) is a subset of (4); that is to say, with respect to O, the logical masking of an error on g ***implies*** the logical masking of an error on f. More precisely in this case, both errors on f and g will be masked when (5) is satisfied.

Definition 2 (*implication-based masking*): A pair of flip-flops X and Y is called an *implication-based masking* (IM) pair if, with respect to all outputs and flip-flops:

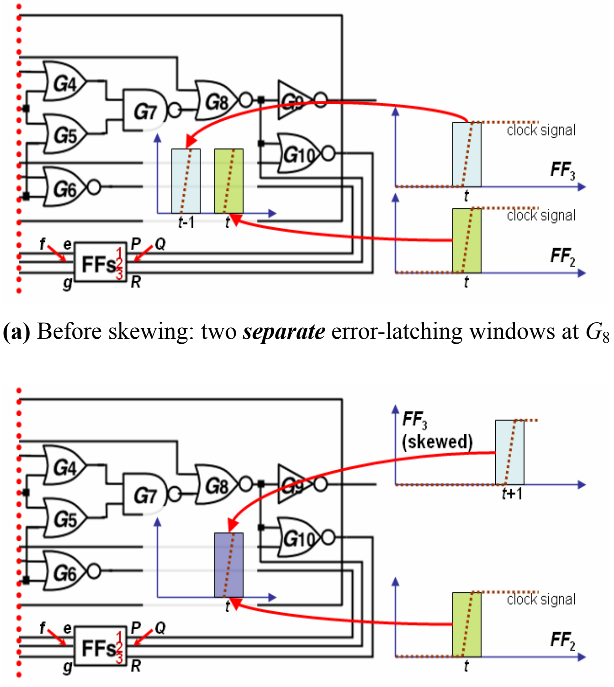
(i) the set of logical masking patterns for errors propagated from X (denoted by LM(X)) contains the one for errors from Y (denoted by LM(Y)), *i.e.*, LM(X) $\supseteq$ LM(Y), or

(ii) the set of logical masking patterns for errors propagated from Y (LM(Y)) contains the one for errors from X (LM(X)), *i.e.*, LM(Y) $\supseteq$ LM(X).

Based on Definition 2, the first category of candidates for CSS can be identified. In circuit *s27*, as shown in Figure 1, (FF_2 and FF_3) is a pair of candidates falling into this category. By overlapping the error-latching windows of these two flip-flops via CSS (see Figure 2(b)), not only can SER be reduced, but also CSS-induced MBUs will be eliminated by implication with a certain probability. This will be demonstrated in Section 6.

3.2. Mutually-Exclusive Propagation

The second type of candidate flip-flops, *mutually-exclusive propagation* pair, in *s27* can be identified by a single side-input assignment, where a side input is a wire along which no error is propagated. Again, we focus on a SEU which occurs at G_8 and may be captured by FF_2 and/or FF_3.



(a) Before skewing: two ***separate*** error-latching windows at G_8

(b) After skewing: one ***joint*** error-latching window at G_8

Figure 2. Overlapping of error-latching windows

To propagate errors from FF_3's present-state line g to R, G_{10} needs a non-controlling value "0" on its side input $G_1 \rightarrow G_{10}$. As seen in Figure 1, the value assignment at the output of G_1 is a controlling value for G_2, at which errors from FF_2's present-state line f are thus logically masked. Therefore, with respect to R, the propagation of an error on g ***implies*** that an error propagated from f is logically masked. In other words, errors on f and g cannot be observable at R simultaneously.

Definition 3 (*mutually-exclusive propagation*): A pair of flip-flops X and Y is called a *mutually-exclusive propagation* (MEP) pair if, with respect to all outputs and flip-flops, the set of logical masking patterns for errors propagated from X (LM(X)) contains the complement of the one for errors from Y (LM(Y)'), *i.e.*, LM(X) $\supseteq$ LM(Y)'. Intuitively, the sets of patterns for propagating errors from X and Y (LM(X)' and LM(Y)') are disjoint.

Based on Definition 3, the second category of candidates for CSS can be identified. Similar to IM pairs, we can overlap the error-latching windows of two flip-flops falling into this category (*e.g.*, FF_2 and FF_3 in *s27*) to achieve MBU-aware soft error tolerance because, due to the property of mutually-exclusive propagation, ***at least*** one of the two errors propagated from this pair of flip-flops will be logically masked before reaching a primary output or a flip-flop. The mutually-exclusive property guarantees that the MBU impact after applying CSS is ***at most*** equivalent to the case of not applying CSS, whereas circuit SER can be significantly reduced as a result of increased timing masking. It is also probable that two errors from a MEP pair are both masked and consequently less MBU impact is expected.

Any two flip-flops are regarded as candidates and will be beneficial for SER reduction as long as they are either IM or MEP pairs. These two properties are the major motivation for our framework aiming at soft error tolerance, and both address the MBU issue by mitigating the occurrence of multiple-bit upsets. More precisely, as mentioned earlier, *overlapping the error-latching windows of flip-flops* increases the probability of timing masking and in turn decreases the soft error rate of a circuit. Furthermore, *overlapping the error-latching windows of* ***a candidate pair of*** *flip-flops*, which meet the IBM or MEP condition, can not only reduce circuit SER but also alleviate potential MBU effects. Hence, for our objective of MBU-aware soft error tolerance, we check all possible pairs of flip-flops and extract as candidates for the proposed CSS-based framework those satisfying the IM or MEP property.

4. Analysis of Soft Error Susceptibility

Before presenting the overall methodology for MBU-aware soft error tolerance, we briefly introduce two metrics associated with SER analysis in this section. The metrics, *mean error impact* (MEI) and *mean error susceptibility* (MES), are used to evaluate the circuit susceptibility to soft errors. Relying on a symbolic framework [14][15] which provides unified treatment of three masking mechanisms through decision diagrams, MEI and MES are calculated and thereafter, the soft error rate (SER) of a sequential circuit can be derived accurately and efficiently.

4.1. Mean Error Impact (MEI) of Internal Gates

The MEI value of a gate quantifies the probability that at least one primary output is affected by an error originating at this gate. The larger MEI a gate has, the higher the probability that an error occurring at this gate will be latched. This implies that those gates with higher MEI make the circuit more vulnerable to soft errors. Please refer to [14][15] for more details about MEI.

4.2. Mean Error Susceptibility (MES) of Primary Outputs

For each primary output F_j, initial duration d and initial amplitude a, *mean error susceptibility* (MES) [14] is defined as the probability of output F_j failing due to errors at internal gates. In [14][15], the authors compute MES of each primary output in the circuit for a discrete set of pairs (d, a) of initial glitch durations and amplitudes. Then, the probability of output F_j failing (output failure probability) due to errors with various durations and amplitudes is calculated as a weighted sum of the discrete set of MES values. Finally, the soft error rate (SER) of output F_j can be derived based on the output failure probability.

5. Clock Skew Scheduling Based on Piecewise Linear Programming

The motivating example in Section 3 is a special case of CSS for MBU-aware soft error tolerance. A fundamental assumption in the example is that we can ***completely*** overlap the error-latching windows of a given pair of flip-flops (FFs) which have been recognized as candidates for CSS. This assumption is not realistic because it is not always possible to completely overlap error-latching windows without incurring any timing violations, *i.e.*, setup time violations owing to long paths or hold time violations owing to short paths. Moreover, adjusting the skew between two FFs may also change skews between affected FFs and unaffected FFs. For a large sequential circuit with hundreds of FFs, optimal skew scheduling, shown to be a signomial problem [16], is difficult to be determined algorithmically. To address this problem, we develop an analytical method which can apply CSS with a global view on all extracted candidate FFs while suppressing timing violations. A generalized problem formulation, based on *piecewise linear programming* (PLP), is presented in the sequel.

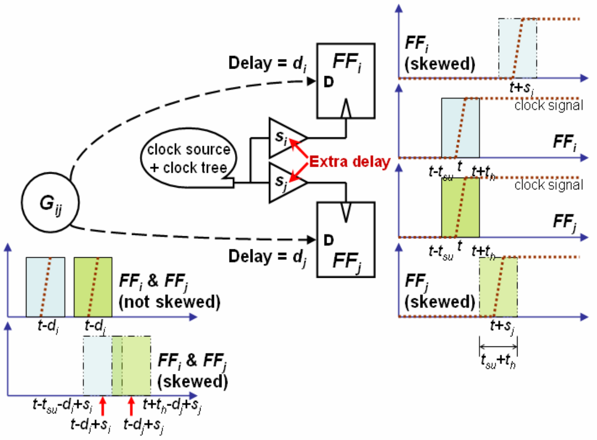


Figure 3. Generalized clock skew scheduling of a candidate pair of flip-flops (FF_i and FF_j) for MBU-aware soft error tolerance

5.1. Problem Formulation

Given a non-skewed sequential circuit (*i.e.*, *skew*(FF_i, FF_j) = 0 for all i and j) and all possible pairs of flip-flops as candidates beneficial for MBU elimination, our objective is to achieve the highest level of MBU-aware soft error tolerance by maximizing the overlap between error-latching windows of each flip-flop pair via clock skew scheduling.

Definition 4 (*intersecting gate*): The *intersecting gate* of two flip-flops FF_i and FF_j is the root gate for the intersection of FF_i's and FF_j's fanin cones. In case of more than one such gate, the one with the largest MEI value is selected.

Definition 5 (*skew*): Given two flip-flops FF_i and FF_j for which the arrival times to clock pins are c_i and c_j respectively, the *skew* between FF_i and FF_j, denoted by *skew*(FF_i, FF_j), is ($c_i - c_j$).

In Figure 3, flip-flops FF_i and FF_j are a pair of candidates whose intersecting gate is gate G_{ij}. The propagation delays from G_{ij} to FF_i and to FF_j are denoted by d_i and d_j respectively. Let the amounts of adjustments in the arrival times of clock signals to FF_i and FF_j be s_i and s_j, where s_i and s_j can be positive or negative. To completely overlap the error-latching windows of FF_i and FF_j at G_{ij}, we have to determine s_i and s_j such that *skew*(FF_i, FF_j) = ($s_i - s_j$) = ($d_i - d_j$). But complete overlapping may need significantly large $|s_i|$ and/or $|s_j|$ and thereby, may induce timing violations, which must be avoided in the resulting design. To suppress timing violations, we set up the first two constraints as follows.

For each possible pair of flip-flops FF_x (skewed by s_x) and FF_y (skewed by s_y) between which there exist combinational paths from FF_x to FF_y, (6) is to prevent setup time violations and (7), hold time violations:

$$s_x + t_{cq} + A_{xy} + t_{su} < s_y + T_{clk} \tag{6}$$
$$s_x + t_{cq} + a_{xy} > s_y + t_h \tag{7}$$

where T_{clk} is the clock period of the sequential circuit, t_{cq}, t_{su} and t_h are respectively the clock-to-output delay, setup and hold times of flip-flops, and A_{xy} and a_{xy} are the maximum and minimum delays of combinational paths from FF_x to FF_y, which can be obtained by performing static timing analysis.

Let w_{ij} denote the reduction in SER of the given circuit obtained by ***completely*** overlapping the error-latching windows of FF_i and FF_j at G_{ij}. The reason for selecting an intersecting gate with the largest MEI is that, by doing so, it is very likely to obtain the largest w_{ij} for CSS.

The theoretical optimal SER reduction is:

$$\sum_{i,j \,|\, (FF_i, FF_j) \in \text{Candidates}} \left(w_{ij} \right) \tag{8}$$

Since the optimum (8) may be unachievable due to constraints (6) and (7), we use another variable, f_{ij} ($0 \leqq f_{ij} \leqq w_{ij}$), to denote the ***actual*** reduction in SER resulting from the overlapping (complete or partial) of FF_i's and FF_j's error-latching windows. Figure 4 shows f_{ij} as a function of s_{ij} (= $skew(FF_i, FF_j) = s_i - s_j$). The rationale behind is that, once overlapped, f_{ij} is linearly proportional to the size of the overlap between FF_i's and FF_j's error-latching windows, and $f_{ij} = w_{ij}$ when completely overlapped at $s_{ij} = (d_i - d_j)$.

From Figure 4, one can note that the relationship of f_{ij} versus s_{ij} is neither convex, nor concave. Instead, the formulation becomes piecewise linear if $f_{ij}(s_{ij})$ is broken into four pieces: $s_{ij} = (d_i - d_j) - (t_{su} + t_h)$, $s_{ij} = (d_i - d_j)$, and $s_{ij} = (d_i - d_j) + (t_{su} + t_h)$. By introducing four new binary variables $p_{ij,1}$, $p_{ij,2}$, $p_{ij,3}$, and $p_{ij,4}$ such that

$$p_{ij,1} + p_{ij,2} + p_{ij,3} + p_{ij,4} = 1 \tag{9}$$

and four new floating variables $r_{ij,1}$, $r_{ij,2}$, $r_{ij,3}$, and $r_{ij,4}$ where

$$0 \leqq r_{ij,k} < p_{ij,k} \text{ for } k = 1, 2, 3, \text{ and } 4, \tag{10}$$

we can re-express s_{ij} as:

$$\begin{aligned} s_{ij} &= s_i - s_j \\ &= \left[p_{ij,1} \times (LB) + r_{ij,1} \times (d_i - d_j - t_{su} - t_h - LB) \right] \\ &+ \left[p_{ij,2} \times (d_i - d_j - t_{su} - t_h) + r_{ij,2} \times (t_{su} + t_h) \right] \\ &+ \left[p_{ij,3} \times (d_i - d_j) + r_{ij,3} \times (t_{su} + t_h) \right] \\ &+ \left[p_{ij,4} \times (d_i - d_j + t_{su} + t_h) + r_{ij,4} \times (UB - d_i + d_j - t_{su} - t_h) \right] \end{aligned} \tag{11}$$

where LB and UB are the lower and upper bounds on s_{ij}.

Similarly, f_{ij} can be rewritten as:

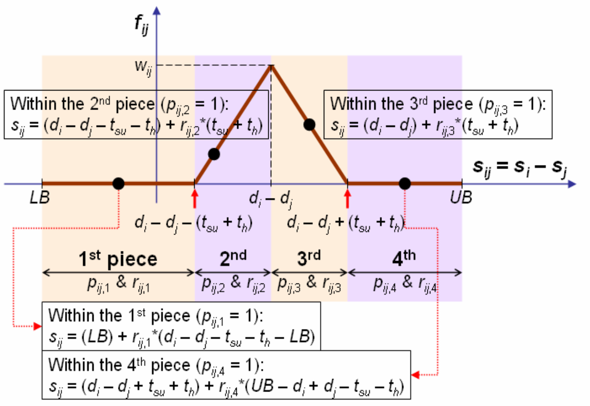
$$\begin{aligned} f_{ij} &= \left[p_{ij,1} \times 0 + r_{ij,1} \times 0 \right] \\ &+ \left[p_{ij,2} \times 0 + r_{ij,2} \times (w_{ij} - 0) \right] \\ &+ \left[p_{ij,3} \times w_{ij} + r_{ij,3} \times (0 - w_{ij}) \right] \\ &+ \left[p_{ij,4} \times 0 + r_{ij,4} \times 0 \right] \end{aligned} \tag{12}$$

Geometrically, as shown in Figure 4, $p_{ij,k} = 1$ means s_{ij} is within the k^{th} piece of $f_{ij}(s_{ij})$ and $r_{ij,k}$ indicates the ratio of s_{ij} within the k^{th} piece. For a valid solution, there must be only one among the four binary variables ($p_{ij,k}$) equal to 1 and only one among the four floating variables ($r_{ij,k}$) greater than or equal to 0. All of the other variables are 0.

Lastly, our proposed PLP-based SER mitigation framework, for MBU-aware soft error tolerance, is formulated as:

$$\texttt{Maximize} \sum_{i,j \,|\, (FF_i, FF_j) \in \text{Candidates}} \left(f_{ij} \right) \tag{13}$$

`Subject to` (6), (7), (9), (10), and (11)



f_{ij}
w_{ij}
Within the 2nd piece ($p_{ij,2}$ = 1):
$s_{ij} = (d_i - d_j - t_{su} - t_h) + r_{ij,2}{}^*(t_{su} + t_h)$
Within the 3rd piece ($p_{ij,3}$ = 1):
$s_{ij} = (d_i - d_j) + r_{ij,3}{}^*(t_{su} + t_h)$
$s_{ij} = s_i - s_j$
LB
$d_i - d_j$
UB
$d_i - d_j - (t_{su} + t_h)$
$d_i - d_j + (t_{su} + t_h)$
1st piece
2nd
3rd
4th
$p_{ij,1}$ & $r_{ij,1}$
$p_{ij,2}$ & $r_{ij,2}$
$p_{ij,3}$ & $r_{ij,3}$
$p_{ij,4}$ & $r_{ij,4}$
Within the 1st piece ($p_{ij,1}$ = 1):
$s_{ij} = (LB) + r_{ij,1}{}^*(d_i - d_j - t_{su} - t_h - LB)$
Within the 4th piece ($p_{ij,4}$ = 1):
$s_{ij} = (d_i - d_j + t_{su} + t_h) + r_{ij,4}{}^*(UB - d_i + d_j - t_{su} - t_h)$

Figure 4. f_{ij} versus s_{ij}, with four pieces that are piecewise linear: $s_{ij} = (d_i - d_j) - (t_{su} + t_h)$, $s_{ij} = (d_i - d_j)$, and $s_{ij} = (d_i - d_j) + (t_{su} + t_h)$

where (6) and (7) ensure no timing violation in the resulting circuit, and (9), (10), and (11) are used to transform the original formulation to a piecewise linear representation.

The optimal solution to (13) can be found by existing mixed integer linear programming (MILP) solvers. The worst-case problem size of our PLP formulation is **O(n^2)** where n is the number of flip-flops in a circuit. This PLP-based methodology has been experimentally verified to be very efficient in runtime, of on the order of a minute for all benchmarks considered.

6. Experimental Results

In this section, we demonstrate various experiments of our proposed framework for MBU-aware soft error tolerance. The benchmark circuits are chosen from the ISCAS'89 suite. The technology used is 70nm, Predictive Technology Model (PTM). The setup (t_{su}) and hold (t_h) times of flip-flops are both assumed to be 10ps. The overall methodology is implemented in C++, where the piecewise linear programming formulation is solved by GNU Linear Programming Kit (GLPK) version 4.33 on a 3GHz Pentium 4 workstation running Linux.

Table 1 reports the experimental results for average MES improvement and SER reduction. For each benchmark in Table 1, we list the numbers of primary inputs, primary outputs and internal gates in column two, and the numbers of flip-flops, candidate pairs along with the corresponding percentage among all possible pairs in column three. For a circuit with n FFs, we check all possible (n*(n-1)/2) pairs and extract those satisfying the IM or MEP property as candidates for clock skew scheduling. The average MES values over all primary outputs before and after applying our PLP-based CSS are shown in columns five and six, for three different initial duration sizes (small: 60ps, medium: 100ps, and large: 140ps). Columns seven and eight demonstrate the MES improvement and the overall SER reduction. The runtime spent on solving the PLP problem, which is not included in the table, is about 1 minute for circuits *s1196* and *s1238* and very few or even less than 1 second for all the others.

For example, circuit *s208* has 10 primary inputs, 1 primary output, 68 internal gates, and 8 flip-flops. Among 28 (= 8*7/2) pairs of FFs, 21 pairs (75%) can be identified as candidates for CSS. Based on (17), we formulate the CSS problem with these 21 pairs and then find its optimal solution by using GLPK. The MES improvements for small (60ps), medium (100ps), and large (140ps)

Table 1. Average mean error susceptibility (MES) improvement and overall soft error rate (SER) reduction

Circuit	#PIs	#FFs	Dur.	Ori.	Opt.	MES	SER	Norm.
	#POs	#Candi.	Size	Avg.	Avg.	Imprv.	Redctn.	Abs.
	#Gates	Pairs	(ps)	MES	MES	(%)	(%)	Adjust
s208	10	8	60	0.0099	0.0084	15.89%	29.21%	5.63%
	1	21	100	0.0409	0.0263	35.69%		
	68	75%	140	0.0767	0.0491	36.05%		
s298	3	14	60	0.0097	0.0071	27.01%	37.79%	2.85%
	14	44	100	0.0228	0.0134	41.27%		
	86	48%	140	0.0464	0.0255	45.10%		
s344	9	15	60	0.0038	0.0028	27.06%	31.35%	1.29%
	11	64	100	0.0131	0.0093	29.34%		
	131	61%	140	0.0260	0.0162	37.66%		
s400	3	21	60	0.0028	0.0022	22.81%	38.28%	1.51%
	2	36	100	0.0132	0.0095	27.84%		
	105	17%	140	0.0315	0.0113	64.17%		
s526	3	21	60	0.0020	0.0015	22.15%	40.12%	6.28%
	21	63	100	0.0108	0.0060	44.01%		
	165	30%	140	0.0214	0.0098	54.20%		
s1196	14	18	60	0.0015	0.0012	18.71%	40.41%	7.21%
	13	76	100	0.0058	0.0028	51.02%		
	487	50%	140	0.0114	0.0055	51.51%		
s1238	14	18	60	0.0014	0.0011	15.26%	33.11%	6.32%
	13	76	100	0.0052	0.0031	40.23%		
	540	50%	140	0.0101	0.0057	43.84%		
Avg.							35.75%	**4.44%**

duration sizes are 15.9%, 35.7%, and 36.1%, respectively. When considering all possible sizes of glitches, the overall SER reduction is 29.2%. On average across all benchmarks, **35.8%** SER reduction can be achieved.

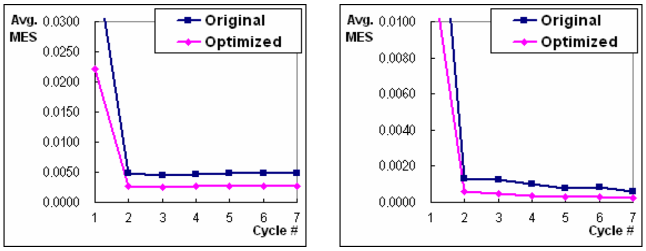
Table 1 also shows the corresponding amount of skews due to CSS. This is measured by *normalized absolute adjustment* in clock signal, which is defined as:

$$\frac{\sum_i \left|\Delta\mathrm{AT}(FF_i)\right|}{\#\mathrm{FFs}\cdot T_{clk}} \qquad \textbf{(14)}$$

where ΔAT(FF_i) is the amount of adjustment in the arrival time of clock signal to FF_i and T_{clk} is the clock period of the circuit.

Normalized absolute adjustment (14) quantifies the cost imposed by CSS in terms of the degree of clock network modification. Intuitively, the larger the value of normalized absolute adjustment, the more aggressive modification the clock network may suffer. As it can be seen in the last column of Table 1, on average **4.4%** normalized absolute adjustment is needed by our CSS-based framework. Note that the adjustment does not necessarily imply additional logic on the clock tree. For an H-tree structure, we can just unbalance wire loads during tree connection/construction to implement the skews between pairs of FFs. This is practically feasible, especially for those circuits which need significantly low adjustments in clock signals. For those circuits needing higher adjustments, wire sizing/rerouting and buffer sizing/relocation [17] are always the very first schemes for creating intentional skews.

Figure 5 shows the mitigation of MBU effects during clock cycles subsequent to particle hits (SEUs). In addition to the SER reduction for the first clock cycle via CSS, the potential CSS-induced MBU effects during the following cycles can be significantly mitigated by using IM and MEP pairs of flip-flops as candidates for CSS. On average across all subsequent cycles (from the 2nd to the 7th) in Figure 5, the MBU effects of circuits *s208* (see Figure 5(a)) and *s298* (see Figure 5(b)) can be mitigated by 43% and 63%, respectively.



(a) *s208* **(b)** *s298*

Figure 5. Mitigation of MBU effects during clock cycles subsequent to particle hits (SEUs)

7. Conclusion

In this paper, we propose an analytical method for MBU-aware soft error tolerance of sequential circuits. The approach adjusts the arrival times of clock signals such that error-latching windows of flip-flops can be overlapped, which in effect increases the probability of timing masking and decreases the soft error rate of a sequential circuit. Moreover, two types of candidate pairs of flip-flops, beneficial for MBU elimination, are introduced. The overall methodology using clock skew scheduling is formulated as a piecewise linear programming problem and can be solved efficiently by GLPK. Experiments on a set of ISCAS'89 benchmarks reveal the effectiveness of our framework.

References

[1] R. Baumann, "Soft errors in advanced computer systems," *IEEE Design and Test of Computers*, May 2005.

[2] Y. S. Dhillon *et al.*, "Analysis and optimization of nanometer CMOS circuits for soft-error tolerance," *IEEE Trans. on VLSI*, May 2006.

[3] S. Mitra *et al.*, "Robust system design with built-in soft-error resilience," *IEEE Computer Magazine*, Feb. 2005.

[4] P. Shivakumar *et al.*, "Modeling the effect of technology trends on the soft error rate of combinational logic," in *Proc. of Int'l Conf. on Dependable Systems and Networks*, June 2002.

[5] Q. Zhou and K. Mohanram, "Gate sizing to radiation harden combinational logic," *IEEE Trans. on CAD*, Jan. 2006.

[6] S. Almukhaizim *et al.*, "Seamless integration of SER in rewiring-based design space exploration," in *Proc. of ITC*, Oct. 2006.

[7] K. Mohanram and N. A. Touba, "Cost-effective approach for reducing soft error failure rate in logic circuits," in *Proc. of ITC*, Sep. 2003.

[8] M. R. Choudhury, Q. Zhou, and K. Mohanram, "Design optimization for single-event upset robustness using simultaneous dual-VDD and sizing technique," in *Proc. of ICCAD*, Nov. 2006.

[9] K.-C. Wu and D. Marculescu, "Power-aware soft error hardening via selective voltage scaling," in *Proc. of ICCD*, Oct. 2008.

[10] S. Krishnaswamy *et al.*, "Enhancing design robustness with reliability-aware resynthesis and logic simulation," *Proc. of ICCAD*, Nov. 2007.

[11] M. Zhang *et al.*, "Sequential element design with built-in soft error resilience," *IEEE Trans. on VLSI*, Dec. 2006.

[12] V. Joshi *et al.*, "Logic SER reduction through flipflop redesign," in *Proc. of ISQED*, March 2006.

[13] S. Krishnaswamy, I. L. Markov, and J. P. Hayes, "On the role of timing masking in reliable logic circuit design," in *Proc. of DAC*, June 2008.

[14] N. Miskov-Zivanov and D. Marculescu, "Soft error rate analysis for sequential circuits," in *Proc. of DATE*, April 2007.

[15] N. Miskov-Zivanov and D. Marculescu, "A systematic approach to modeling and analysis of transient faults in logic circuits," in *Proc. of ISQED*, March 2009.

[16] J. P. Fishburn, "Clock skew optimization," *IEEE Trans. on Computers*, July 1990.

[17] J. L. Neves and E. G. Friedman, "Design methodology for synthesizing clock distribution networks exploiting nonzero localized clock skew," *IEEE Trans. on VLSI*, June 1996.