Reliability Improvement against Soft Errors in Nanometer Digital Circuits

By

Warin Sootkaneung

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Electrical Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

2012

Date of final oral examination: 08/16/12

The dissertation is approved by the following members of the Final Oral Committee:

Kewal K. Saluja, Professor, Electrical and Computer Engineering

J. Leon Shohet, Professor, Electrical and Computer Engineering

Yu Hen Hu, Professor, Electrical and Computer Engineering

Stephen J. Wright, Professor, Computer Sciences

Mikko H. Lipasti, Philip Dunham Reed Professor, Electrical and Computer Engineering

Acknowledgements

First and foremost, I would like to express my utmost gratitude to my supervisor, Professor Kewal K. Saluja, for his invaluable guidance and assistance over the last five years of my graduate career. Without his immense knowledge and intuitive ideas, each step of my research sequence would not have been possible to proceed. I also appreciate his comments and endless patience for helping me review and write academic reports including several published conference/journal papers and this thesis.

Besides my supervisor, I would like to thank the other members of my thesis committee, Professor Mikko H. Lipasti, Professor Stephen J. Wright, Professor Yu Hen Hu, and Professor J. Leon Shohet for their kind assistance and valuable suggestions to improve the quality of our work and this dissertation.

I would like to extend my sincere thanks to Professor Nam Sung Kim and Professor Volkan Kursun for their kind advices and insightful discussions. I am also grateful to our friends and colleagues, who are former and current students at the Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Xaingning Yang, Shriram Vijayakumar, Ranjith Kumar, Chunhua Yao, Roengrut Rujanakraikarn, Abhishek Sinkar, Rehan Ahmed, Jungseob Lee, Hao Wang, and Spencer Millican for their kind assistance on either academic or nonacademic issues.

I would like to acknowledge the full financial support given by the Ministry of Science and Technology of Thailand. My special thanks go to the Office of Educational Affairs, Thai Embassy for professional financial management and valuable instructions.

Finally, I feel a deep sense of gratitude to my friends and families. Many thanks go to my friends for their valuable encouragement and help during my tough situations. I am forever indebted to my parents for their constant love and support. I would like to express my heartfelt thanks to both of my grandfathers, who will always be missed and loved, for their distinguished abilities in arts, sciences, and spirituality that have significant influence on me and help balance my life.

Abstract

As technology nodes are shrinking to nanometers, handling soft errors due to particle strikes in digital circuits is becoming increasingly challenging. Most of the previously proposed techniques for soft error rate reduction require high cost in terms of circuit performance penalty and area overhead. This study explores the effect of soft errors on nanometer circuits, develops soft error models, and proposes novel mitigation techniques for adaption in combinational and sequential circuits.

For combinational logic circuits, we first develop a simulator which takes into consideration gate/circuit input dependence of soft errors. Our methods to reduce soft error rate in combinational circuits include an overhead free technique called *gate input reconfiguration*, which can assign the best input configuration for soft error hardening, and two sizing based techniques employing optimization formulation and heuristics. Moreover, to improve reliability due to combinative effect of leakage and soft errors in logic circuits, we develop a novel way of combining the two into a single metric, *mean time to failure (MTTF)* and we introduce an optimal and a heuristic solution that maximize the overall MTTF through body bias.

For sequential circuits, we develop a simulator which includes input dependence of soft errors and all masking probabilities, such as electrical masking, logical masking and timing window masking. In this part, we propose an optimal flip-flop selection for soft error detection which is formulated as a binary integer linear programing (BILP) problem. This approach ensures that maximum soft error detection can be achieved for a given number of selected flip-flops. Consequently, we define a new metric called *re-execution penalty index* which represents the time penalty required to recover from a soft error. We further develop a combined flip-flop selection

and gate sizing technique which uses the re-execution penalty index to identify the best distribution of area overhead assigned to flip-flops and logic gates.

The effectiveness of each of the proposed techniques is evaluated for a number of benchmark circuits. The experimental results reveal that our techniques provide satisfactory reliability gains and outperform the previously known approaches.

Contents

Acknowledgements i

Abstract iii

Contents v

List of Figures viii

List of Tables ix

Chapter 1. Introduction 1

1.1 Impact of Soft Errors in Nanometer Digital Circuits 1

1.2 Soft Error Reduction in Digital Circuits 3

1.3 Contributions 4

1.3.1 Soft Error models and simulators for combinational and sequential circuits 4

1.3.2 Soft Error reduction in combinational logic circuits 4

1.3.3 Combined leakage and soft error optimization for combinational circuits 5

1.3.4 Soft error reduction in sequential circuits 5

Chapter 2. Related Studies 6

2.1 Input Dependence of SER 6

2.2 Sizing Techniques for SER Reduction 7

2.3 Leakage and Soft Error Interrelations 8

2.4 Flip-flop Protection against Soft Errors 10

Chapter 3. Soft Error Models and Simulators 12

3.1 Device Level Models of Soft Errors 12

3.2 Determination of Q_{crit} and Input Dependence of SER ... 14

3.3 Gate Level Soft Error Model and Simulator for Combinational Circuits ... 15

3.4 Manifestation of Soft Error in Sequential Circuits ... 18

Chapter 4. Soft Error Reduction in Combinational Circuits ... 22

4.1 Gate Input Reconfiguration Technique ... 22

4.2 Sizing Based Techniques and Their Limitations ... 24

4.2.1 Upsizing all devices ... 24

4.2.2 Upsizing transistor networks ... 26

4.3 Nonlinear Optimization Solution for Sizing ... 29

4.4 Heuristic Driven Device Size Management ... 33

4.4.1 Weighted area distribution algorithm ... 33

4.4.2 Gate SER saturation consideration algorithm ... 35

4.5 Experimental Results ... 36

4.5.1 Gate input reconfiguration technique ... 37

4.5.2 Nonlinear optimization solution for sizing ... 37

4.5.3 Heuristic based sizing ... 39

Chapter 5. Combined Leakage and Soft Error Optimization for Combinational Circuits ... 46

5.1 Leakage and Soft Error Interrelation ... 46

5.2 Defining a Common Metric ... 47

5.3 Optimal Solution ... 49

5.4 Heuristic Solution ... 51

5.5 Experimental Results 53

5.5.1 Results from the optimal solution 53

5.5.2 Justification of the heuristic based approach 54

5.5.3 Impact of body bias voltage levels 57

5.5.4 Results for reduced body bias voltage level 59

Chapter 6. Soft Error Reduction in Sequential Circuits 60

6.1 Defining a Flip-Flop Selection Problem 60

6.2 Optimal Solution for Flip-Flop Selection 62

6.3 Re-Execution Penalty Index 63

6.4 Combined Flip-Flop Selection and Sizing Technique 64

6.5 Experimental Results 66

6.5.1 Results from optimal flip-flop selection 67

6.5.2 Results from combined flip-flop selection and sizing technique 68

Chapter 7. Summary 70

7.1 Conclusion 70

7.2 Future Directions 72

7.2.1 Hierarchical approach 73

7.2.2 Soft error models for next generation devices 73

7.2.3 Power constraint optimization 74

7.2.4 Multiple-strike effect 74

Bibliography 76

List of Figures

Figure 3.1: A particle strike modeled as a current source in a two-input NAND gate 13

Figure 3.2: SE simulation framework for sequential circuits 18

Figure 3.3: Flowchart of gate level SE simulation for sequential circuits 20

Figure 4.1: Pseudocode to determine optimal input configuration for each gate...23

Figure 4.2: Q_{crit} of a 32nm two-input NAND gate as a function of sizing factor 25

Figure 4.3: Normalized SER of the circuit s15850 as a function of sizing factor for all devices 26

Figure 4.4: Normalized SER of the circuit c432 as a function of sizing factor for parallel transistor networks 28

Figure 4.5: Example Circuit 30

Figure 4.6: Flowchart of weighted area distribution algorithm 34

Figure 4.7: Flowchart of gate SER saturation consideration algorithm...36

Figure 4.8: Average SER of all experimental circuits with the area overhead of 2% (a) and 5% (b)......................44

Figure 5.1: Leakage current and SER vs. BB voltage plots for the circuit c17 47

Figure 5.2: Flowchart of heuristic driven based leakage and soft error-related reliability improvement 52

Figure 5.3: Percentage decrease in leakage current of the circuit i2 for different sets of BB voltages, w_{ZBB} = 25. .57

Figure 5.4: Percentage increase in MTTF of the circuit i2 for different sets of BB voltages with w_{ZBB} = 25..........57

Figure 5.5: BB voltages applied to each gate in the circuit i2 with w_{ZBB} = 25 58

Figure 6.1: Fault coverage table 61

Figure 6.2: Heuristic flip-flop selection pseudocode 64

Figure 6.3: Flowchart of the combined flip-flop selection and sizing technique...65

List of Tables

Table 3.1: Q_{crit} Corresponding to Input Patterns of A Two-Input NAND Gate 14

Table 4.1: Information and Normalized SER Results of Benchmark Circuits 37

Table 4.2: Optimal Results for 32, 45, 65, and 90 nm Benchmark Circuits with Area Overhead of 2% and 5% ... 38

Table 4.3: Heuristics Results for 32 nm Benchmark Circuits with Area Overhead of 2% and 5% 40

Table 4.4: Heuristics Results for 45 nm Benchmark Circuits with Area Overhead of 2% and 5% 41

Table 4.5: Heuristics Results for 65 nm Benchmark Circuits with Area Overhead of 2% and 5% 42

Table 4.6: Heuristics Results for 90 nm Benchmark Circuits with Area Overhead of 2% and 5% 43

Table 5.1: Optimal Solutions for Small Circuits with Different w_{ZBB} Values 54

Table 5.2: Heuristic Based Solutions for Small Circuits with Different w_{ZBB} Values 55

Table 5.3: Heuristic Based Solutions for Large Circuits and Others with Different w_{ZBB} Values 56

Table 5.4: Maximization of MTTF with Three Available BB Voltages for All Circuits 59

Table 6.1: Area Overhead and Normalized SER from Optimal and Heuristic Flip-Flop Selection 67

Table 6.2: Combined Flip-Flop Selection and Sizing Results 68

Chapter 1. Introduction

As device geometries are decreasing to low nanometers, particle strike induced soft errors are becoming one of the most problematic reliability issues. Circuit designers are actively researching and developing soft error mitigation techniques that provide satisfactory improvement with reasonable area and/or performance overheads. Unfortunately, adding certain soft error tolerant features in nanoscale technologies is likely to cost large overhead. On the other hand, tuning some circuit parameters to enhance circuit performance may cause an increase in soft error rate. As a result, handling soft errors is increasingly challenging, and this becomes a major responsibility of circuit designers to overcome. In this chapter, we provide a brief discussion on the definition of soft errors and their impact in nanometer digital circuits. Examples of recently proposed soft error mitigation techniques are also given. Finally, we list the contributions of this research and the organization of this dissertation.

1.1 Impact of Soft Errors in Nanometer Digital Circuits

A transient current glitch called *single event transient (SET)* in a digital circuit is generated when a particle strikes near a reverse-biased p-n junction in a CMOS device. A nuclear reaction from the strike induces charge deposition, which if sufficiently large, can complement the current output of a gate and potentially bring a circuit into faulty state when it appears at the primary output(s) or is stored in a flip-flop(s) [1]. This causes a transient error called *soft error (SE).* Two predominant sources of SETs are alpha particles from packaging impurities and neutrons from the interaction of cosmic rays in the upper earth's atmosphere [2], [3], [4]. Advances in packaging technology have significantly decreased SEs induced by alpha particles [5]. However,

the steady decrease in the node capacitance of submicron circuits, as a result of technology scaling, has led to an increase in *soft error rate (SER)* from even low-energy neutron strikes.

In addition to alpha particles and neutrons, there are other sources of SEs that may affect reliability of digital circuits. These include protons, photons, and various signal integrity deficiencies due to cross talk and power supply noises [4], [6], [7], [8]. Although protons and photons coming from cosmic rays can significantly be blocked by the earth atmosphere, the flux incident on the earth's surface is not negligible and this can be another source of SEs. In addition, in space environments, a device in a spacecraft may receive an intense flux of radiation from cosmic rays which is a serious concern especially for devices when critical dimensions are in the nanometer scale. Moreover, crosstalk and power supply noises are major signal integrity problems that can cause SEs [6], [7]. The rate of these noises, with sufficient strength to cause functional failures in digital circuits, may be significant in future generations. SEs can be created or induced in dielectric layers as well as in conductors at the chip level due to particle strikes or radiation.

SE vulnerability of digital circuits increases continuously as their SE masking probabilities decrease under the influence of technology scaling and architectural movement. Extended shrinkage in device geometries weakens the electrical masking probability. Small number of logic stages in a pipe, as a result of deep pipelining, further reduces the logical masking probability. The temporal masking probability is also diminished due to the steady increase in system clock frequency. Unfortunately, adding certain SE tolerant features in smaller technologies is likely to cost a large amount of performance overhead. In addition, post-silicon tuning approaches which require change in some circuit parameters, such as device *body bias (BB)*, for either reliability or performance enhancement may dramatically affect SER of a circuit

[9]. For instance, a part of our work in [5] reveals through simulation that although applying *reverse body bias (RBB)* to transistors can reduce leakage current, this may cause a decrease in the amount of *critical charge* (Q_{crit}) of the device, which in turn can lead to an increase in transistor SER. For this reason, more thoughtful integrated design techniques and new development of SE mitigation techniques are required to reconcile dependable systems fabricated in state-of-the-art or future nanometer technologies.

1.2 Soft Error Reduction in Digital Circuits

Many techniques have been proposed to address and reduce SEs in processors' storage units. The work in [10] validated its study on the cause of super-linear increase in *detected unrecoverable errors* (*DUEs*) in write-back caches using both actual particle strikes and simulation experiments. For SE protection, various approaches including hardware, software, and coding based techniques which are applicable to memories and *register files* (*RFs*), have been studied. Works in [11], [12], and [13] use *error correcting code* (*ECC*) based techniques to reduce SER in memories and RFs. Use of additional hardware has been proposed for main or cache memories in systems which require rigid real-time performance [14], [15]. Moreover, compiler and microarchitecture approaches proposed in [16], [17] are friendly with RFs in most processors since no power overhead from additional hardware is required. All these works focused on SEs in processor's main memories and RFs.

Unlike storage components, it is substantially more challenging to protect combinational parts from SEs since these parts are heavily reliant on hardware based techniques. The SE protection techniques used in digital circuits include hardening flip-flops [3], [18], [19], [20]; flip-flop selection, [15], [21]; filtering SE noise [22]; and sizing [23], [24], [25], [26], [27], [28], [29],

[30], [31]. Gate sizing techniques employing mathematical optimization, proposed in [26], [32], [33], can reduce SER within desired constraints. We believe that for combinational logic circuits, optimization and heuristic based gate sizing approaches are strong candidates and hence the first major contribution of this study is directed to deal with sizing based methods for combinational circuits. On the other hand, for hardening sequential circuits against soft errors we study the combination of flip-flop selection and sizing techniques.

1.3 Contributions

The main contributions of this study are as follows:

1.3.1 Soft Error models and simulators for combinational and sequential circuits

In this study, we developed SE models and simulators for combinational and sequential circuits. The input dependence of SET generation is investigated and integrated into our simulators. We include the effect of electrical and logical masking probabilities in the SE simulator for pure combinational logic circuits. On the other hand, for sequential circuits which consist of logic circuits and flip-flops, timing-window masking property is additionally taken into consideration.

1.3.2 Soft Error reduction in combinational logic circuits

An overhead-free gate input reconfiguration technique was developed based on the input dependence of SER. We use the gate input reconfiguration technique to initialize all experiment circuits with the best input configuration against SEs before performing further treatments. Additionally, we investigated limitations of conventional sizing methods when applied to small nanometer technologies and developed new sizing based techniques which offer higher yields compared to the traditional approaches. These techniques include an optimization and a heuristic based sizing to improve SE tolerance in combinational logic circuits.

1.3.3 Combined leakage and soft error optimization for combinational circuits

In this dissertation, we propose a BB based reliability optimization which integrates the effect of leakage and SER for combinational circuits. A *mean time to failure (MTTF)* is used as a common evaluation metric which combines long term reliability degradation from increased power/temperature and neutron-induced SEs. We introduce an optimization based approach and a heuristic driven methodology to maximize combined circuit MTTF through BB assignment.

1.3.4 Soft error reduction in sequential circuits

We propose two techniques to improve circuit immunity against SEs for sequential circuits. An optimal solution and a heuristic greedy algorithm for flip-flop selection were developed to maximize SE coverage. We defined a *re-execution penalty index* which represents extra time required for SE recovery. Finally, we developed an integrated flip-flop selection and gate sizing technique to improve re-execution penalty.

The dissertation is organized as follows. Chapter 2 gives a review of previously proposed works related to this study. Chapter 3 explains SE models and simulators for combinational and sequential circuits. Chapter 4 shows SE mitigation in combinational logic circuits through gate input reconfiguration and gate sizing. Chapter 5 discusses leakage and SE aware reliability optimization. Chapter 6 provides a combined flip-flop selection and gate sizing technique to limit SEs in sequential circuits. Finally, we summarize the research in Chapter 7.

Chapter 2. Related Studies

This chapter discusses the SE-related issues and provides an in depth survey of the SE reduction techniques proposed in literature which have close relation to this study. Many studies have investigated important characteristics of SE and developed accurate SE models. Among those disclosed SE characteristics, the input dependence of SE and the leakage/SE interrelation are two most distinct behaviors focused in this thesis. In addition, other studies have been developed varieties of SE mitigation in combinational and sequential circuits. These SE reduction techniques include several sizing approaches which are typically used for hardening combinational units, and flip-flop selection techniques which selectively modify some flip-flops to have capability to capture SEs in sequential circuits.

2.1 Input Dependence of SER

From extensive SPICE simulation on each gate type in a cell library, it is discovered that the SER of a circuit (gate) is dependent on the inputs to the circuit (gate) which directly impacts SET generation (electrically) as well as propagation (logically). Numerous studies by other researchers have observed this behavior [25], [27], [34], [35]. The study in [27] compared the SER of some small circuits calculated from different methodologies based on the integration of masking probabilities. That study took into account the effect of different input vectors on the sensitized paths of the SET glitches. But, it did not take into account the dependence of electrical masking probability (in both amplitude and duration of the generated SET glitch) on the input to the victim gate where a strike occurs; therefore, that work may inject unrealistic SET glitches from each particle strike. In [34] and [35], the effect of gate input on SET pulse generation was

considered. However, not all possible gate input combinations were included in those studies. As a result, this may affect the accuracy of SER estimation for small technology node circuits which are likely to be sensitive for all possible gate input combinations. In this dissertation and our previously published papers [28], [29], [30], [31], [33] which contain some of the initial results of this dissertation, we investigated the impact of gate input on the Q_{crit} (implies electrical masking during SET generation) of each transistor in the gate. Hence, this study is expected to provide satisfactory precision for addressing SER.

2.2 Sizing Techniques for SER Reduction

Sizing techniques reduce the probability of SET generation by increasing Q_{crit} of a device. This technique is generally used to harden combinational logic circuits. Recently, several optimization and heuristic driven approaches to improve SE resilience have been developed as discussed below.

A Gate sizing technique that is closely related to this work is the technique of simultaneous optimization of multiple objectives introduced in [26]. In that study, nodes with high masking probability were selected to be optimized. Unlike the nonlinear optimization formulation proposed by us in this dissertation, the objective function was defined as a linear function. Such linear formulation requires many approximations, and is likely to produce less accurate results, but it is more efficient in CPU time compared to our work. An SE optimization in a 0.18 μm standard cell circuits based on gate sizing and multi-objective genetic algorithm (MOGA) was proposed in [32]. The expression for SER included the effect of logical, electrical, and latching-window masking but neither the input dependence nor the variation on probability of neutron strike was considered in that study.

Many heuristic based techniques to combat SEs through sizing sensitive parts of a circuit have also been proposed. A fault sensitivity concept was applied to Analog-to-Digital Converters (ADCs) to increase the circuit reliability against alpha particle strikes [23]. Only mix-signal blocks in an ADC which have high probability of failure (POF) were upsized. Although its use of SPICE level simulation is not practical for large scale circuits, the local redesign approach proposed in that study is adapted in this dissertation for SER reduction in large digital circuits. A sizing technique in [25] selected sensitive gates to be upsized based only on logical masking probability of each gate. They set the target Q_{crit} equally for all selected gates and tuned the size of each gate such that the Q_{crit} goal is achieved. Unlike our work, which is an area-oriented sizing approach, the method of [25] requires much larger area overhead.

2.3 Leakage and Soft Error Interrelations

As CMOS geometries are being reduced to low nanometers, another important reliability issue for modern processers is the leakage current often responsible of larger power consumption and higher device temperature. To achieve system integration and performance requirements for a design, the scaling of threshold voltage (V_{th}) of a device is required to maintain a proper gate overdrive [36]. This causes leakage power to increase exponentially, and as a result, it becomes the largest portion of overall chip power consumption. Elevated leakage power raises the cost of cooling and packaging design to stabilize circuit thermal reliability. It also causes significant decrease in battery life for portable devices [37], and can break connections at gate terminals which may induce partial and full open defects for a wide spectrum of technologies [38]. To deal with leakage power issues as addressed above, one of the most successful approaches employs body bias (BB) tuning based techniques as they require reasonably low area overhead [39], while

return acceptably high gain for today's technology.

Many BB based techniques to reduce leakage power, in both design time and run time environments, have recently been proposed [39], [40], [41], [42], [43], [44], [45]. In these works, device V_{th} is selectively adjusted by varying device BB and thereby, changing the leakage and performance profile of the device. Particularly, V_{th} increases when reverse body bias (RBB) is applied to a device. This scenario causes the leakage to reduce but the delay to increase. On the other hand, forward body bias (FBB) lowers device V_{th} and hence, timing performance is improved at the expense of an increase in leakage. However, we know through simulation that changes in BB for either reliability or performance enhancement may dramatically affect SER of the circuit. Generally, circuit SER is relatively high compared to hard error rate due to increased power and temperature [31], [46]. Unfortunately, recent circuit design techniques focusing on SER improvement have not considered the integrated impact of leakage. Also, few works related to leakage reduction have associated SE to their proposed techniques.

Leakage power dissipation influences the permanent failure rate of a circuit considerably as follows. Increase in leakage current can cause an elevation in temperature which is an important factor of most permanent failure mechanisms including electromigration, negative bias temperature instability, thermal cycling, and time-dependent dielectric breakdown. On the other hand, SEs directly affect transient failure rate of a circuit. In [37], [46], [47], and [48] the MTTFs were defined to quantify several types of permanent failures due to power and temperature. The study in [49] investigated the SER of a 10T SRAM cell under the variations of supply voltage and BB. Nevertheless, that study did not consider leakage-related reliability problems. Our work proposed in this dissertation and a previously published paper [9] is one of the first efforts which established that SE degradation happens when leakage reduction through tuning device RBB is

performed. We found that raising the RBB voltage to lessen the leakage power can cause SER of a circuit to increase and as a result, we introduced a technique to minimize the leakage under SER bound constraint. In this dissertation, the work in [9] is improved as follows. The work in this report directly addresses the issue of maximizing the circuit reliability through the MTTF, whereas the work in [9] requires several repetitions by varying SER constraint to achieve the objective of maximized MTTF for a circuit. Further, unlike [5] which considered only RBB for leakage reduction, this improved work employs both FBB and RBB to have maximum impact on MTTF. Hence, it provides larger margin for searching optimal solution from either leakage or SER reduction while maintaining the performance of the method.

2.4 Flip-flop Protection against Soft Errors

There are numerous previously proposed SER mitigation techniques that reconfigure latches/flip-flops to protect combinational parts of a circuit against SEs or other types of transient errors. Two major mechanisms used in these techniques include error correction [18], [19], [20] , and error detection [21], [50]. A flip-flop which is hardened for error correction purpose requires additional hardware for keeping the correct value, or for changing its timing behavior to filter undesired transient glitches. This may significantly impact delay performance of the circuit. On the other hand, flip-flops designed to detect the error require less hardware overhead, yet the circuit needs to re-execute the task after the SE is detected.

For flip-flop related treatment developed in this study, we employ the detection scheme as used in [21]. Yet, our proposed optimization and heuristic solutions, which can maximize SE coverage for given number of selected flip-flops, outperform the work in [21]. The work in [20] proposes a technique which integrates flip-flop selection and sizing. The flip-flop selection

algorithm in [20] uses the slack available at flip-flops and hardens them with additional delay overhead. Therefore, it may yield little or no improvement for circuits where most flip-flops are in long paths. Also, the method of [20] is unable to directly address the impact of SEs that are not captured at selected flip-flops. In this work, our technique addresses the SE-related reliability degradation with the re-execution penalty index. This common metric can treat two SER reduction gains from two different techniques fairly.

Chapter 3. Soft Error Models and Simulators

Many SE models have been developed to capture the impact of SEs at different levels of abstraction of a design. In this research, we adopt a well-known device level model from [51], [52] and integrate it into our gate level SE models for combinational and sequential circuits. The advantages of handling SEs in the gate level are as follows. First, at lower levels (device or transistor, and gate levels), SE generation can be accurately modeled since the investigation domain is bounded around the area where the exact reaction from particle strikes happens. Second, SET propagation and appearance of SEs are easily tractable using logic simulation. Third, managing SEs at lower levels requires less performance penalty compared to recovering SEs at higher levels.

In this chapter, to begin with, device level models of SE are briefly discussed. Next, we give a detailed explanation on the input dependence of SE. Finally, our proposed gate level SE simulators for addressing SER in combinational and sequential circuits are described.

3.1 Device Level Models of Soft Errors

When a device is hit by a particle, the amount of charge deposition generated by the strike increases and causes the current to flow through the body-drain junction of a transistor. If the amount of charge deposition is sufficient to flip the logic level of a gate output (the change in output voltage is over $V_{DD}/2$), we call this a *critical charge* (Q_{crit}). The amount of Q_{crit} of each transistor is dependent on gate type and gate input.

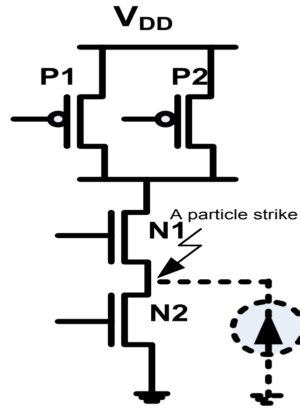


Figure 3.1: A particle strike modeled as a current source in a two-input NAND gate

Figure 3.1 shows a current source model of a particle strike placed at the drain of a victim transistor in a two-input NAND gate. In order to identify Q_{crit}, one of the following two current equations is used depending on the type of particle. The double exponential current source [53] as given in (3-1) is generally employed for alpha-particle strikes.

$$I(t) = \frac{Q}{\tau_1 - \tau_2}\left(e^{-t/\tau_1} - e^{-t/\tau_2}\right) \quad (3\text{-}1)$$

For neutron strikes, the single exponential current source model [51], [52], given in (3-2), is widely used.

$$I(t) = \frac{Q}{\Gamma}\sqrt{\frac{t}{\Gamma}}e^{-t/\Gamma} \quad (3\text{-}2)$$

In (3-1) and (3-2), Q is the amount of charge deposition. τ_1,τ_2, and Γ are the charge-collection time constants which are technology dependent. The direction of the current depends on whether the strike happens on a PMOS or an NMOS device [54]. We must point out that various other current source models, such as triangular current, have been studied in [55], which concludes that different shapes of the SET waveform profoundly affect Q_{crit} or SER of a circuit. However, exponential model, also studied in [55], is the most prevalent model.

3.2 Determination of Q_{crit} and Input Dependence of SER

The values of Q_{crit} for a two-input NAND gate, shown in Figure 3.1, given in Table 3.1, are obtained from SPICE simulation using (3-2). The gate is designed for minimum size such that it has symmetric rise/fall delay. All gates consist of complementary transistor network mapped with 32, 45, 65, and 90 nm predictive technologies from [56]. Note in Table 3.1 some transistors are shown as "not sensitive" to a particle strike (marked as "NS" which means "Not Sensitive"), because these transistors have either very high Q_{crit} or a particle strike on these transistors cannot cause a change in voltage level to be deemed as an opposite transition. Thus, the Q_{crit} is given only for the sensitive transistors in Table 3.1.

Further, Table 3.1 clearly shows that the smaller the technology node, the smaller the amount of Q_{crit}. This is because a decrease in device size causes device capacitance to decrease, thereby raising its sensitivity to SE glitches. In addition, the amount of Q_{crit} also relates to the amount of capacitance shared among nearby devices around the strike area which relies on the inputs to a gate. For instance (see Figure 3.1), in the case that a strike occurs at the drain of the transistor N1 of a 2-input NAND gate, for the input "00"; i.e., both transistors N1 and N2 are "OFF", the amount of charge stored in the parasitic capacitor of N2 is larger than for the input "01"; i.e., N2 is "ON" and its parasitic capacitor is completely discharged. Since large amount of charge at the

Table 3.1: Q_{crit} Corresponding to Input Patterns of A Two-Input NAND Gate

Input	**Q_{crit} (fC)**															
	P1				***P2***				***N1***				***N2***			
	32 nm	***45 nm***	***65 nm***	***90 nm***	***32 nm***	***45 nm***	***65 nm***	***90 nm***	***32 nm***	***45 nm***	***65 nm***	***90 nm***	***32 nm***	***45 nm***	***65 nm***	***90 nm***
00	NS	NS	NS	NS	NS	NS	NS	NS	4.4	7.9	12.5	20.2	9.6	61.2	NS	NS
01	NS	NS	NS	NS	NS	NS	NS	NS	1.6	3.5	6.1	9.4	15.6	23.8	40.7	57.6
10	NS	NS	NS	NS	NS	NS	NS	NS	1.6	3.5	6.2	10.1	1.6	3.5	6.2	9.9
11	3.5	5.9	7.4	13.9	3.5	5.9	7.4	13.9	NS	NS	NS	NS	NS	NS	NS	NS

strike node can prevent the faulty current to pull the output down, the amount of Q_{crit} of the transistor N1 for the input "00" is larger than for the input "01".

We performed SPICE simulation to determine Q_{crit} for all the gates in the cell library mapped with different technologies, similar to the experiment conducted in [57]. However, unlike [57], in our experiment, we consider all possible input combinations. Furthermore, in order to capture propagating property of SET glitches, a fixed load was added at the output of each gate type in the library and a glitch that has sufficient amplitude and duration to appear at the output of the load defines the Q_{crit}. After Q_{crit} of each transistor corresponding to each input vector had been identified, we found that all gates have input dependence behavior, and many of them are quantitatively similar to the two-input NAND gate. As a result, we combine the input dependence of SER into SER estimation as discussed below.

3.3 Gate Level Soft Error Model and Simulator for Combinational Circuits

The charge deposition from the ionization of the material caused by a particle hit is dependent on the particle energy, technology, and the material [1]. We adopt an energy transfer model from [1] to trace the charge deposition back to the energy of the striking particle which produces Q_{crit}. In other words, it causes a circuit to fail. For the case of neutron strikes, the focus of this study, we use the JEDEC89A Standard [58] to obtain the total neutron flux above the strike energy which can be directly interpreted as a *strike rate per unit area*, $\emptyset_{i(t,j)}$ for a gate *i*, transistor *t*, and gate input *j*. In the standard, neutron differential flux as a function of neutron energy is provided. To obtain the total neutron flux, we need to integrate the neutron differential flux over the neutron energy. Although the standard contains a very large range of neutron energy, we simplify the upper limit of integration and set the value of max Q_{crit} to 120 fC. For silicon, this corresponds to

6 MeV energy level of the ionizing neutron and probability of energy level higher than this is nearly zero [24].

Once an error occurs on a transistor or a gate, electrical and logical masking properties of the circuit can prevent the error to appear at a primary output. While Q_{crit} contains the electrical masking property of a gate, the *error count*, $E_{i(j)}$ of transistor *i* and gate input vector *j* is defined to capture the logical masking characteristic of the target circuit. This *error count* can be obtained by performing extensive logic simulations and SET injections. In our simulation process, for each randomly generated input to a circuit, the output logic of each gate, one at a time, is flipped to its complementary value and $E_{i(j)}$ is updated only when the fault can propagate to the circuit primary output. Using (3-3), we can obtain $Tr\ SER_{i(t,\ j)}$ which is the SER of transistor *t* of a gate *i* for the gate input vector *j* [28].

$$TrSER_{i(t,j)} = \frac{1}{k} * \emptyset_{i(t,j)} * Ad_{i(t)} * w_{i(t)} * E_{i(j)} \quad (3\text{-}3)$$

In the above equation, *k* is the total number of simulated input vectors, $\emptyset_{i(t,j)}$ is the strike rate per unit area or the total flux of neutron which can produce the charge deposition greater than Q_{crit} at transistor *t* of a gate *i* and gate input vector *j*, $Ad_{i(t)}$ is the *active area* of transistor *t* of gate *i* (drain area of the transistor which was hit), and $w_{i(t)}$ is the *weighting factor*, the ratio of the *active area* to the *circuit area*. Since we assume that only one device in a circuit fails at a time, the SER of gate *i*, $GateSER_i$, can be calculated by summing the transistor SER in (3-3) as given in (3-4).

$$GateSER_i = \sum_{i(j)} \sum_{i(t)} TrSER_{i(t,j)} \quad (3\text{-}4)$$

Next, the total circuit SER, *CircuitSER* can be written as the summation of SER of each gate *i* as shown in (3-5).

$$CircuitSER = \sum_i GateSER_i \quad (3\text{-}5)$$

CircuitSER in (3-5) has a unit of per second and its reciprocal can be interpreted as the *mean time to failure* (*MTTF*) due to SEs in a circuit.

In summary, to build an SE simulator for combinational logic circuits, first of all, HSPICE is used to determine Q_{crit} values of each gate in the cell library with the original size. After Q_{crit} information has been collected, our JAVA based simulator performs following tasks:

a) *Scaling Q_{crit} corresponding to device size*

b) *Mapping Q_{crit} to neutron flux*

c) *Performing SET injection and logic simulation to obtain the error count*

d) *Calculating the circuit SER*

Since the circuit SER obtained from our SE simulator is workload dependent, to achieve exact results, we require actual input traces to precisely estimate the circuit SER. Although this study determined the circuit SER using randomly generated inputs, our simulator and simulation methodology, without any changes, allow the users to apply any workload traces to arrive at workload driven conclusions.

We use this SE simulator to explore SE in nanometer logic circuits, and to evaluate our proposed SE mitigation techniques as provided in the following chapters. For combinational circuits, we assume that the timing-window masking probability has no impact on the SER. However, to achieve precise value of the SER of logic parts with pipelining, information about circuit frequency, pulse-width of SET glitches, and timing characteristic of flip-flops is required to obtain timing-window property. We discuss a SE model for sequential circuits or pipelined combinational circuits in the next section.

3.4 Manifestation of Soft Error in Sequential Circuits

As discussed in the previous section, the Q_{crit} of each gate relates to the amount of atmospheric neutron energy which brings the gate to faulty state. Due to technology scaling, Q_{crit} is steadily decreasing, and this means that the device cannot tolerate low energy neutrons which has high rate of strike. In addition, small amount of Q_{crit} extends the width of SET and hence, this reduces electrical masking probability when a SET is propagating to the circuit output. The larger width of the faulty glitch also increases the probability that the SET can appear within the latching window of a flip-flop which consequently stores an incorrect value. For the reasons explained above, SE tolerance in nanometer sequential circuits is sharply degrading.

For sequential circuits, we also use the information of atmospheric neutron recorded in the JEDEC89A standard [58] and the energy transfer model from [1] to map the charge deposition due to neutron strike to the probability that the strike with corresponding energy can be found [24]. However, unlike addressing SEs in combinational circuits, we use this probability of strike to weight randomly generated charge deposition for each SET injection. We assume that a transient current from the strike is totally masked if the amount of charge deposition of the induced SET is smaller than Q_{crit}.

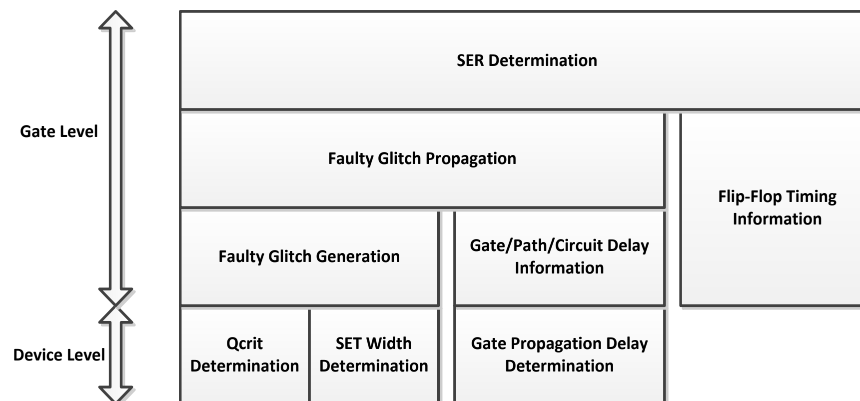


Figure 3.2: SE simulation framework for sequential circuits

The framework for SE simulation for sequential circuits is illustrated in Figure 3.2. The simulation procedures involve device and gate levels of the design. In device level, we performed SPICE simulation to determine Q_{crit}, SET width, and propagation delay for each mapping gate in the library. After that, our Java based gate level simulator was used to estimate overall SER of the circuit. Gate level simulation includes faulty glitch generation and propagation, and SER estimation. Further, gate/path/circuit delay and flip-flop timing information is also taken into account to capture the electrical and timing window masking probabilities of the circuit. The detailed process of the gate level simulation is discussed as follows.

The flowchart in Figure 3.3 shows our gate level SE simulation methodology for sequential circuits. To begin with, we randomly assign input to the circuit for each cycle. Then, a SET is injected to the output of each gate, one at a time. The value of charge deposition of the SET is weighted using the probability of atmospheric neutron strike as discussed before, and the time of the strike is randomly assigned within the circuit cycle time. Next, we check if the amount of charge deposition of any SET is greater than the Q_{crit} of that gate. If it is, we acquire corresponding SET pulse width from SPICE simulation and continue to propagate the faulty glitch. During fault propagation, in addition to performing logic simulation, we take into consideration the propagation delay of each gate obtained from the previous SPICE simulation to update timing characteristic of the propagating glitch(es) along the sensitized path(s). However, apart from being logically masked, any glitch that has time duration smaller than gate propagation delay is filtered out from the gate output. Finally, circuit SER can be estimated based on the appearance of glitches at primary output(s) or within the latching window of a flip-

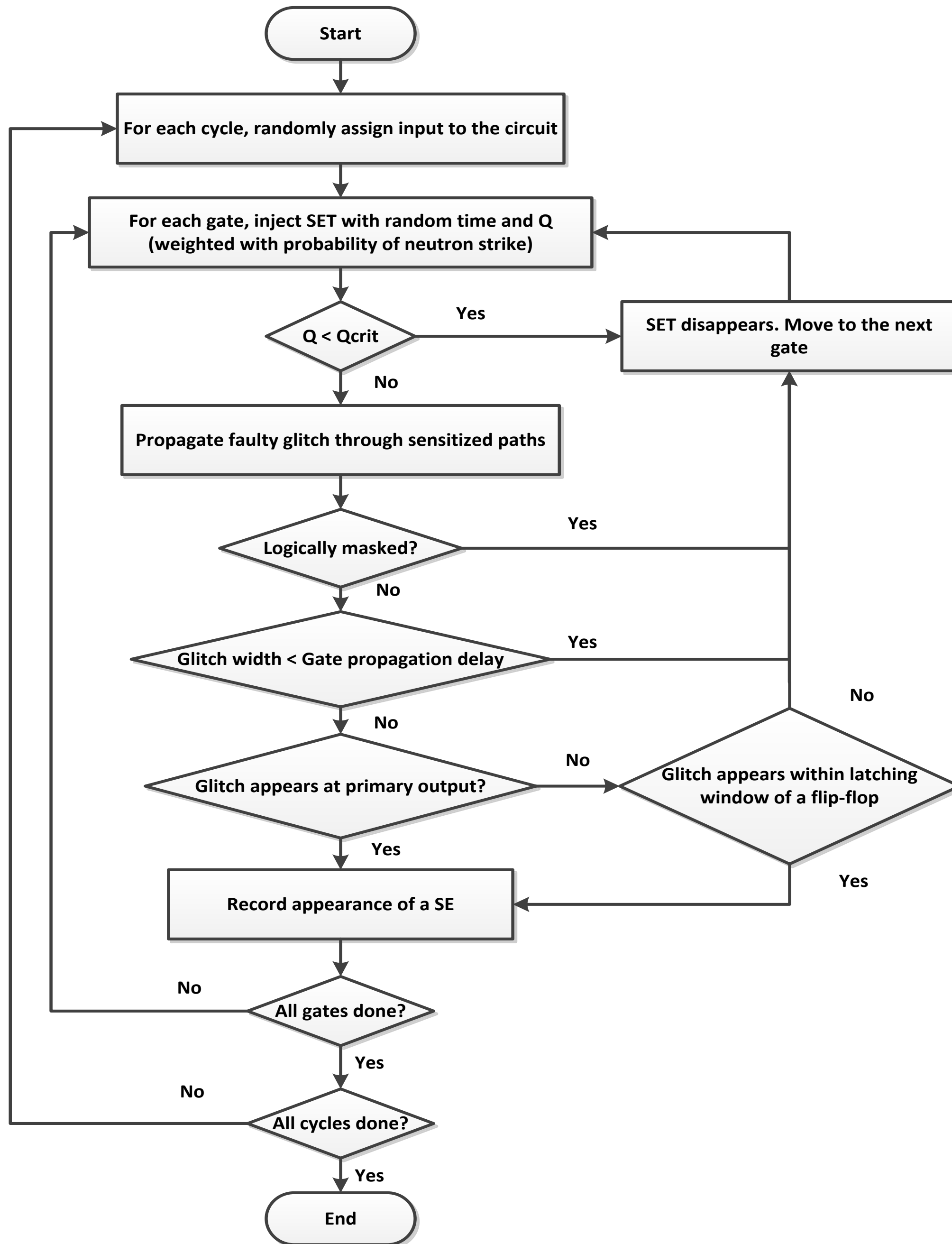


Figure 3.3: Flowchart of gate level SE simulation for sequential circuits

flop during simulating time frame. We assume that only single node is hit once at a time. Thus, using equation (3-6) below, we can obtain the circuit SER.

$$circuit\ SER = \frac{\sum_{i=1}^{i=k*\#gates} E_i * w_i}{k * cycle\ time} \tag{3-6}$$

In (3-6), E_i is 0 if the injected SET does not cause a SE, and it is 1 if the SET appears at an output or flip-flop, w_i is the ratio of active area of the gate being hit to the circuit area, and k is the number of total cycles which is equal to total number of simulated input vectors as used in (3-3).

Chapter 4. Soft Error Reduction in Combinational Circuits

In this chapter, we discuss a gate input reconfiguration technique which is developed based on the input dependence of SER. This technique rearranges gate input pins to the best configuration providing the minimum SER for each gate for a given workload. The most attractive part of this technique is that it requires no area overhead. Furthermore, an investigation on limitations of conventional sizing methods when applied to small nanometer technologies is discussed. The conclusion drawn from this investigation led us to further develop new sizing scenarios which offer higher yields compared to the traditional approaches. We introduce an optimization based sizing technique for SER mitigation to overcome the impact of small device geometries. In this technique, a nonlinear optimization problem is formulated for SER minimization with area overhead as a constraint. We also introduce a heuristic sizing approach with high efficacy in runtime compared to searching for nonlinear optimization solutions which requires large CPU runtime. Finally, experimental results are provided to evaluate the performance of our proposed methods.

4.1 Gate Input Reconfiguration Technique

The input dependence of SER led us to develop a gate input reconfiguration technique. Initial results using this method appear in our paper [29] presented at a workshop. The algorithm of the gate input reconfiguration technique is provided in this section, and complete results of this technique are given in section 4.5.

We notice from Table 3.1 in Chapter 3 that Q_{crit} of N2 for the input “01” is larger than for input “10”; therefore, it is fair to conclude that the SER of N2 when the input is “01” must be

```
// Start at gate i = 1
For (i = 1; i <= total number of gates; i++)
      {
// Start at input configuration l = 1.
/* Note for a gate i with n_i input pins, there are n_i! configurations.
*/
      For (l = 1; l <= n_i! l++)
            {
             Calculate GateSER_i for each config_l using (3-3) and (3-4);
            }
      GateSER_i(config_optim) = min[GateSER_i(config_l)];
      }
Calculate CircuitSER using (3-5);
```

Figure 4.1: Pseudocode to determine optimal input configuration for each gate

lower than SER when the input is “10”. This suggests that for any two-input NAND gate which sees the input value of “10” much more often than the input value of “01”, if its two input signals are switched or the position of the two NMOS transistors is interchanged, then we should see a reduction in the gate SER. This observation led us to develop the gate input reconfiguration technique to reduce the SER by rearranging gate inputs to each gate such that every gate has lowest SER. It is important to note that when the inputs to a gate are reconfigured, they do not change the functionality of the circuit or the inputs to other gates. Therefore, optimal input configuration for each gate can be found without affecting the optimality of configuration of other gates in the circuit.

The best input configuration for each gate is given as follows. We reorder inputs of gate i, which has n number of input pins and calculate the *GateSER* using (3-3) and (3-4) in Chapter 3 for each possible configuration $config_l$. The optimal value of a configuration, $config_{optim}$, for a gate i is chosen using equation (4-1) and the pseudocode of the algorithm to achieve this is given in Figure 4.1.

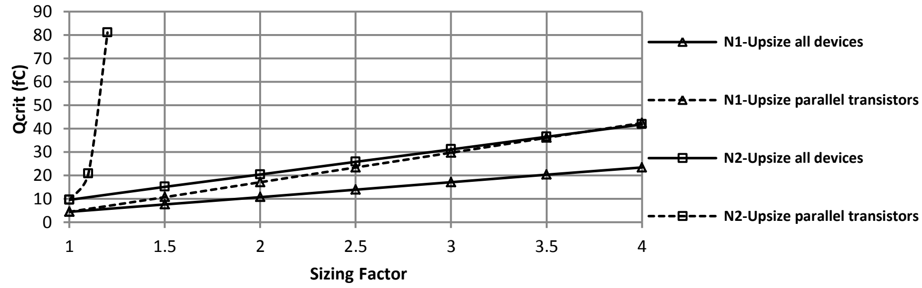
$$GateSER_{i(config_{optim})} = \min\left[GateSER_{i(config_{l})}\right], l \in \{1,..,n!\} \quad (4\text{-}1)$$

4.2 Sizing Based Techniques and Their Limitations

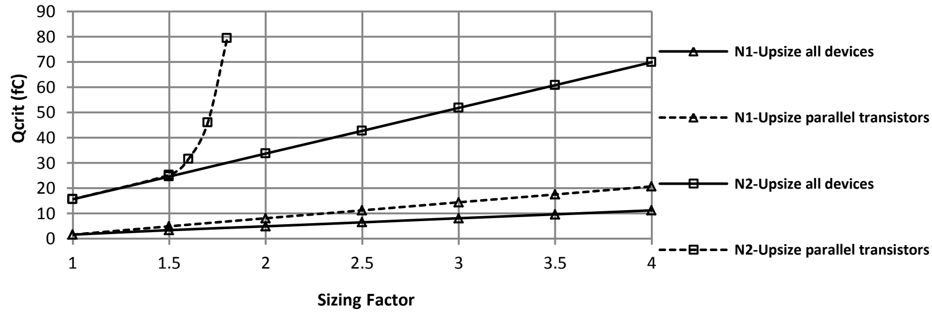
An upsizing technique is a SER reduction technique which increases the Q_{crit} by increasing the width of a device. This approach is generally used in combinational circuits, since it has low performance penalty. In this section, to begin with, we discuss a traditional upsizing method in which all devices are upsized by a constant factor. Next, we investigate the other sizing technique in which transistor networks, parallel or serial, of the gates are selectively upsized.

4.2.1 Upsizing all devices

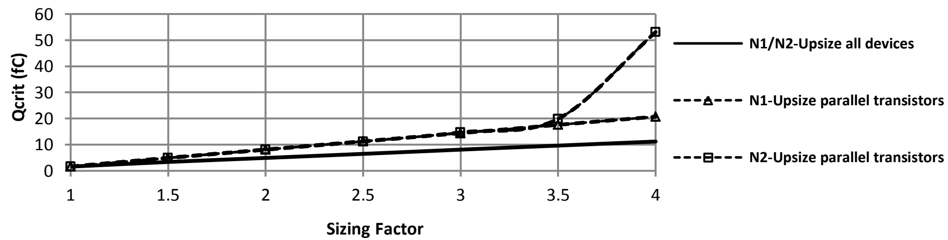
The simplest way to limit circuit SEs is upsizing all transistors with the same upsizing factor. This approach increases all capacitance components of a gate and raises Q_{crit} of a transistor for all possible input patterns. As a result, the circuit achieves more tolerance to particle strikes, resulting in reduced circuit SER. However, this simple explanation is not valid for very small geometries. We conducted an experiment in which we computed the value of Q_{crit} for the sensitive transistors for a two-input NAND gate implemented in 32 nm technology. For each input combination, we considered various upsizing options as shown in Figure 4.2. In this figure, **solid line** plots show the value of Q_{crit} as a function of sizing factor when all transistors are upsized. It can be seen in Figure 4.2 that this sizing method offers small increase in Q_{crit} but it may cause an increase in particle strike rate due to large increase in the active area of all transistors. This traditional sizing approach may provide poor SER reduction or even degrade SE reliability for some small technologies.



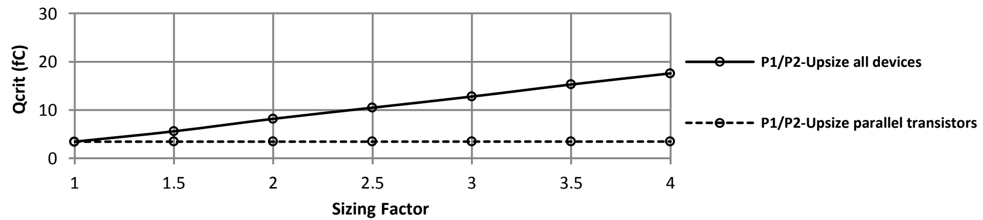
(a) For input logic "00", N1 and N2 are sensitive



(b) For input logic "01", N1 and N2 are sensitive



(c) For input logic "10", N1 and N2 are sensitive



(d) For input logic "11", P1 and P2 are sensitive

Figure 4.2: Q_{crit} of a 32nm two-input NAND gate as a function of sizing factor

The evidence is given in Figure 4.3 which plots the normalized SER vs. sizing factor, distributed to all devices, for the benchmark circuit c6288 mapped with different technology

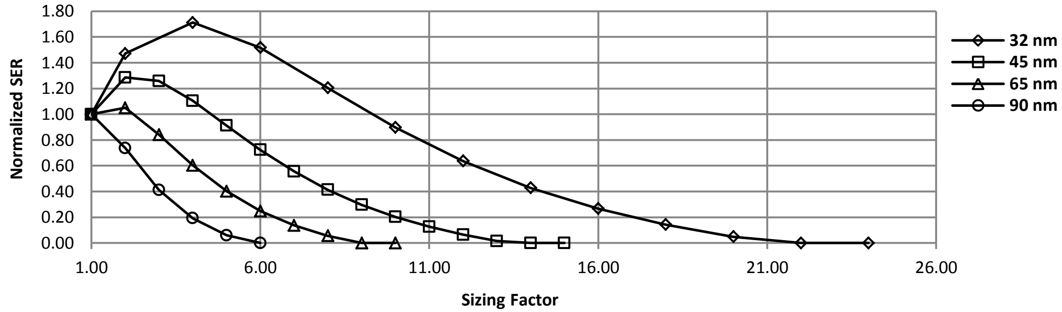


Figure 4.3: Normalized SER of the circuit s15850 as a function of sizing factor for all devices

nodes. The normalized values of SER give the relative change in SER of the upsized circuit compared to the original circuit. Thus, in this plot, the SER of the original circuit is assumed to be 1.The value of SER greater than 1 means that SER of the upsized circuit is larger than the original circuit. In addition, the initial size was chosen such that the devices were of minimum size while maintaining the symmetrical rise/fall-delay property, and the input pins of each gate were reconfigured using our gate input reconfiguration technique discussed in section 4.1.

It can be concluded from the data in Figure 4.3 that for 32 nm, 45 nm and 65 nm technologies, there exist degradation regions in which the normalized SER increases when additional area is given to the circuit. However, we also notice that such effect does not occur for 90 nm technology nodes. We conducted similar experiment for 130 nm nodes and the results were similar to that for the 90 nm technology nodes. As a result, some changes to the traditional sizing are needed for sub-90 nm circuits to be SE resilient.

4.2.2 Upsizing transistor networks

We further investigated the upsizing scenario which offers large Q_{crit} improvement with small additional area. This made the increase in neutron strike rate not significant. The **dash lines** in Figure 4.2 show the relationship between the Q_{crit} and sizing factor of a two-input NAND for each possible input value when its parallel transistors are upsized. From those graphs in Figure

4.2, upsizing parallel network (P1 and P2 are upsized) provides larger increase in Q_{crit} of N1 and N2 (see Figure 4.2 (a), (b), and (c)) compared to upsizing all devices, but it gives small improvement in Q_{crit} of parallel transistors, P1 and P2 (see Figure 4.2 (d)). This effect can be explained as follows. For any n-input NAND gate, upsizing parallel transistors increases driving capability of the pull-up transistors, so it can prevent a downward output transition due to a particle strike on one of the serial transistors. On the other hand, upsizing parallel transistors returns no increase in their Q_{crit} because it yields no increase in the pull-down current, retarding a rising SE glitch from a particle strike on one of the parallel transistors. This situation is inversely consistent for upsizing serial transistors which can increase the Q_{crit} of parallel transistors for the 2-input NAND and other gates.

In summary, based on extensive device level simulation, we can make the following two statements:

a) *For all conventional CMOS logic gates, the serial transistor network tends to be most sensitive to SEs.*

b) *The Q_{crit} of one complementary transistor network can be increased sharply by upsizing the opposite network.*

The first statement follows from the result shown in Table 3.1 in Chapter 3. It is evident that for a two-input NAND gate, the stack of NMOS transistors (the pull-down network) is sensitive for most inputs except “11”. For an n-input NAND gate, the stack of NMOS transistors is insensitive for the input of “1…1”. In the same manner, for an n-input NOR gate, the stack of PMOS transistors (the pull-up network) is sensitive for most inputs except “0…0”. Although the frequency of each gate input pattern is dependent on the workload, probabilities that the input of any n-input NAND gate is “1…1” and the input of any n-input NOR gate is “0…0” are expected

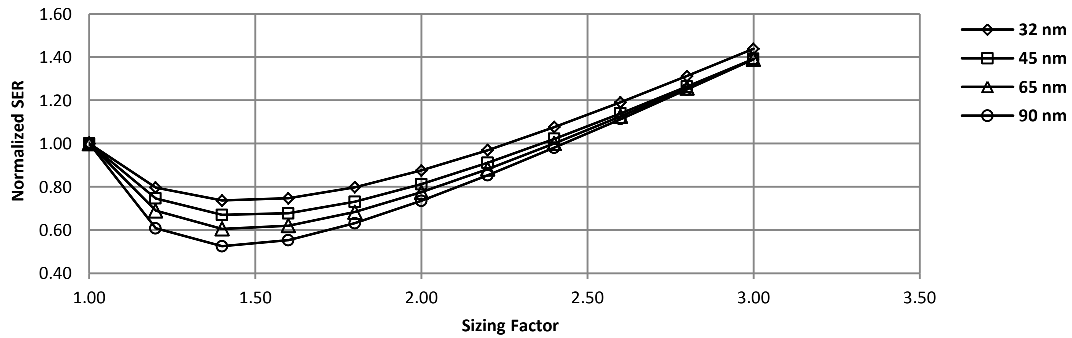


Figure 4.4: Normalized SER of the circuit c432 as a function of sizing factor for parallel transistor networks

to be very low. Hence, it is reasonable to state that SERs of the parallel transistor networks are insignificant, compared to the serial transistor networks.

The second statement can be validated by Figure 4.2 and the previous discussion. As a result, we enhance SE vulnerability of serial transistors by upsizing parallel transistors of each gate. We believe that this approach can effectively deal with the degradation in SE immunity as experienced in upsizing all devices.

Unfortunately, when very large size is assigned to parallel transistors, the particle strike rate, which is area dependent, increases significantly. Even though the oversized parallel transistor network can bring the SER of the serial transistors down to zero, the SER of parallel transistors may increase as a result of the increase in particle strike rate due to superfluous active area. This effect can be seen in Figure 4.4 which shows the relationship between the normalized SER of the circuit c432 and the sizing factor when all parallel transistors are upsized. We notice from this figure that unlike upsizing all transistors, upsizing parallel transistors performs better for small area overhead but causes the circuit SER to increase for larger area overhead.

It is evident from the above results that if the area budget required to improve the SER is not very high (typically no more than 10% area overhead) then instead of upsizing all transistors, upsizing transistors in the parallel network is a superior choice. However, this technique requires

insightful algorithms for limiting the effect of oversizing seen in Figure 4.4. We propose two sizing techniques based on upsizing transistors in parallel networks, or parallel transistors, in the following two sections.

4.3 Nonlinear Optimization Solution for Sizing

The optimal size of parallel transistors can be found using a nonlinear optimization formulation for SER minimization. Parts of the results from the use of this approach were initially reported in our work presented in [33]. In section 4.5, a complete set of results for SER reduction and CPU runtime are provided for various benchmark circuits and technology nodes.

In the nonlinear optimization SER formulation, sizing factor of each transistor is defined as a variable and the area overhead is considered as a constraint. First of all, the definitions of all notations are given and some important notations are explained using the example circuit consisting of five NAND gates as shown in Figure 4.5.

- n_i – number of input pins of a gate i.
- $i(t, j)$ denotes a transistor t in a gate i and j is the input pattern for this gate. For a gate with n_i inputs, the transistor index t varies from 1 to $2n_i$ and the input index j varies from 0 to $2^{n_i} - 1$
- tp_p and ts_q are parallel transistor (tp) and serial transistor (ts). The indices, p and q of each CMOS transistor in a gate i vary from 1 to n_i.
- xp_i – upsizing factor of the parallel transistor network of a gate i.
- $Q_{crit0\ i(t,j)}$ – initial Q_{crit} value of a transistor t in a gate i for a gate input j.
- k – number of generated input vectors.
- $A_{p0\ i}$ – original area of parallel transistors of a gate i

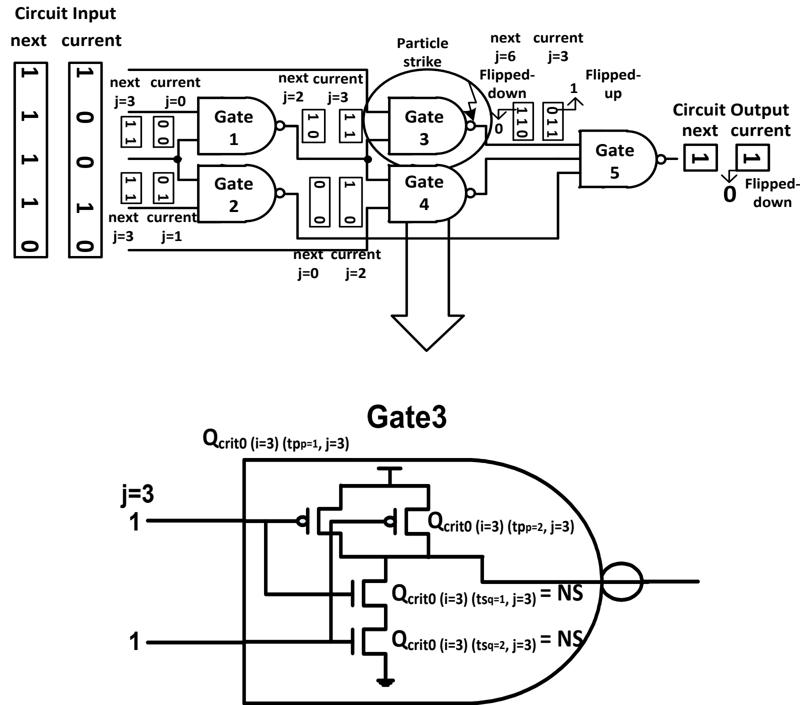


Figure 4.5: Example Circuit

- $AD_{p0\,i}$ and $AD_{s0\,i}$ – original drain area of parallel serial transistor networks of a gate i, respectively.
- $A_{circuit0}$ – original circuit area
- $A_{circuit}$ – circuit area after upsizing.

Figure 4.5 shows a circuit containing four 2-input and a 3-input NAND gates, and a transistor network in Gate3. We apply two input vectors, the current and next input vectors to the circuit. Let us consider that there is an upset at the output of Gate3 for each circuit input. At the current circuit input “10010”, the current input index for Gate3 is $j = 3$ (input value of “11”). If a strike at one of the PMOS transistors in Gate3 causes an up transition on Gate3’s output, the SET can propagate to the circuit output. In this circumstance, *the error count* of Gate3 corresponding to its current input, $E_{(i=3)\,(j=3)}$, is incremented. For the next circuit input vector shown in Figure 4.5,

the input of Gate3 is j=2. The error count, $E_{(i=3)\ (j=2)}$ is not incremented because the down transition on the gate output is logically masked by Gate5.

We also need to consider the Q_{crit} values of all transistors in all gates corresponding to all possible gate inputs. For example, for Gate3 at j=3, in Table 3.1 in Chapter 3, the parallel PMOS transistors in a two-input NAND gate are sensitive at the input "11". As a result, to calculate the neutron flux, we consider two values of initial Q_{crit}, $Q_{crit0\ (i=3)(tp_{p}=1,\ j=3)}$ and $Q_{crit0\ (i=3)(tp_{p}=2,\ j=3)}$ which are the initial critical charge values of the PMOS transistors, $tp_{p=1}$ and $tp_{p=2}$, in Gate3 for the gate input of "11" (j=3). On the other hand, the notations for the Q_{crit} of serial NMOS transistors in Gate3, which are sensitive at the input "10" (j=2), are expressed as $Q_{crit0\ (i=3)(ts_{q}=1,\ j=2)}$ and $Q_{crit0\ (i=3)(ts_{q}=2,\ j=2)}$.

When the area of the parallel network is increased by a factor xp_i, this does not change the value of Q_{crit} of any of the parallel transistor tp_p but it increases the original Q_{crit} of each serial transistor ts_q to $Q_{crit\ i(ts_q,j)}$. The Q_{crit} of the transistor ts_q after upsizing the parallel network by a factor xp_i is given by the following equation:

$$Q_{crit\ i(ts_q,j)} = f_{i(t,j)}(xp_i) * Q_{crit0\ i(ts_q,j)} \tag{4-2}$$

where $f_{i(t,j)}$ is a scaling factor function which is a nondecreasing function of xp_i. To obtain $f_{i(t,j)}$, we performed SPICE simulation to investigate the relationship between Q_{crit} and sizing factor of the parallel transistor network for each gate type and gate input. The scaling factor was fitted into a first order polynomial (linear) function of xp_i; i.e., $f_{i(t,j)}(xp_i) = m * xp_i + n$, where m and n are the fitting constants.

We also precalculate $\emptyset_{i(t,j)}$, from [58] and rewrite it as an exponential function of Q_{crit} shown in (4-3)

$$\emptyset_{i(t,j)} = a + be^{-Q_{crit\ i(t,j)}/c} \tag{4-3}$$

where a, b, and c are curve fitting constants.

Equation (4-4) below gives the SER of each gate i after the parallel network is upsized by a factor of xp_i. This equation is obtained using (3-3) and (3-4) in Chapter 3.

$$GateSER_i = \frac{1}{k \cdot A_{circuit}} \cdot \{ \sum_{p=1}^{n_i} \sum_{j=1}^{2^{n_i}-1} [E_{i(j)} \cdot \emptyset_{i(tp_p,j)} \cdot {AD_{p0\,i}}^2 \cdot {xp_i}^2] + \sum_{q=1}^{n_i} \sum_{j=1}^{2^{n_i}-1} [E_{i(j)} \cdot \emptyset_{i(ts_q,j)}(xp_i) \cdot {AD_{s0\,i}}^2] \} \quad (4\text{-}4)$$

Equation (4-4) is explicitly written as an addition of two summations: sum of SER of the parallel transistors (the first term) and sum of SER of the serial transistors (the second term). The variable xp_i scales the active area $AD_{p0\,i}$ in the first term and it impacts the strike rate $\emptyset_{i(ts_q,j)}$ in the second term of (4-4). We would like to point out that in (4-4), the first term is a function of the square of xp_i and the second term consists of more complicated function of xp_i as expressed in (4-2) and (4-3). Thus, $GateSER_i$ and the sizing factor of parallel transistor network, xp_i, have a nonlinear relation.

In (4-4), $A_{circuit}$ is the total circuit area after optimization. $A_{circuit}$ can be written as given in (4-5).

$$A_{circuit} = A_{circuit0} + \sum_{i=1}^{total\ \#\ of\ gates} (A_{p0\,i} \cdot (xp_i - 1)) \quad (4\text{-}5)$$

The SER of the circuit can be obtained by summing the SER of each gate i. Equations (4-4) and (4-5) above can be formulated as a nonlinear optimization problem over a set of xp_i variables as expressed in (4-6), (4-7), and (4-8).

$$min \sum_{i=1}^{total\ \#\ of\ gates} GateSER_i \quad (4\text{-}6)$$

$$Subject\ to: \sum_{i=1}^{total\ \#\ of\ gates} A_{p0\,i} \cdot (xp_i - 1) \leq overhead \cdot A_{circuit0} \quad (4\text{-}7)$$

$$and \qquad xp_i \geq 1 \quad (4\text{-}8)$$

The objective of the problem is defined as a minimization of the circuit SER as shown in (4-6), subject to two constraints in (4-7) and (4-8). Since we consider only the parallel networks as candidates for upsizing, the additional area in (4-7) is related to the increase in the size of parallel

transistors only. Furthermore, the inequality shown in (4-8) is to ensure that no transistor network or transistor is downsized.

4.4 Heuristic Driven Device Size Management

In this section, we introduce a heuristic based sizing for SER reduction. Parts of this proposed work were published in [30], [31], [33]. This approach employs two algorithms. The first algorithm supports the “fairness” of area distribution which is based on the premise that more vulnerable gates deserve more additional area. The second algorithm prevents the increase in gate SER that may result when the parallel transistor network of a gate is over upsized as mentioned in section 4.2. Results of the proposed heuristics using both algorithms are reported in the next section.

4.4.1 Weighted area distribution algorithm

This algorithm takes into consideration the weight of additional area provided to sensitive gates to improve their SE immunity. This weighted area is assigned based on the SER of each sensitive gate. The algorithm starts with calculating the SER of all gates in a target circuit. The SER of each gate is sorted and indexed in descending order. Next, it requires the user to define the most s vulnerable gates selected to be upsized. Let $SER_{0,i}$ be the original SER of a sensitive gate i, and $SER_{0,s}$ be the original SER of the sensitive gate $i = s$. Note that the sensitive gate i has higher SER than the sensitive gate i+1 and as a result, $SER_{0,s}$ is the smallest among all the gates that are selected for upsizing. We give additional area Δa_i to a sensitive gate i to reduce its SER. To be exact, the relationship between Δa_i and the decrease in the SER of gate i, ΔSER_i is quite complicated since it is dependent on workload and circuit topology [30]. However, it is reasonable to assign Δa_i to be proportional to the desired ΔSER_i as follow:

$$\Delta a_i \propto \Delta SER_i \tag{4-9}$$

Now, since the gates with more vulnerability to SE (those gates with lower index *i*) need larger ΔSER_i, the following equation can be set:

$$\Delta SER_i \propto g_i(\frac{SER_{0,i}}{SER_{0,s}}) \tag{4-10}$$

where g_i is an increasing function. The exact function g_i varies from gate to gate and is difficult to identify. However, this study simply assigns each g_i to be identical for all gates in the same circuit, i.e.; the global function g can be set such that $g := g_i$. Furthermore, we also define $g(x)$ $:= x^r$, where *r* is a positive real number. To be more flexible, we allow *r* to vary and we select the best value of *r* that offers the maximum decrease in the SER of each experimental circuit. We can obtain the additional area in (4-9) for sensitive gate *i*, as written in the equation below:

$$\Delta a_i = \alpha * (\frac{SER_{0,i}}{SER_{0,s}})^r \tag{4-11}$$

where α is a global constant of proportionality (the constant for each selected gate is assumed to be the same). Clearly, the summation of each additional area term for sensitive gate *i* in (4-11) is

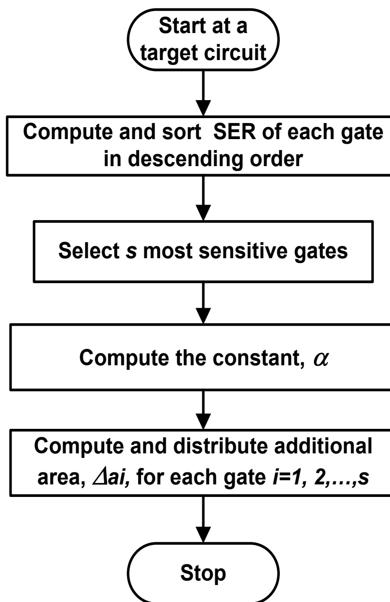


Figure 4.6: Flowchart of weighted area distribution algorithm

equal to the total circuit area overhead. Hence, the constant, α can be determined as follow:

$$\alpha = \frac{desired\ circuit\ area\ overhead}{\sum_{i=1}^{S}(\frac{SER_{0,i}}{SER_{0,S}})^{r}} \tag{4-12}$$

It can be seen from the set of equations above that our weighted area distribution algorithm satisfies the fairness of area distribution as it assigns larger area to more sensitive gates. This algorithm is summarized in the flowchart illustrated in Figure 4.6.

4.4.2 Gate SER saturation consideration algorithm

To prevent the increase in the circuit SER which may occur when we over upsize parallel transistors (as seen in Figure 4.4, section 4.2), it is important to identify the maximum size of each selected gate. What it means is that when a gate is upsized beyond the maximum limit, the gate SER starts increasing. This can be done by separately testing each gate type in the cell library. After the upper bound on the sizing factor of each gate type has been determined, we start upsizing from the most sensitive gate i = 1 through all other sensitive gates in descending order of their SE vulnerability with the following conditions.

a) *If the assigned area (can be obtained from the weighted area distribution algorithm) of gate i is greater than the maximum value, the excess area beyond the allowed limit is stored in an area pool for reassigning to other sensitive gates.*
b) *If the assigned area is within the maximum bound, we give the additional area taken from all or part of the area pool to this gate until it reaches the maximum value.*

The priority that more sensitive gates can take the additional area from the area pool is set to be higher as long as their size does not exceed the precomputed maximum value. The flowchart of this algorithm is illustrated in Figure 4.7. Note that before entering this algorithm, the additional area and the maximum size for each gate have completely been determined.

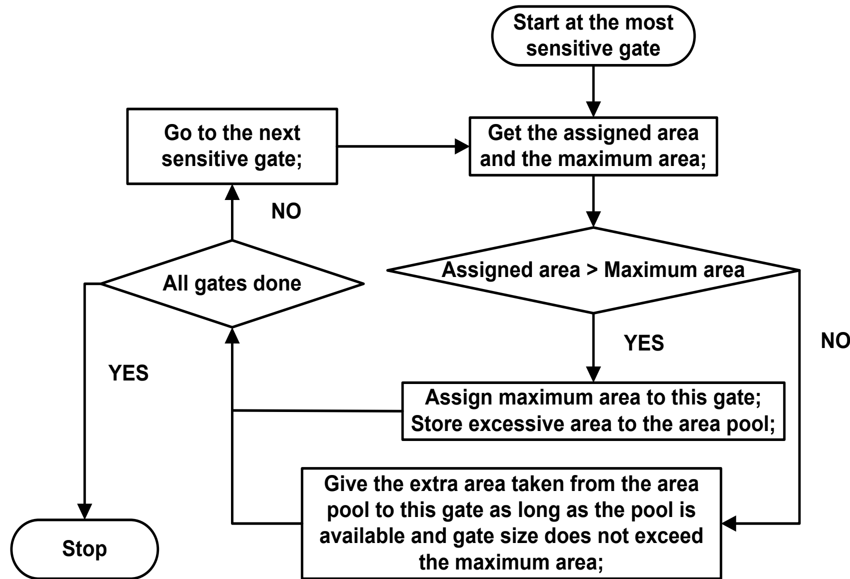


Figure 4.7: Flowchart of gate SER saturation consideration algorithm

4.5 Experimental Results

To evaluate our proposed techniques, various circuits in ISCAS'85, ISCAS'89 (combinational parts), and ITC benchmark suits are selected as the experimental circuits that are mapped with varied technology nodes from predictive technology models in [35]. The circuits operate at a temperature of 25°C. The cell library contains 2-, 3-, and 4-input NAND and NOR gates, and inverters. Using the gate level SE simulator for combinational circuits presented in the previous chapter, all target circuits are applied 100,000 random input patterns. The framework was implemented in an Intel Pentium Core 2 Duo machine with 3-GHz processor and 2-GB RAM running Microsoft Windows Vista. We compute the SER results in normalized form with respect to the base case of circuit layout in which each gate meets the original design objective of equal rise/fall delay and minimum size.

Table 4.1: Information and Normalized SER Results of Benchmark Circuits

Circuit	Circuit Information			SER			
	No. of PI's	*No. of PO's*	*No. of Gates*	*32 nm*	*45 nm*	*65 nm*	*90 nm*
c432	36	7	159	0.85	0.84	0.84	0.83
c499	41	32	578	0.70	0.72	0.74	0.77
c1196	32	31	472	0.84	0.84	0.84	0.84
c1908	33	25	459	0.78	0.80	0.82	0.84
c6288	32	32	2672	0.98	0.98	0.97	0.97
i1	25	13	40	0.86	0.87	0.88	0.90
i2	201	1	148	0.85	0.85	0.84	0.84
i3	132	6	138	0.26	0.25	0.24	0.23
i4	192	6	184	0.89	0.89	0.89	0.89
i5	133	66	288	0.88	0.89	0.90	0.90
i6	138	67	340	1	1	1	1
i7	199	67	512	0.62	0.62	0.62	0.62
i8	133	81	1685	0.54	0.54	0.53	0.53
s13207	700	790	9577	0.95	0.95	0.95	0.95
s15850	611	684	12101	0.94	0.94	0.94	0.94

4.5.1 Gate input reconfiguration technique

We conducted a set of experiments to investigate the effect of technology node variation on the yield of the gate input reconfiguration technique. Table 4.1 shows the circuit size information (number of primary inputs, outputs, and gates in the circuit) and results of normalized values of the *CircuitSER* of the experimental benchmark circuits mapped with 32, 45, 65, and 90 nm predictive technology nodes. It is evident from this table that this area overhead-free technique is effective in reducing the circuit SER. We notice that for the circuit i3, the normalized circuit SER reduces by 77% whereas for the circuit i6, all original gate input pins surprisingly meet the best configuration for SE tolerance; hence, this technique offers no benefit. Moreover, the gains from the gate input reconfiguration technique are almost identical regardless of technology variation.

4.5.2 Nonlinear optimization solution for sizing

Before using optimization based sizing method, each selected benchmark circuit was initialized

Table 4.2: Optimal Results for 32, 45, 65, and 90 nm Benchmark Circuits with Area Overhead of 2% and 5%

Circuit	**2% Area Overhead**								**5% Area Overhead**							
	32 nm		*45 nm*		*65 nm*		*90 nm*		*32 nm*		*45 nm*		*65 nm*		*90 nm*	
	SER	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*
c432	**0.9091**	101	**0.8698**	78	**0.8854**	86	**0.8641**	78	**0.8450**	110	**0.8288**	111	**0.8009**	112	**0.7596**	103
c1196	**0.7723**	805	**0.7558**	795	**0.7323**	975	**0.7129**	756	**0.6819**	828	**0.6560**	828	**0.6307**	824	**0.6021**	822
c1908	**0.8274**	889	**0.8336**	878	**0.8258**	722	**0.8115**	734	**0.7532**	910	**0.7417**	952	**0.7181**	913	**0.6842**	911
i3	**0.6874**	19	**0.6283**	21	**0.5860**	19	**0.5547**	19	**0.5289**	26	**0.4816**	29	**0.4494**	32	**0.4193**	36
i4	**0.8012**	82	**0.7693**	65	**0.7339**	82	**0.7068**	92	**0.6823**	188	**0.6493**	182	**0.6202**	222	**0.5944**	192

by the gate input reconfiguration technique, and its improved SER was computed. Next, we employed MATLAB nonlinear optimization toolbox to solve the problem [44]. All SER results are reported in the normalized form with respect to the SER of experimental circuits after applying the gate input reconfiguration technique. In addition, all CPU runtime results exclude the time for collecting the error count which was precomputed using our simulator for combinational circuits described in Chapter 3 and 100,000 random input vectors for each circuit.

The solutions from the nonlinear optimization formulation given in equations (4-6), (4-7), and (4-8) are shown in Table 4.2. It is evident that this method offers substantial improvement in SER for all the circuits for both 2% and 5% overhead. This technique can offer up to 60% SER reduction in some cases. However, the Table 4.2 also shows the runtime of the program for the candidate benchmark circuits which is in 100s of seconds. All these circuits are relatively small (have number of gates less than one thousand) and for the large circuits, this approach requires even larger CPU runtime. For example, for the circuit i8 which has 1685 gates and is not shown in the above table, the required runtime was over an hour to complete the task while for other larger circuits, the program did not complete the task and was aborted. To reduce CPU time while searching for nonlinear optimization solutions, we may either use an approximate linear optimization formulation or reduce the number of variables in the nonlinear optimization formulation. However, the results of approximate linear optimization from [22] may be expected

to lack accuracy. The number of variables in the nonlinear formulation can be reduced by selecting s most vulnerable gates and setting $xp_i = 1$ for the remaining gates less sensitive to SEs. Nonetheless, since the value of s varies from circuit to circuit, it is difficult to determine a good value of s.

We conclude that this approach is prohibitive for larger circuits consisting of thousands of gates. Therefore, we further investigated the proposed heuristic based upsizing methodology which can satisfactorily reduce circuit SER in substantially smaller computation time.

4.5.3 Heuristic based sizing

In this experiment, the gate input reconfiguration technique was also initially applied to all experimental circuits. We extended the investigation domain to include larger circuits than those considered in the previous experiment. In addition, we analyze five different sizing methods under area overhead and technology node variations. Evaluated by our JAVA based simulator, the results from those methods are compared to justify their performance.

Table 4.3, Table 4.4, Table 4.5, and Table 4.6 report the results of the normalized SER and the CPU runtime from different heuristic based upsizing methods applied to selected benchmark circuits mapped with 32, 45, 65, and 90 nm technology nodes, respectively. Method I is the simplest method which upsizes all devices with equal sizing factor. Method II, Method III, Method IV, and Method V differently upsize parallel transistors of the target circuit. Both Method II and Method III upsize all parallel transistors. However, Method II operates without weighted area and SER saturation consideration, whereas Method III includes our two proposed algorithms. Method IV and Method V initially select only top 10% of all gates to be upsized based on the gate SER. Method IV operates without our two algorithms, whereas Method V uses our two algorithms. For Method III and Method V which include both algorithms, the weighted

area distribution algorithm is applied first to assign the additional area to all (Method III) or part (Method V) of the gates in each experimental circuit with appropriate *r* values as in (4-11), then the gate SER saturation consideration algorithm is used to prevent the effect of oversizing.

Table 4.3: Heuristics Results for 32 nm Benchmark Circuits with Area Overhead of 2% and 5%

(A) 2% AREA OVERHEAD

Circuit	Method I		Method II		Method III			Method IV		Method V		
	SER	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*
c432	1.0112	0.010	0.9691	0.006	0.9424	4.17	1.9	0.9540	0.004	0.9413	4.12	1.9
c499	1.0122	0.013	0.9644	0.012	0.9026	13.89	1.9	0.9530	0.006	0.8936	14.94	1.9
c1196	1.0110	0.006	0.9633	0.006	0.8113	9.77	1.8	0.8949	0.003	0.8021	9.92	1.5
c1908	1.0115	0.007	0.9593	0.006	0.8563	10.97	1.9	0.8810	0.003	0.8493	11.65	1.6
c6288	1.0128	0.026	0.9585	0.0238	0.9529	38.40	1.9	0.9522	0.012	0.9507	38.33	1.9
i1	1.0110	0.0005	0.9483	0.0005	0.8778	1.09	1.9	0.8697	0.002	0.8689	1.12	1.9
i2	1.0109	0.005	0.8907	0.001	0.6147	6.00	1.5	0.6619	0.002	0.6153	6.18	0.1
i3	1.0099	0.002	0.9289	0.002	0.7083	3.49	1.9	0.8123	0.002	0.7055	3.53	1.9
i4	1.0106	0.004	0.9577	0.003	0.8386	6.49	1.9	0.8722	0.002	0.8258	6.64	1.9
i5	1.0118	0.004	0.9832	0.004	0.9699	8.58	1.9	0.9792	0.002	0.9694	8.74	1.4
i6	1.0108	0.010	0.9731	0.009	0.9662	23.07	1.9	0.9890	0.005	0.9687	24.70	1.9
i7	1.0102	0.013	0.9426	0.011	0.8750	24.57	1.9	0.8578	0.007	0.8477	24.78	1.9
i8	1.0105	0.057	0.9512	0.042	0.8843	88.30	1.9	0.9232	0.024	0.8910	90.19	1.7
s13207	1.0120	0.106	1.0033	0.064	0.9262	99.58	1.9	0.9674	0.035	0.9167	96.15	1.6
s15850	1.0121	0.118	1.0036	0.103	0.9453	109.16	1.9	0.9731	0.041	0.9363	112.83	1.9
Ave.	1.0112	-	0.9598	-	0.8715	-	-	0.9027	-	0.8655	-	-

(B) 5% AREA OVERHEAD

Circuit	Method I		Method II		Method III			Method IV		Method V		
	SER	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*
c432	1.0280	0.007	0.9313	0.006	0.8757	3.81	1.9	0.9213	0.005	0.8800	3.82	1.9
c499	1.0304	0.011	0.9210	0.016	0.8332	13.60	1.9	0.9633	0.006	0.8446	13.25	1.3
c1196	1.0273	0.006	0.9191	0.005	0.7201	9.41	1.8	0.8465	0.003	0.7063	9.45	1
c1908	1.0285	0.006	0.9108	0.006	0.7841	10.43	1.6	0.7952	0.003	0.7792	10.35	1.1
c6288	1.0317	0.022	0.9060	0.019	0.8933	35.42	1.9	0.9012	0.012	0.8994	35.05	1.9
i1	1.0272	0.0006	0.8839	0.0006	0.7510	1.04	1.9	0.7481	0.0004	0.7473	1.05	0.9
i2	1.0270	0.003	0.7997	0.001	0.5678	5.67	1	0.7543	0.001	0.5682	5.58	0.1
i3	1.0244	0.002	0.8403	0.002	0.5453	3.28	1.9	0.6373	0.002	0.5436	3.24	1.9
i4	1.0263	0.004	0.9054	0.004	0.7148	6.25	1.9	0.7875	0.002	0.7066	6.09	1.7
i5	1.0295	0.004	0.9640	0.004	0.9375	8.27	1.9	0.9741	0.002	0.9373	8.16	0.1
i6	1.0269	0.010	0.9402	0.009	0.9229	21.14	1.9	0.9972	0.005	0.9292	21.08	0.8
i7	1.0252	0.012	0.8769	0.011	0.7646	24.93	1.9	0.7514	0.006	0.7385	23.99	1.9
i8	1.0261	0.043	0.8931	0.045	0.7950	85.82	1.6	0.8885	0.023	0.8231	84.31	0.9
s13207	1.0298	0.070	1.0147	0.055	0.8728	98.99	1.9	0.9668	0.034	0.8643	94.11	1.9
s15850	1.0300	0.126	1.0150	0.103	0.8979	109.52	1.9	0.9731	0.040	0.8934	103.60	1.9
Ave.	1.0279	-	0.9148	-	0.7917	-	-	0.8604	-	0.7907	-	-

Table 4.4: Heuristics Results for 45 nm Benchmark Circuits with Area Overhead of 2% and 5%

(A) 2% AREA OVERHEAD

Circuit	Method I		Method II		Method III			Method IV		Method V		
	SER	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*
c432	1.0092	0.006	0.9623	0.005	0.9302	3.97	1.9	0.9370	0.005	0.9257	3.79	1.9
c499	1.0105	0.011	0.9571	0.011	0.8919	13.22	1.9	0.9405	0.006	0.8866	13.82	1.9
c1196	1.0086	0.006	0.9538	0.005	0.7847	9.27	1.7	0.8591	0.003	0.7690	9.53	1.3
c1908	1.0094	0.006	0.9508	0.005	0.8503	10.44	1.9	0.8591	0.003	0.8408	10.38	1.4
c6288	1.0109	0.022	0.9507	0.019	0.9426	35.54	1.9	0.9384	0.012	0.9366	35.16	1.9
i1	1.0079	0.0006	0.9330	0.0006	0.8385	1.04	1.9	0.8273	0.0004	0.8263	1.04	1.9
i2	1.0077	0.003	0.8413	0.001	0.3995	5.56	1.9	0.5373	0.001	0.3970	5.63	0.4
i3	1.0091	0.002	0.9075	0.002	0.6444	3.24	1.9	0.7536	0.001	0.6413	3.27	1.9
i4	1.0079	0.004	0.9466	0.004	0.7958	6.11	1.9	0.8273	0.002	0.7802	61.14	1.9
i5	1.0100	0.004	0.9787	0.004	0.9628	8.29	1.9	0.9694	0.003	0.9601	8.22	1.9
i6	1.0086	0.010	0.9672	0.009	0.9592	20.93	1.9	0.9864	0.005	0.9645	20.99	1.9
i7	1.0079	0.012	0.9290	0.010	0.8546	23.94	1.9	0.8280	5.91	0.8190	24.26	1.9
i8	1.0080	0.049	0.9385	39.40	0.8626	83.89	1.9	0.9042	0.020	0.8741	83.38	1.5
s13207	1.0102	0.068	0.9994	0.056	0.9149	95.18	1.9	0.9523	0.035	0.9017	93.94	1.7
s15850	1.0104	0.105	1.0004	0.093	0.9349	105.42	1.9	0.9606	0.042	0.9223	103.57	1.9
Ave.	1.0091	-	0.9478	-	0.8378	-	-	0.8720	-	0.8297	-	-

(B) 5% AREA OVERHEAD

Circuit	Method I		Method II		Method III			Method IV		Method V		
	SER	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*
c432	1.0226	0.007	0.9156	0.006	0.8520	3.84	1.9	0.8870	0.005	0.8511	3.76	1.9
c499	1.0258	0.011	0.9041	0.011	0.8199	13.24	1.8	0.9507	0.005	0.8381	13.22	1.1
c1196	1.0211	0.006	0.8985	0.005	0.6825	9.56	1.5	0.7993	0.003	0.6627	9.33	0.9
c1908	1.0233	0.006	0.8922	0.005	0.7667	10.41	1.6	0.7698	0.003	0.7628	10.34	0.9
c6288	1.0268	0.022	0.8871	0.019	0.8690	36.03	1.9	0.8776	0.012	0.8756	34.95	1.9
i1	1.0193	0.0006	0.8507	0.0006	0.6835	1.04	1.9	0.6834	0.0004	0.6823	1.05	0.7
i2	1.0187	0.003	0.7109	0.001	0.3404	5.67	1.0	0.5648	0.001	0.3409	5.56	0.1
i3	1.0147	0.002	0.7958	0.002	0.4947	3.27	1.7	0.5535	0.001	0.4903	3.23	1.3
i4	1.0192	0.004	0.8809	0.004	0.6634	6.11	1.9	0.7281	0.002	0.6548	6.09	1.8
i5	1.0247	0.004	0.9533	0.004	0.9246	8.26	1.8	0.9513	0.003	0.9246	8.28	0.1
i6	1.0212	0.010	0.9264	0.009	0.9074	21.02	1.9	0.9900	0.005	0.9203	21.01	0.1
i7	1.0192	0.012	0.8484	0.011	0.7359	24.06	1.9	0.7221	0.006	0.7128	23.80	1.9
i8	1.0197	0.046	0.8655	0.035	0.7595	83.37	1.5	0.8656	0.020	0.8036	83.88	0.7
s13207	1.0251	0.068	1.0058	0.056	0.8571	94.63	1.9	0.9399	0.035	0.8463	94.02	1.5
s15850	1.0256	0.109	1.0076	0.096	0.8802	105.98	1.9	0.9487	0.042	0.8755	103.56	1.7
Ave.	1.0218	-	0.8895	-	0.7491	-	-	0.8155	-	0.7494	-	-

Table 4.5: Heuristics Results for 65 nm Benchmark Circuits with Area Overhead of 2% and 5%

(A) 2% AREA OVERHEAD

Circuit	Method I		Method II		Method III			Method IV		Method V		
	SER	*Run time (s)*	*SER*	*Run time (ms)*	*SER*	*Run time (s)*	*r value*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*
c432	1.0054	0.007	0.9522	0.006	0.9163	3.76	1.9	0.9164	0.006	0.9126	3.83	0.9
c499	1.0070	0.012	0.9458	0.012	0.8858	14.02	1.9	0.9313	0.005	0.8883	13.54	1.9
c1196	1.0044	0.006	0.9412	0.006	0.7677	9.46	1.6	0.8250	0.003	0.7472	9.44	1.2
c1908	1.0058	0.006	0.9394	0.005	0.8446	10.70	1.9	0.8427	0.003	0.8338	10.38	1.2
c6288	1.0073	0.022	0.9370	0.020	0.9263	36.10	1.9	0.9220	0.012	0.9197	35.28	1.9
i1	1.0033	0.0006	0.9141	0.0005	0.7962	1.06	1.9	0.7835	0.0004	0.7820	1.07	1.5
i2	1.0027	0.004	0.7877	0.002	0.3259	5.62	1.8	0.4648	0.001	0.3230	5.66	0.6
i3	1.0010	0.002	0.8829	0.002	0.5976	3.28	1.9	0.6976	0.001	0.5944	3.25	1.9
i4	1.0034	0.003	0.9321	0.003	0.7623	6.77	1.9	0.7864	0.002	0.7452	6.10	1.9
i5	1.0063	0.004	0.9717	0.004	0.9541	8.20	1.9	0.9552	0.002	0.9497	8.18	0.9
i6	1.0046	0.010	0.9585	0.009	0.9546	21.92	1.5	0.9804	0.005	0.9631	20.92	1.9
i7	1.0038	0.014	0.9119	0.012	0.8388	23.97	1.9	0.8114	0.006	0.8020	23.98	1.9
i8	1.0038	0.044	0.9217	0.035	0.8480	83.65	1.8	0.9005	0.020	0.8776	83.84	1.3
s13207	1.0062	0.071	0.9930	0.055	0.9016	94.79	1.9	0.9359	0.035	0.8847	94.08	1.8
s15850	1.0065	0.116	0.9943	0.099	0.9200	105.63	1.9	0.9439	0.044	0.9038	105.08	1.9
Ave.	1.0048	-	0.9322	-	0.8160	-		0.8465	-	0.8085	-	

(B) 5% AREA OVERHEAD

Circuit	Method I		Method II		Method III			Method IV		Method V		
	SER	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*
c432	1.0129	0.008	0.8941	0.007	0.8247	3.77	1.9	0.8511	0.005	0.8315	3.92	1.3
c499	1.0170	0.011	0.8800	0.020	0.8057	13.29	1.7	0.9418	0.007	0.8281	13.27	0.8
c1196	1.0104	0.006	0.8727	0.005	0.6592	9.51	1.4	0.7647	0.003	0.6403	9.56	0.8
c1908	1.0141	0.006	0.8692	0.005	0.7461	10.40	1.7	0.7580	0.003	0.7538	10.41	0.8
c6288	1.0176	0.021	0.8565	0.020	0.8338	35.36	1.9	0.8551	0.012	0.8526	35.24	1.8
i1	1.0078	0.0006	0.8127	0.0005	0.6229	1.04	1.9	0.6318	0.0004	0.6303	1.04	0.5
i2	1.0062	0.003	0.6286	0.001	0.2600	5.59	0.9	0.5392	0.001	0.2616	5.59	0.1
i3	1.0020	0.002	0.7485	0.002	0.4664	3.25	1.3	0.4908	0.002	0.4549	3.25	0.9
i4	1.0079	0.003	0.8509	0.003	0.6302	6.17	1.9	0.6893	0.002	0.6280	6.10	1.3
i5	1.0152	0.004	0.9379	0.004	0.9065	8.27	1.9	0.9232	0.002	0.9055	8.19	0.1
i6	1.0109	0.010	0.9075	0.009	0.8933	21.00	1.9	0.9793	0.005	0.9153	21.04	0.1
i7	1.0090	0.012	0.8158	0.010	0.7215	24.04	1.9	0.7187	0.006	0.7098	23.87	1.9
i8	1.0090	0.043	0.8317	0.035	0.7292	85.16	1.4	0.8756	0.024	0.7950	83.99	0.1
s13207	1.0150	0.076	0.9925	0.058	0.8375	94.85	1.9	0.9175	0.035	0.8282	94.13	1.3
s15850	1.0156	0.115	0.9946	0.098	0.8587	105.81	1.9	0.9226	0.040	0.8557	103.92	1.8
Ave.	1.0114	-	0.8595	-	0.7197	-	-	0.7906	-	0.7260	-	-

Table 4.6: Heuristics Results for 90 nm Benchmark Circuits with Area Overhead of 2% and 5%

(A) 2% AREA OVERHEAD

Circuit	Method I		Method II		Method III			Method IV		Method V		
	SER	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*
c432	0.9987	0.007	0.9355	0.006	0.8864	3.91	1.9	0.8875	0.005	0.8822	4.14	0.8
c499	1.0009	0.011	0.9275	0.011	0.8711	13.43	1.9	0.9218	0.006	0.8825	13.98	1.7
c1196	0.9973	0.006	0.9223	0.006	0.7430	9.46	1.6	0.7873	0.003	0.7194	9.37	1.0
c1908	0.9997	0.006	0.9228	0.005	0.8349	10.47	1.9	0.8298	0.003	0.8235	10.47	1.1
c6288	1.0008	0.023	0.9126	0.020	0.8979	35.48	1.9	0.8982	0.012	0.8948	35.19	1.8
i1	0.9959	0.0006	0.8867	0.0006	0.7426	1.04	1.9	0.7306	0.0004	0.7284	1.12	1.1
i2	0.9945	0.003	0.7120	0.001	0.2803	5.57	1.1	0.4124	0.001	0.2848	5.61	1.0
i3	0.9931	0.002	0.8497	0.002	0.5594	3.27	1.9	0.6364	0.001	0.5564	3.24	1.9
i4	0.9960	0.003	0.9104	0.003	0.7332	6.22	1.9	0.7429	0.002	0.7157	6.10	1.9
i5	0.9997	0.004	0.9594	0.004	0.9341	8.21	1.9	0.9317	0.002	0.9273	8.19	1.9
i6	0.9977	0.010	0.9439	0.009	0.9403	21.35	1.3	0.9678	0.005	0.9638	21.03	0.1
i7	0.9972	0.013	0.8880	0.011	0.8257	24.76	1.9	0.7974	0.006	0.7910	23.91	1.9
i8	0.9967	0.044	0.8966	0.035	0.8247	83.80	1.6	0.8761	0.020	0.8674	84.14	1.1
s13207	0.9990	0.072	0.9816	0.056	0.8809	94.56	1.9	0.9055	0.037	0.8609	93.76	1.8
s15850	0.9993	0.117	0.9829	0.100	0.8963	105.65	1.9	0.9119	0.044	0.8787	104.35	1.9
Ave.	0.9978	-	0.9088	-	0.7901	-	-	0.8158	-	0.7851	-	-

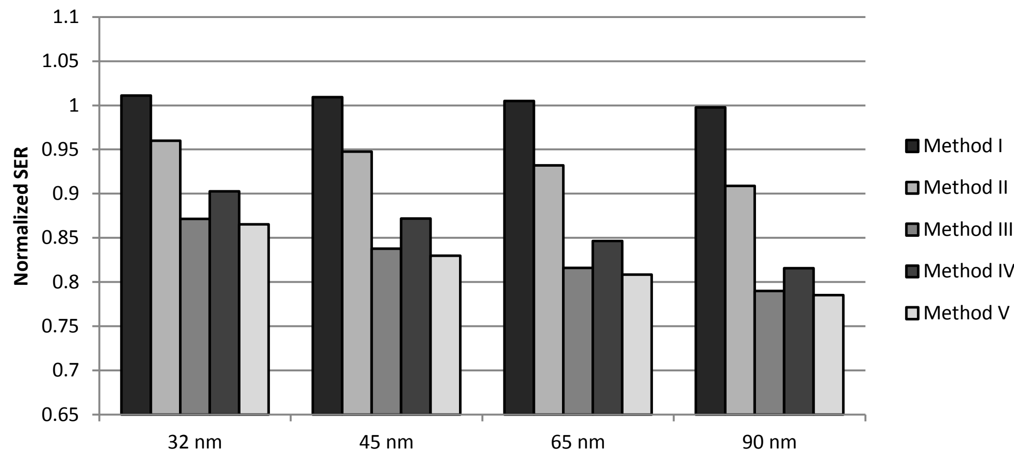
(B) 5% AREA OVERHEAD

Circuit	Method I		Method II		Method III			Method IV		Method V		
	SER	*Run time (s)*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r valus*	*SER*	*Run time (s)*	*SER*	*Run time (s)*	*r value*
c432	0.9961	0.007	0.8592	0.007	0.7809	3.78	1.7	0.8074	0.006	0.7901	3.79	1.0
c499	1.0016	0.011	0.8423	0.012	0.7690	14.35	1.6	0.9267	0.006	0.7987	13.52	0.4
c1196	0.9927	0.006	0.8360	0.006	0.6261	9.44	1.3	0.7342	0.003	0.6250	9.57	0.7
c1908	0.9987	0.006	0.8362	0.005	0.7130	10.44	1.9	0.7513	0.003	0.7433	10.42	0.1
c6288	1.0012	0.023	0.8045	0.019	0.7755	36.18	1.9	0.8306	0.012	0.8272	35.40	1.4
i1	0.9893	0.0006	0.7606	0.0005	0.5595	1.05	1.9	0.5859	0.0004	0.5837	1.14	0.3
i2	0.9857	0.004	0.5259	0.001	0.1575	5.62	1.9	0.5786	0.001	0.1596	5.62	1.9
i3	0.9825	0.002	0.6867	0.002	0.4376	3.29	1.1	0.4373	0.001	0.4194	3.26	0.7
i4	0.9894	0.003	0.8086	0.003	0.6056	6.17	1.7	0.6597	0.002	0.6174	6.14	0.8
i5	0.9987	0.004	0.9114	0.004	0.8695	8.28	1.9	0.8812	0.002	0.8668	8.22	0.2
i6	0.9936	0.010	0.8764	0.009	0.8683	21.33	1.2	0.9591	0.005	0.9091	21.15	0.1
i7	0.9926	0.013	0.7724	0.011	0.7038	25.31	1.9	0.7200	0.006	0.7144	25.19	0.1
i8	0.9912	0.046	0.7840	0.035	0.6914	90.20	1.3	0.8460	0.020	0.7842	88.65	0.1
s13207	0.9969	0.072	0.9691	0.057	0.8138	95.27	1.9	0.8670	0.036	0.8070	94.62	1.0
s15850	0.9976	0.122	0.9708	0.104	0.8305	106.77	1.9	0.8653	0.044	0.8284	104.23	0.9
Ave.	0.9939	-	0.8163	-	0.6801	-	-	0.7634	-	0.6983	-	-

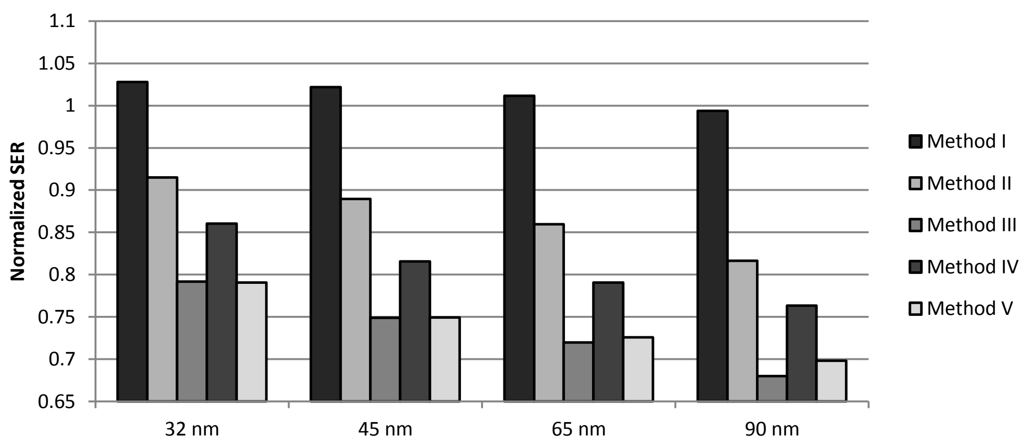
We draw the following conclusions from the experimental results shown in Table 4.3, Table 4.4, Table 4.5, and Table 4.6. On average, Method I increases the SER of the experimental circuits which are mapped to below 90 nm technology. Method II (all circuits except s13207 and s15850) and Method IV yield the marginal reduction in circuit SER but Method IV performs

better for almost all of the circuits. It is clearly seen from the tables that Method III and Method V, which employ our two proposed algorithms, on average, offer the largest reduction in SER.

Figure 4.8 shows the comparison of the average SER of all experimental circuits receiving the area overhead of 2% and 5% (the last row of each of Tables 4.3, 4.4, 4.5, and 4.6). It is evident from these results that the proposed heuristics, especially, Method III and Method IV can arrive at near optimal solution obtained in the previous experiment for almost all of the small circuits.



(a) Area overhead of 2%



(b) Area overhead of 5%

Figure 4.8: Average SER of all experimental circuits with the area overhead of 2% (a) and 5% (b)

We believe that the yields for larger circuits are also equivalently effective although the results for large circuits with optimization based techniques cannot be verified. A remarkable efficiency of the heuristic or fault sensitivity based sizing technique is that for very large circuits, the CPU runtime are incomparably smaller than the time used for searching nonlinear optimization solutions which may have taken several hours or days.

The weighted area distribution algorithm as discussed in section 4.4.1 is one of the most important factors which affects the performance of its application. In this study, the relationship between the increase in gate area and the decrease in gate SER as given in (4-10) and (4-11) is approximately defined as the global function (function of the power of r, positive real number) for all gates in a circuit. This relationship actually varies from gate to gate depending on gate input, gate type, circuit topology, and technology node. Therefore, it is not easy to obtain the exact function for each sensitive gate. In addition, the number of sensitive gates selected to be upsized (s) can cause variations in SER reduction gains. In this study, we compare only two selection methods in which 100% of gates are selected (Method III) and 10% top sensitive gates are selected (Method V) for upsizing and thus, the effect of s on SE resilience cannot be seen clearly. As reported in [33], the number of selected gates, s impacts the results from both weighted area distribution and gate SER saturation consideration algorithms. As a result, this influences the outcome of use of our two algorithms.

Chapter 5. Combined Leakage and Soft Error Optimization for Combinational Circuits

In this Chapter, we first provide an integrated reliability model which combines power/temperature induced hard errors and neutron-induced SEs. The reliability metric, MTTF, is used since it can help model both effects into one integrative term. It will then be possible to optimize the MTTF to obtain the best BB configuration providing maximum overall reliability. After that, we introduce an optimization based approach to maximize combined circuit MTTF due to leakage and SER without performance penalty. This is achieved by setting the delay constraint identical to the critical delay of the original circuit. Next, we discuss a heuristic driven technique which is highly efficient in providing a good BB configuration for reliability improvement. This approach requires relatively small CPU runtime compared to the optimization method, yet provides near optimal BB values for most circuits. Finally, the experimental results and discussion are given.

5.1 Leakage and Soft Error Interrelation

From device level simulation, we found that the BB can alter the circuit SER. In particular, the Q_{crit} of a gate decreases when RBB is applied but increases in the presence of FBB. The amount of Q_{crit} relates to the strike rate of neutron which can generate a SET in a gate as discussed before. Thus, RBB causes SER to increase while FBB causes SER to decrease. The evidence of the above statement is seen in Figure 5.1 that plots the leakage current and SER as a function of BB for the circuit c17 mapped with 32-nm technology nodes from [56]. It can be seen from Figure 5.1 that when BB voltage of all gates is in RBB region (negative values of BB voltage),

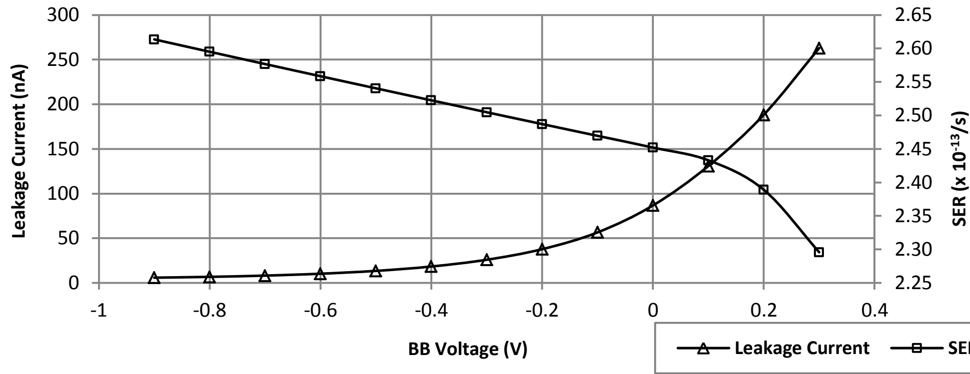


Figure 5.1: Leakage current and SER vs. BB voltage plots for the circuit c17

leakage current decreases exponentially whereas SER increases. On the other hand, in FBB region (positive values of BB voltage), the circuit c17 has a sharp increase in leakage current but its SER decreases substantially. This motivated us to study the joint effect of these two on the circuit reliability. The goal is to find a method that compromises the opposite outcomes of BB on leakage and SER.

5.2 Defining a Common Metric

This section introduces a common metric, MTTF which is used to integrate the effects of SEs and leakage. We first provide a definition of MTTF due to SEs. Then we discuss the thermal effect of elevated leakage current and the other MTTF term which reflects the life time reliability degradation. Finally, we combine both MTTFs into the total circuit MTTF.

The MTTF due to SE ($MTTF_{SE}$) in (5-1) is defined as the reciprocal of the $Circuit\ SER$ in (3-5). The $MTTF_{SE}$ determines the average time for a SE to appear on the output of the circuit.

$$MTTF_{SE} = \frac{1}{Circuit\ SER} \tag{5-1}$$

From SPICE simulation, we can obtain the leakage current of a gate i, $I_{leak,i}$ as a function of BB voltage of this gate, $V_{BB,i}$. During the simulation, we separately test all the gates in the library

with the input such that each gate has the worst case leakage. The relationship between $I_{leak,i}$ and $V_{BB,i}$ can be fitted into an exponential equation below.

$$I_{leak,i} = \alpha + \beta e^{\gamma V_{BB,i}} \quad (5\text{-}2)$$

In (5-2) α, β, and γ are positive constants. For FBB, $V_{BB,i}$ has a positive value whereas for RBB, $V_{BB,i}$ is negative. From (5-2), we can further obtain the total leakage current in a silicon cell in the circuit by summing $I_{leak,i}$ for all gates. Since the leakage power of each gate i is the product of $I_{leak,i}$ and supply voltage, $V_{DD,i}$, the leakage power consumed by a cell, $P_{leak,cell}$ can be achieved by summing the leakage power of each gate as given in (5-3).

$$P_{leak,cell} = \sum_{i=1}^{total\ \#\ of\ gates} (I_{leak,i} * V_{DD,i}) \quad (5\text{-}3)$$

An increase in leakage power raises the temperature of a circuit block. We can obtain the equilibrium temperature as a result of the change in leakage using a cell level model in [48]. This model estimates a change in temperature of a silicon cell from initial operating temperature, $T_{0,cell}$ to the new temperature, $T_{1,cell}$, when the leakage power consumed by the circuit block changes from $P_{leak0,cell}$ to $P_{leak1,cell}$ as given in (5-4). $R_{\theta s}$ in this equation is the thermal resistance of the substrate which is a function of the material and the dimension of the circuit block.

$$T_{1,cell} - T_{0,cell} = R_{\theta s}(P_{leak1,cell} - P_{leak0,cell}) \quad (5\text{-}4)$$

An increase in temperature of a device deteriorates long term reliability of the circuit. The relation of the changes in temperature and leakage power can be explained using a large thermal cycle model for MTTF given in [47]. Under the condition that only single silicon cell occupies the whole circuit, the change in MTTF due to leakage ($MTTF_{leak}$) of the circuit at new operating temperature, T_1, compared to initial temperature, T_0, can be expressed by the empirical equation shown in (5-5) below.

$$MTTF_{leak} = k\left(\frac{1}{T_1 - T_0}\right)^{2.35} \quad (5\text{-}5)$$

where k is the proportionality constant which is unknown. In order to obtain the exact $MTTF_{leak}$ value, we define w_{ZBB} as the ratio of $MTTF_{leak}$ and $MTTF_{SE}$ at the zero BB (ZBB) and allow this ratio to be varied from 25 to 35 [31], [46]. As a result, we can obtain k by the following equation.

$$k = w_{ZBB} * MTTF_{SE}|_{V_{BB}=0} * \left(T_1|_{V_{BB}=0} - T_0\right)^{2.35} \tag{5-6}$$

We assume that the effects of SE and leakage are completely independent. Therefore, the combined MTTF for the circuit due to these two effects can be obtained using (5-7) below.

$$MTTF_{circuit} = \frac{1}{\frac{1}{MTTF_{SE}} + \frac{1}{MTTF_{leak}}} \tag{5-7}$$

Note that the circuit MTTF, $MTTF_{circuit}$, above is the reciprocal of the summation of failure rates due to SEs and leakage.

In this study, we assumed that w_{ZBB} of each circuit varies from 25 to 35 and we used this assumption to estimate the $MTTF_{leak}$ throughout our experiments. Although the recent $MTTF_{leak}$ model is expressed as the proportional function of the change in temperature as seen in (5-5), to the best of our knowledge none of the recent studies has provided the absolute value of $MTTF_{leak}$. As a result, the calculated $MTTF_{leak}$ in this study can be assumed to be an estimation, albeit a good approximation in our view. In any case, our proposed techniques can comply with more accurate $MTTF_{leak}$ models, if available, without any change in the main algorithms. In addition, since this study mainly focuses on the effects of SE and leakage power under BB voltage variation, we exclude the dynamic power which also partly contributes to the overall circuit MTTF.

5.3 Optimal Solution

In our optimization problem, the maximization of $MTTF_{circuit}$ in (5-7) is defined as the objective

function with BB values, $V_{BB,i}$, and gate output arrival delays as variables. The constraints of the problem consist of conservation of the critical delay, and the boundary condition of the delay of each gate.

The worst case propagation delay of each gate in the cell library can be obtained by SPICE simulation. From SPICE simulation, we conclude that the relationship between worst case propagation delay and BB of all the mapping gates is linear. In the rest of this chapter, the term "*propagation delay*" is referred to the "*worst case propagation delay*".

In this study, we require no delay penalty and thus, the critical delay, T_{max} for each circuit at original configuration is conserved. The T_{max} is associated to the constraint of the problem as a limit of the arrival time at each primary output. We can extract T_{max} using an algorithm for finding the critical paths in the circuit graph.

The optimization problem is formulated as follows. For a gate i with propagation delay d_i, we define a variable a_i as the arrival time at the output of the gate i. Given a gate f connected to an input of the gate i, the objective and constraints of the problem are defined as given in (5-8), (5-9), (5-10), and (5-11) [59].

$$max \{MTTF_{circuit}\} \tag{5-8}$$

$$a_f + d_i \leq a_i, \forall f \in fanin\ of\ i \tag{5-9}$$

$$\left.\begin{matrix} a_i = 0 \\ d_i = 0 \end{matrix}\right\} \forall i \in PI \tag{5-10}$$

$$a_i \leq T_{max}, \forall i \in set\ of\ gates\ connected\ to\ PO \tag{5-11}$$

In the problem constraints stated above, we set each primary input as a dummy gate which satisfies the condition given in (5-10). The arrival delay at the primary output must not exceed T_{max} as stated in (5-11). In addition to the circuit/gate delay related constraints, the upper and lower limits of the BB voltage variable as given in (5-12) are also required for the problem.

$$V_{BB,MIN} \leq V_{BB,i} \leq V_{BB,MAX} \tag{5-12}$$

To improve the solvability of the optimization problem, we first consider all $V_{BB,i}$ variables as continuous values. The optimal results are then required to be discretized to desired voltage levels. This can be done as follows. Initially, each continuous value of $V_{BB,i}$ is rounded to the nearest available BB voltage. After that, the new critical delay as a result of new BB configuration is evaluated. If the new critical delay does not satisfy (5-11), we select a gate in critical path, one at a time, and reassign its BB one level towards FBB to recover the circuit performance. The candidate gate must have the lowest leakage rate of increase with respect to BB voltage and largest SER when moving towards FBB. We iteratively treat one gate at a time and improve the circuit delay until the new critical delay does not exceed T_{max}. Note that optimality of the solution is not guaranteed due to the last step, but in all cases it is expected to be close to optimal, if not optimal.

5.4 Heuristic Solution

In our heuristic driven technique, we focus on decreasing the amount of leakage current while limiting the increase in SER of each gate, and rigorously maintaining critical delay of the circuit. The basic idea behind this method is that large RBB should be assigned to gates that have sharp leakage reduction gain when BB is moved towards RBB, but are less sensitive to SE. Hence, we can lower large amount of leakage current of these gates without significantly impacting the SER. On the other hand, those gates that have small leakage reduction gain but high SE sensitivity are assigned large FBB, and as a result, they are good candidates for recovering the circuit delay without high cost of increased leakage.

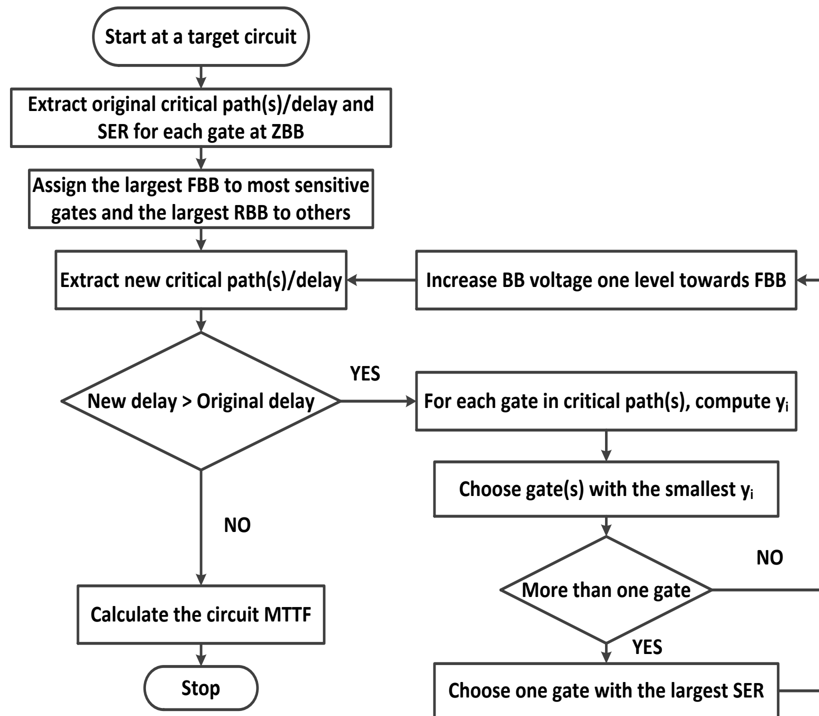


Figure 5.2: Flowchart of heuristic driven based leakage and soft error-related reliability improvement

The flowchart, as given in Figure 5.2, shows the working of our heuristic based leakage and SE-related reliability improvement method. Initially, the original critical delay of a circuit and SER of each gate with ZBB condition are computed. Then, all gates which are most sensitive to SE are assigned the largest FBB level, whereas the others are assigned the largest RBB. The number of the most sensitive gates is defined by the user. Since any change in BB towards RBB causes the gate or the circuit delay to increase, the following steps are used to recover the excessive delay to maintain the overall required critical delay.

We start the first iteration by extracting new critical delay of the circuit. After this process, we need to check whether new critical delay, as a result of assigned RBB, is lager or not than the original critical delay of the circuit. If the new critical delay is lower or equal to the original critical delay, the algorithm is complete. However, if the new critical delay is greater than the original critical delay, we must increase BB of some gates towards FBB to recover the circuit

performance. The procedure for selecting the gates to recover the delay is as follows. In each round of iterations, we search for a gate in the new critical path(s) that has the smallest leakage increasing rate, y_i, with respect to its FBB. The value of y_i can be obtained by finding the derivative of $I_{leak,i}$ in (5-2) (with respect to $V_{BB,i}$) as shown in (5-13).

$$y_i = \left.\frac{\partial I_{leak,i}}{\partial V_{BB,i}}\right|_{V_{BB,i}=\text{assigned BB}} \quad (5\text{-}13)$$

If more than one gate shares the same y_i, we select the gate which is the most sensitive to SE, or has the largest SER. Once such a gate is found, we then increase its BB one step towards FBB to make this gate faster. We iteratively search for new critical path(s), update gate delay(s), and readjust the BB of the candidate gate until the circuit delay is satisfied.

5.5 Experimental Results

We used MATLAB optimization toolbox to solve the nonlinear optimization problem with linear constraints as stated in (5-8)-(5-12), while the heuristic driven simulator was implemented with JAVA. The framework was implemented in an Intel Quad-Core machine with 4-GHz CPU clock and 6-GB SDRAM. Our cell library contains 2-, 3-, and 4-input NAND and NOR gates, and inverters. All results reported in this section are normalized with respect to the values of baseline configuration (ZBB is applied to all gates in each circuit). We also show the average CPU runtime for solving the optimization problem with different w_{ZBB} values. Note that runtime for fault simulation required for collecting logical masking probability is excluded from these results.

5.5.1 Results from the optimal solution

Table 5.1 shows the normalized results of delay, SER, leakage current, and MTTF; and the

Table 5.1: Optimal Solutions for Small Circuits with Different w_{ZBB} Values

Circuit	w_{ZBB}= 25				w_{ZBB} = 30				w_{ZBB} = 35				Ave. CPU time (10^3 s)
	Delay	*SER*	*Leakage*	*MTTF*	*Delay*	*SER*	*Leakage*	*MTTF*	*Delay*	*SER*	*Leakage*	*MTTF*	
C499	0.9998	0.9958	0.6482	1.0294	0.9976	0.9852	0.7635	1.0303	0.9959	0.9837	0.7952	1.0281	17.24
C880	0.9893	0.9970	0.6565	1.0278	1.0000	0.9972	0.6321	1.0246	0.9996	0.9928	0.6797	1.0241	20.64
C1355	0.9983	0.9966	0.6896	1.0263	0.9983	0.9966	0.6905	1.0225	1.0000	0.9958	0.7060	1.0200	15.76
S208	0.9920	0.9860	0.6612	1.0388	0.9952	0.9834	0.7221	1.0344	0.9952	0.9808	0.7570	1.0330	0.76
S420	0.9984	0.9862	0.6467	1.0394	0.9937	0.9829	0.7074	1.0357	0.9843	0.9811	0.7626	1.0325	3.22
S838	0.9997	0.9854	0.6541	1.0398	0.9935	0.9832	0.7024	1.0357	0.9937	0.9820	0.7088	1.0340	18.47
i1	0.9904	0.9620	0.8126	1.0542	0.9904	0.9606	0.8368	1.0517	0.9904	0.9600	0.8488	1.0501	0.07
i2	0.9920	0.9138	0.5891	1.1239	0.9921	0.9132	0.6040	1.1190	0.9823	0.9115	0.6674	1.1150	2.48
i3	0.9963	0.9259	0.7536	1.0988	0.9963	0.9258	0.7558	1.0957	0.9866	0.9257	0.7628	1.0932	0.60

average CPU time. In this experiment, the minimum RBB and maximum FBB are set to -0.4 and 0.3 V, respectively because beyond this BB range, we may face either leakage reduction gain saturation or gate malfunction. The BB voltage assigned to each gate is available in steps of 0.1 V between the minimum RBB and maximum FBB. We also alter the w_{ZBB} to 25, 30, and 35. In this table, the increase in circuit MTTF yielded from our optimization based approach ranges from 2% to 12% (the circuit i2 gains the most, whereas the circuit C1355 gains the least benefit). This approach can reduce the leakage current up to 40% and interestingly, no increase in SER is seen in all experimental circuits.

As seen in Table 5.1, the disadvantage of this approach is that it requires very large computation time. As a result, using the optimal solution to solve large circuits is inefficient or impractical and in some cases impossible. To extend the investigation domain to large circuits, we need a method with reduced complexity while providing optimal solution. To fulfill this requirement, we developed a heuristic based approach, and we will discuss this novel technique for reliability improvement for large circuits in the next section

5.5.2 Justification of the heuristic based approach

Table 5.2 shows the normalized results of delay, SER, leakage current, and the percentage increase (positive sign) or decrease (negative sign) in MTTF compared to the optimal MTTF in

Table 5.2: Heuristic Based Solutions for Small Circuits with Different w_{ZBB} Values

Circuit	Delay	SER	Leakage	%Change in MTTF			CPU time (s)
				w_{ZBB}			
				25	*30*	*35*	
C499	0.9997	0.9902	0.7497	-0.029	-0.427	-0.428	8.16
C880	0.9997	0.9821	0.7524	+0.924	+0.937	+0.761	1.66
C1355	0.9992	0.9878	0.8307	-0.029	+0.127	+0.216	9.19
S208	0.9992	0.9874	0.6484	-0.067	-0.048	-0.203	0.14
S420	1.0000	0.9849	0.6793	-0.038	-0.058	-0.019	0.39
S838	0.9999	0.9831	0.6980	-0.010	0.029	-0.068	1.46
i1	0.9907	0.9709	0.7681	-0.588	-0.637	-0.686	0.08
i2	0.9966	0.9120	0.6657	-0.214	-0.143	-0.054	0.53
i3	0.9954	0.9244	0.8153	-0.273	-0.201	-0.128	0.15

Table 5.1. Table 5.2 also contains the CPU time for the heuristic based approach for the same small circuits as reported in Table 5.1 in the previous section. In this experiment, we allow BB to be available in the steps of 0.1 V within the range of minimum and maximum BB of -0.4 and 0.3 V, respectively. The BB of top 10% of the most sensitive gates is initially assigned with 0.3 V which is the maximum FBB considered in this study. In addition, the same values of w_{ZBB} as reported in Table 5.1 are used (w_{ZBB} is varied as 25, 30, and 35). It is evident from Table 5.2 that our heuristic technique provides the circuit MTTF very close to the optimal results. In some circuits, the MTTF change from this technique is as small as 0.01% relative to the result from optimization approach while requiring insignificant CPU runtime for all cases compared to runtimes required for solving the optimization problem. We can also see from Table 5.2 that in some cases, the heuristic solutions are better than the optimal solutions because of the following reason.

The algorithm to discretize optimal solutions to desired BB voltage levels may cause the results to lack the optimality. For instance, the MTTF values for the circuits C880 and C1355 resulted from the optimization based technique, as reported in Table 5.1, are found to be lower than the results from the heuristic approach in Table 5.2. Basically, the optimization problem focused in this study can be solved using an exact integer nonlinear programming (INLP) formulation. However, searching for an INLP solution is inefficient in solvability and runtime.

Table 5.3: Heuristic Based Solutions for Large Circuits and Others with Different w_{ZBB} Values

Circuit	Delay	SER	Leakage	MTTF			CPU time (s)
				w_{ZBB}			
				25	*30*	*35*	
C1908	0.9997	0.9653	0.7949	1.0519	1.0493	1.0474	2.90
C5315	0.9999	0.9640	0.7688	1.0553	1.0523	1.0502	44.56
C6288	1.0000	1.0082	0.6814	1.0152	1.0113	1.0086	123.6
S13207	1.0000	0.9961	0.5817	1.0324	1.0277	1.0243	103.6
S15850	1.0000	0.9969	0.6015	1.0307	1.0262	1.0229	358.1
S35932	0.9999	0.9965	0.8695	1.0144	1.0126	1.0113	29790
i4	0.9996	0.9631	0.6894	1.0615	1.0577	1.0549	0.60
i5	0.9959	0.9900	0.7508	1.0292	1.0261	1.0238	0.25
i6	0.9970	1.0091	0.7831	1.0081	1.0053	1.0033	6.52
i7	0.9973	0.9794	0.7212	1.0421	1.0386	1.0361	7.36
i8	0.9997	0.9692	0.6811	1.0554	1.0515	1.0487	67.15

Our proposed optimization formulation in conjunction with discretization algorithm can still provide satisfactory results with small deviation from optimality. Yet, it performs with high runtime efficacy compared to INLP formulation.

The results from the heuristic technique for large circuits and others are reported in Table 5.3. The MTTF results in this table are normalized with respect to the original MTTF of each baseline circuit (ZBB applied to all gates). With the same conditions as set in the experiment corresponding to Table 5.2, it can be seen that our proposed method can improve the MTTF up to 6%, while the CPU runtime values vary within the large range from few seconds to a few hours depending on the size of the circuit.

In the heuristic based technique, we initially assigned the maximum FBB voltage to 10% gates with highest SE vulnerability for all experimental circuits. We drew this conclusion from directly observing the optimal solutions for small circuits which tend to assign large FBB to a small number of gates that are most sensitive. Since number of those candidate gates varies from circuit to circuit and is difficult to determine, we strictly set the number of topmost sensitive gates for all circuits to 10%. Although, this setting can cause declination in MTTF gain, it is evident from the experimental results that most heuristic solutions are still close to the optimal results.

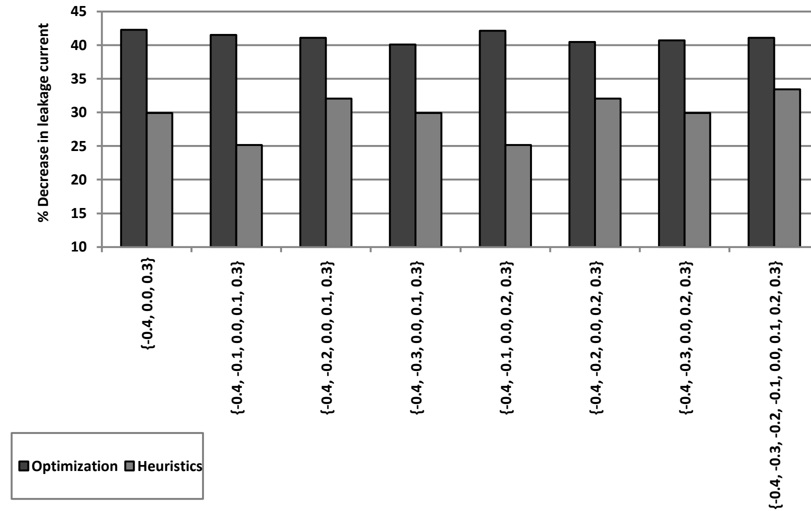


Figure 5.3: Percentage decrease in leakage current of the circuit i2 for different sets of BB voltages, $w_{ZBB} = 25$.

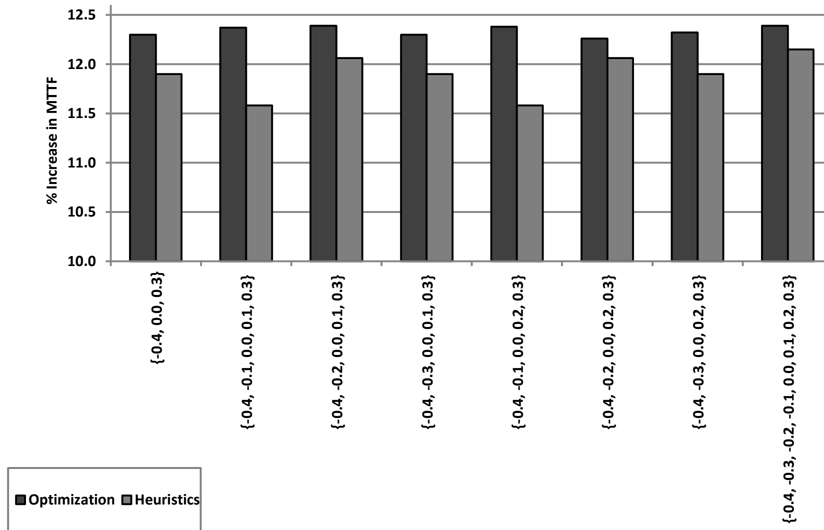


Figure 5.4: Percentage increase in MTTF of the circuit i2 for different sets of BB voltages with $w_{ZBB} = 25$.

5.5.3 Impact of body bias voltage levels

We further investigate the impact of available BB levels on leakage reduction and MTTF. Figure 5.3 and Figure 5.4 plot the decrease in leakage current and increase in MTTF of the circuit i2, respectively, with $w_{ZBB} = 25$. We also provide the comparison of the optimal and heuristic

solutions in these figures. In this experiment, a number of BB configurations, containing different sets of available BB voltages, shown below horizontal axes of the bar charts in Figure 5.3 and Figure 5.4, are assigned to the circuit. It can be seen from Figure 5.3 that choices of BB voltage levels have considerable impact on heuristic results. The leakage reduction gain, from the heuristic based technique, in Figure 5.3 drops from 34% to 25% for some BB configurations as opposed a small decrease for optimal solution. However, we notice that the MTTF gains illustrated in Figure 5.4, for both optimization and heuristic approaches, are not only close but also less sensitive to BB voltage configurations. For the smallest configuration, which contains only three BB voltage levels of {-0.4, 0.0, 0.3}V, the percentage increase in MTTF is high and even higher than the results from some configurations which allow more voltage levels. The advantage for having the small number of available BB voltage levels is that the overall layout cost of routing and placement complexity is reduced substantially. We believe that our proposed technique will retain this benefit without significant impact on reliability improvement yield.

Figure 5.5 compares BB voltage of each gate in the circuit i2 resulted from the optimization and heuristic based techniques for reliability improvement assuming that the set of available BB voltages consists of {-0.4, 0.0, 0.3} V. In this figure, each gate is indexed by the integer from 0 to 147 (the circuit i2 contains 148 gates). In addition, for the optimization solution, the w_{ZBB} is set to be 25. It can be seen from Figure 5.5 that both traces have good correlation, in other words

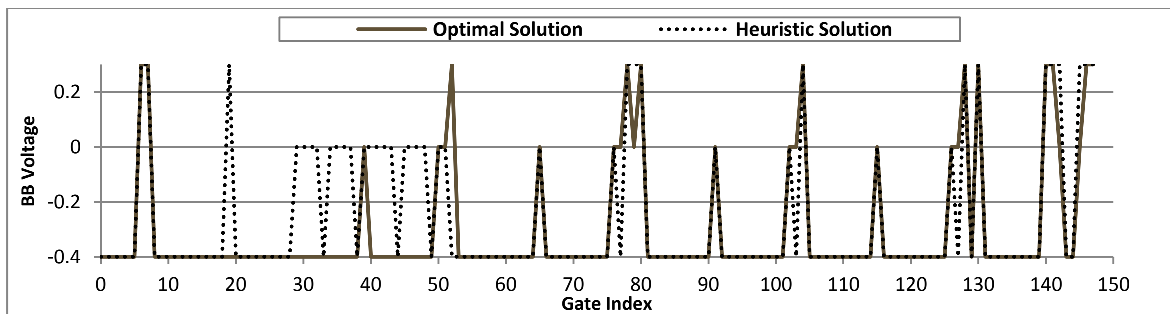


Figure 5.5: BB voltages applied to each gate in the circuit i2 with $w_{ZBB} = 25$

Table 5.4: Maximization of MTTF with Three Available BB Voltages for All Circuits

Circuit	Delay	SER	Leakage	MTTF $w_{ZBB} = 25$	CPU Time (s)
C499	0.9978	0.9874	0.8264	1.0266	2.20
C880	0.9991	0.9821	0.7659	1.0364	0.56
C1355	0.9977	0.9862	0.9508	1.0178	2.73
C1908	0.9967	0.9654	0.8121	1.0506	0.85
C5315	0.9995	0.9641	0.7757	1.0546	13.52
C6288	0.9997	1.0079	0.7278	1.0128	33.77
S208	0.9942	0.9869	0.6698	1.0374	0.08
S420	0.9971	0.9849	0.6919	1.0382	0.17
S838	0.9976	0.9830	0.7115	1.0389	0.56
S13207	0.9996	0.9961	0.5836	1.0323	36.30
S15850	0.9995	0.9979	0.6094	1.0293	112.0
S35932	0.9995	0.9961	0.9310	1.0097	7996
i1	1.0000	0.9702	0.7782	1.0480	0.05
i2	0.9954	0.9120	0.6822	1.1203	0.18
i3	0.9848	0.9243	0.8684	1.0913	0.10
i4	0.9949	0.9629	0.7164	1.0600	0.24
i5	1.0000	0.9900	0.7536	1.0291	0.14
i6	1.0000	1.0012	1.0159	0.9974	2.87
i7	1.0000	0.9781	0.7772	1.0398	2.60
i8	0.9936	0.9690	0.7060	1.0541	18.41

our heuristic based technique provides solution which is very similar to the one obtained by the optimal BB configuration.

5.5.4 Results for reduced body bias voltage level

Table 5.4 contains the normalized results of delay, SER, leakage current, MTTF, and CPU runtime for all experimental circuits using the heuristic driven approach with reduced number of available BB voltage levels. In this investigation, we allow BB voltages of {-0.4, 0.0, 0.3} V. to be available for each gate. Compared with the results in Table 5.2 and Table 5.3 which require many more BB voltage levels (all increments of 0.1 V. between -0.4 and 0.3 V.), in most cases, the MTTF results in Table 5.4 are close to those from the previous experiments. Besides, the CPU time required for solving the problem for each circuit decreases substantially, since fewer voltage levels reduce the number of iterations during delay recovery process.

Chapter 6. Soft Error Reduction in Sequential Circuits

In sequential circuits, which consist of combinational logic circuit and flip-flops, one of the efficient methods to combat SEs is selectively replacing original flip-flops with hardened flip-flops. This method is known as flip-flop selection technique. Using sizing based methods as proposed in Chapter 4 in conjunction with the flip-flop selection technique can help further reduce SER and we discuss this combined technique in this chapter.

First of all, the following section describes the definition of flip-flop selection problem. Next, we introduce a novel optimization formulation for selecting candidate flip-flops which can maximize SE coverage. We define a new metric called re-execution penalty index as a function of SER to indicate overall timing penalty required to resume the operation after appearance of a SE. Then, we describe a SE reduction technique based on the combination of sizing and flip-flop selection approaches. Finally, the experimental results are shown.

6.1 Defining a Flip-Flop Selection Problem

SEs generated in combinational units of a sequential circuit can be detected by a capturing flip-flop. This flip-flop is a modified version of ordinary flip-flop and it can be designed as follows. Based on timing redundancy mechanisms, a set of duplicated flip-flops, with a comparator and time shifted clock or data inputs [21], [50], can replace an original flip-flop to have ability to detect transient errors. Time shifted clock input is done by supplying a delayed clock to one of the flip-flops whereas for time shifted data input, the data input of one of the flip-flops is delayed. Both solutions make two flip-flops sample the data at different time. When outputs of the main and shadow flip-flops are mismatched due to a SE, an error flag is raised by the

	FF-1	FF-2	FF-3	…	FF-j	…	FF-n	Weight
Fault-1	b_{11}	b_{12}	b_{13}	…	b_{1j}	…	b_{1n}	w_1
Fault-2	b_{21}	b_{22}	b_{23}	…	b_{2j}	…	b_{2n}	w_2
Fault-3	0	1	1	…	b_{3j}	…	b_{3n}	0.13
Fault-i	b_{i1}	b_{i2}	b_{i3}	…	b_{ij}	…	b_{in}	w_i
:	:	:	:	:	:	:	:	:
Fault-m	b_{m1}	b_{m2}	b_{m3}	…	b_{mj}	…	b_{mn}	w_m

Figure 6.1: Fault coverage table

comparator indicating that the SE is detected and the system requires re-execution.

In general, a flip-flop selection technique offers smart choices for adding SE detectors to candidate flip-flops. The objective of this method is to maximize detected SEs. In particular, the flip-flop selection problem can be derived as follows. We use the proposed SE simulator for sequential circuit to construct a fault coverage table as shown in Figure 6.1. For a circuit with m total faults (SEs) and n flip-flops, this table records information of all faults, including the weight of each fault, and flip-flops where the fault can be detected. In the table, b_{ij} =1 if Fault-i is stored in (or can be detected by) flip-flop j, and b_{ij} = 0 otherwise. The weighting factor w_i of corresponding fault i is the ratio of active area to the circuit area, which depends on the position of the strike, and it is also used in (3-6) in Chapter 3. For example, in Figure 6.1, Fault-3 can be detected at FF-2 and FF-3 but not at FF-1. The corresponding weighting factor of Fault-3 is equal to 0.13. If a flip-flop is selected to be a SE detector, all faults captured by this flip-flop are then removed from the table (i.e., if FF-2 in Figure 6.1 is selected, then Fault-3 will be discarded). The sum of wi of each removed fault implies the number of detected SEs. Finally, the problem to be solved can be stated that for given area overhead how we select candidate flip-flops such that the number of detected SEs is maximized.

In the following section, we introduce an optimization based flip-flop selection which can guarantee the maximization of the removed faults.

6.2 Optimal Solution for Flip-Flop Selection

In this section, a binary integer linear programing (BILP) based optimization approach for flip-flop selection is introduced. This technique guarantees the best SE coverage obtainable by selected flip- flops.

The BILP formulation consists of solving for two vectors of binary variables. The first vector ***a*** = [a_1, a_2, a_3… a_n], $a_j \in \{0, 1\}$ is called a vector of variables of selection. A variable a_j indicates whether the corresponding flip-flop j is selected (value of 1) or not (value of 0). The second vector ***v*** = [v_1, v_2, v_3… v_m], $v_i \in \{0, 1\}$ is related to faults that are candidate set of SEs**.** If the Fault-i is detected, then the corresponding variable v_i is assigned a value of 1 and if this fault is not detected, then the variable is assigned a value of 0. Two constants in this formulation are the binary constant b_{ij}, and the weighting factor w_i of corresponding fault i and flip-flop j contained in the table as illustrated in Figure 6.1 and discussed in the previous section. Note that to reduce the number of variables in the problem, for some faults that can be detected by the same set of flip-flops, we reckon all of them as one combined fault i with its w_i equal to the sum of weighting factor of each fault. All constants in the fault coverage table given in Figure 6.1 can be obtained during gate level simulation. The variables in ***a*** and ***v*** will be solved by BILP optimizer. The proposed BILP formulation is as follows.

We define the number of flip-flops, l, as a constraint and the objective is to maximize the number of SEs detected at these flip-flops with appropriate weights. These conditions are stated as a constraint function shown in (6-1) below:

$$\sum_{j=1}^{n} a_j = l \qquad (6\text{-}1)$$

and an objective function in the BILP formulation for flip-flop selection is expressed in (6-2).

$$max \sum_{i=1}^{m} w_i v_i \qquad (6\text{-}2)$$

For fault detection mechanism, a Fault-i is detected (v_i is set to be 1) when it appears in one of the flip-flops which is selected (one of the b_{ij} components is equal to 1 while a_j of the corresponding flip-flop j is also equal to 1). As a result, this condition can be written as a set of the constraints given in (6-3) and (6-4).

$$\forall\, j{:}\, b_{ij}a_j \;\leq\; v_i \qquad (6\text{-}3)$$

$$\textstyle\sum_{i=1}^{n} b_{ij}a_j \;\geq\; v_i \qquad (6\text{-}4)$$

We use MATLAB optimization toolbox to solve the BILP problem as expressed in (6-1)-(6-4). The results of applying this technique on a number of benchmark circuits are reported and discussed in section 6.5.

6.3 Re-Execution Penalty Index

We believe that a fault that is not detected at lower level of abstraction may potentially bring severe penalty to a system to resume the operation at higher level. This performance penalty can be considered as re-execution requirement. For a hardened circuit against SEs using error detection scheme, a re-execution penalty index P is defined as a function which combines the detected SE ($SER_{removed}$) with the undetected SE ($SER_{not\text{-}removed}$) as shown in (6-5). Clearly, there is no penalty associated with faults that are masked.

$$P = K_{removed}SER_{removed} + K_{not-removed}SER_{not-removed} \qquad (6\text{-}5)$$

In equation (6-5), $K_{removed}$ and $K_{not\text{-}removed}$ are constants which vary from circuit to circuit. Although addressing both types of SEs requires re-execution, SEs that cannot be detected ($SER_{not\text{-}removed}$) contribute more substantially to the total performance loss or the defined re-execution penalty index, compared to those which are detected. As a result $K_{removed}$ is smaller than $K_{not\text{-}removed}$. For comparison purpose, the re-execution penalty index after hardening is

normalized with respect to the original circuit condition. If P_0 is the re-execution penalty index of a circuit at the original design with SER_0 as the original SER, the normalized re-execution penalty index can be written as (6-6)

$$\frac{P}{P_0} = \frac{1}{SER_0}\left(\frac{K_{removed}}{K_{not-removed}} SER_{removed} + SER_{not-removed}\right) \quad (6\text{-}6)$$

6.4 Combined Flip-Flop Selection and Sizing Technique

We integrate flip-flop selection and sizing approaches together to improve our ability to handle SEs. This combined technique shares the area overhead between selected flip-flops and sensitive gates. The best distribution of area overhead to flip-flops and logic gates in the circuit is finally chosen based on the reduction in normalized re-execution penalty index of a circuit.

For each round of treatment, flip-flop selection is performed first. For small circuits, we can use the optimization approach to select candidate flip-flops as discussed before, yet solving the BILP problem for large circuits requires significant computation time. Hence, for large circuits, we use a heuristic with greedy approach, shown in Figure 6.2, to select flip-flops which offer

```
For each injected fault {
     If fault appears at primary outputs or flip-flop(s)
         Record fault's weight (wi), gate that generates the fault, and
         flip-flops that capture the fault;
}

For each flip-flop {
     Select a flip-flop with the largest sum of weight of each fault;
     Remove the detected faults from other flip-flops and primary
     outputs;
}
Update sensitivity of all gates;
```

Figure 6.2: Heuristic flip-flop selection pseudocode

large amount of SE coverage. This flip-flop selection approach can be explained as follows.

During fault injection, a fault coverage table as shown in Figure 6.1 is constructed. In addition, to obtain the SE sensitivity of each gate, the index of any gate that is the source of the SE is also recorded. After all faults are injected, we search for a flip-flop with the largest sum of fault weight and add it to the set of selected flip-flops. Then, we remove the detected faults from primary outputs and other flip-flops. We repeatedly search for a flip-flop with the largest fault coverage one at time until the number of selected flip-flops is satisfied. After we completely identify all selected flip-flops, the sensitivity to SE of each gate is updated to be used in the next step.

In the next step, we apply sizing technique to further reduce $SER_{not\text{-}removed}$ of a circuit. The part of the area overhead which is not used for flip-flops is distributed to most sensitive gates using parallel network sizing based technique discussed in Chapter 4. This sizing approach includes the weighted area distribution algorithm which fairly assigns additional area to gates

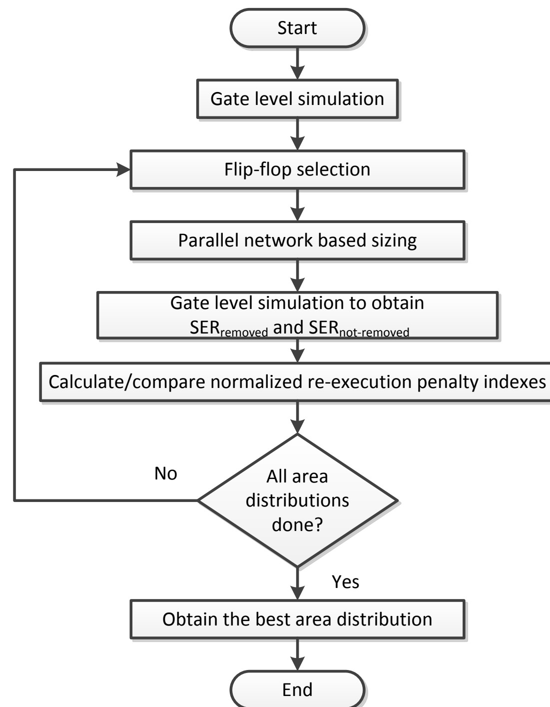


Figure 6.3: Flowchart of the combined flip-flop selection and sizing technique

based on their sensitivity by providing larger area to more sensitive gates. In addition, it also takes into consideration the SER saturation which limits a decrease in SER of a gate when its parallel network is over upsized. After candidate flip-flops are selected and additional area is distributed to most sensitive gates, we perform gate level simulation for sequential circuits (as illustrated in Figure 3.3 in Chapter 3) on the hardened circuit again to obtain the $SER_{removed}$ and $SER_{not\text{-}removed}$. Information from this simulation is used to evaluate SE mitigation yield of our method.

We use (6-6) to evaluate the improvement by the methods proposed in this chapter. For a given circuit, various area distributions (related to the number of selected flip-flops and area given to sensitive gates) may be investigated. Therefore, we can compare the normalized re-execution penalty indices for different area distributions, and identify the best distribution providing the smallest normalized re-execution penalty index. The flowchart of combined flip-flop selection and sizing technique is illustrated in Figure 6.3. We provide experimental results of our proposed re-execution penalty reduction in the next section.

6.5 Experimental Results

To assess the SE mitigation using our methods and to compare it with other methods, we selected a number of benchmark circuits from ISCAS'89 to be our experimental circuits. All circuits are mapped with 32-nm predictive technology model from [56]. The gate cell library used for all circuits consists of 2-, 3-, and 4-input NAND and NOR gates; inverters; and flip-flops. During gate level simulation, 100,000 random input vectors are generated at the beginning of the cycle time for each circuit. The results of SER and re-execution penalty index are reported in normalized form with respect to the original design.

Table 6.1: Area Overhead and Normalized SER from Optimal and Heuristic Flip-Flop Selection

Fraction of selected flip-flops = 0.4					
Circuit	*#Selected flip-flops*	*%Area overhead*	*Optimal* $SER_{removed}$ [this work]	*Heuristic* $SER_{removed}$ [this work]	*Heuristic* $SER_{removed}$ [21]
S298	6	4.58	0.5265	0.5265	0.5188
S344	6	4.60	0.4691	0.4691	0.4690
S526	8	3.32	0.4752	0.4752	0.4604
S838	13	4.31	0.2500	0.2500	0.2499
S1196	7	1.49	0.1985	0.1985	0.1985
Fraction of selected flip-flops = 0.6					
Circuit	*#Selected flip-flops*	*%Area overhead*	*Optimal* $SER_{removed}$ [this work]	*Heuristic* $SER_{removed}$ [this work]	*Heuristic* $SER_{removed}$ [21]
S298	8	6.11	0.6412	0.6412	0.6342
S344	9	6.91	0.6172	0.6172	0.6172
S526	13	6.56	0.6881	0.6881	0.6854
S838	19	6.30	0.3467	0.3467	0.3456
S1196	11	8.64	0.2719	0.2719	0.2719
Fraction of selected flip-flops = 0.8					
Circuit	*#Selected flip-flops*	*%Area overhead*	*Optimal* $SER_{removed}$ [this work]	*Heuristic* $SER_{removed}$ [this work]	*Heuristic* $SER_{removed}$ [21]
S298	11	8.40	0.8018	0.8018	0.8004
S344	12	9.21	0.7525	0.7525	0.7525
S526	17	7.07	0.8343	0.8343	0.8322
S838	26	8.62	0.4339	0.4339	0.4339
S1196	14	2.98	0.3166	0.3166	0.3166

6.5.1 Results from optimal flip-flop selection

Table 6.1 shows the results from the BILP flip-flop selection algorithm based on equations (6-1)-(6-4). In Table 6.1, we also compare the normalized $SER_{removed}$ of our optimization formulation with our proposed heuristic solution and the method proposed in [21] for different fractions of selected flip-flops. Interestingly, we notice from Table 6.1 that our heuristic flip-flop selection achieves exactly the same results as the optimal solution for each experimental circuit. On the other hand our methods (heuristic as well as optimal) outperform the method proposed in [21] However, the runtime of the optimal method, though not reported in the table, is an order of magnitude larger and also for very large circuits, the optimal method could not complete execution even in a very long time. Though optimality of the proposed heuristic flip-flop selection method cannot be guaranteed, we choose our proposed heuristic flip-flop selection method for use in further experiments reported in the next subsection for its high efficiency in

Table 6.2: Combined Flip-Flop Selection and Sizing Results

All flip-flops selected without sizing							
Circuit	*#Selected FFs*	*%Area overhead*	*Not removed SER*	*Removed SER*	*Re-execution penalty index*		
					*k**=1.5	*k*=2	*k*=2.5
S298	14	10.69	0.094	0.906	0.698	0.547	0.456
S344	15	11.51	0.147	0.853	0.716	0.574	0.488
S526	21	8.73	0.047	0.953	0.682	0.524	0.428
S838	32	10.61	0.585	0.416	0.862	0.792	0.751
S1196	18	3.82	0.553	0.447	0.851	0.776	0.732
S1494	6	0.82	0.779	0.221	0.926	0.889	0.867
S5378	179	8.99	0.326	0.674	0.775	0.663	0.595
Fraction of selected flip-flops = 0.8 with sizing							
Circuit	*#Selected FFs*	*%Area overhead*	*Not removed SER*	*Removed SER*	*Re-execution penalty index*		
					k=1.5	*k*=2	*k*=2.5
S298	11	10.69	0.097	0.783	0.619	0.489	0.411
S344	12	11.51	0.188	0.661	0.628	0.518	0.452
S526	17	8.73	0.085	0.789	0.611	0.479	0.400
S838	26	10.61	0.615	0.303	0.817	0.766	0.736
S1196	14	3.82	0.555	0.378	0.807	0.744	0.706
S1494	5	0.82	0.798	0.192	0.926	0.894	0.875
S5378	143	8.99	0.269	0.585	0.659	0.562	0.503
Fraction of selected flip-flops = 0.6 with sizing							
Circuit	*#Selected FFs*	*%Area overhead*	*Not removed SER*	*Removed SER*	*Re-execution penalty index*		
					k=1.5	*k*=2	*k*=2.5
S298	8	10.69	0.100	0.721	0.580	0.460	0.388
S344	9	11.51	0.274	0.581	0.661	0.564	0.506
S526	13	8.73	0.181	0.655	0.618	0.509	0.443
S838	19	10.61	0.732	0.249	0.898	0.857	0.832
S1196	11	3.82	0.585	0.337	0.809	0.753	0.719
S1494	4	0.82	0.821	0.165	0.931	0.904	0.887
S5378	107	8.99	0.265	0.562	0.640	0.546	0.490
Fraction of selected flip-flops = 0.4 with sizing							
Circuit	*#Selected FFs*	*%Area overhead*	*Not removed SER*	*Removed SER*	*Re-execution penalty index*		
					k=1.5	*k*=2	*k*=2.5
S298	6	10.69	0.145	0.648	0.576	0.468	0.404
S344	6	11.51	0.314	0.463	0.623	0.546	0.499
S526	8	8.73	0.331	0.458	0.636	0.560	0.514
S838	13	10.61	0.877	0.187	1.000	0.970	0.951
S1196	7	3.62	0.663	0.254	0.832	0.790	0.765
S1494	2	0.82	0.907	0.091	0.967	0.952	0.943
S5378	72	8.99	0.280	0.525	0.630	0.543	0.490

* $\frac{1}{k} = \frac{K_{removed}}{K_{not-removed}}$

CPU runtime and memory usage compared to solving BILP problems.

6.5.2 Results from combined flip-flop selection and sizing technique

Table 6.2 shows the results from our combined flip-flop selection and sizing for improving the re-execution penalty. For all experiments, we vary the fractions of selected flip-flops and the area distributed to logic gates, while we set the area overhead of each experimental circuit to be the same as the area required for selecting all the flip-flops in the circuit without upsizing

combinational logic gates (as reported in the topmost section of Table 6.2). For example, for the circuit s526, if all flip-flops were selected, then the total area overhead would be 8.73%. We set this limit for the overhead of s526 in the remaining experiments. With different values of *k*, the ratio of $K_{removed}$ to $K_{not\text{-}removed}$, the normalized re-execution penalty index yields of various experimental circuits are reported in Table 6.2. It can be seen from this table that when all flip-flops are selected, the $SER_{not\text{-}removed}$ for all circuits decreases significantly; e.g., the $SER_{not\text{-}removed}$ of the circuit S526 is reduced by 95%. However, if we consider the values of normalized re-execution penalty index of the circuit S526, this circuit receives the largest improvement when we apply flip-flop selection with 0.8 fractions of total flip-flops and upsizing the most sensitive gates with the rest of the area budget. In general, the largest decrease in re-execution penalty index depends on the value of *k*. The best flip-flop selection and sizing combinations are chosen such that it provides the smallest value of re-execution penalty index.

Chapter 7. Summary

This chapter provides the summary of this dissertation and future directions of this research. We have been studying on SE reliability issues in digital circuits for many years, and parts of this research have been presented in a number of conferences and a workshop [9], [28], [29], [30], [31], [33]. However, we have improved on some of the methods proposed by us in our published papers and the improved methods are included in this dissertation. Hence, the methodologies and the results described in this thesis are more complete and efficient compared to our works in those papers.

7.1 Conclusion

This thesis addresses SEs in nanometer combinational and sequential circuits and develops novel techniques to improve circuit reliability against SEs. All proposed techniques and their accomplishment are summarized as follows.

For combinational circuits, we develop a SE simulator which includes the input dependence of SE and propose novel SE reduction techniques. First, a gate input reconfiguration technique is developed based on gate input dependence of SER. This technique is almost overhead-free since reconfiguring local connections are hardly likely to impact global placement and routing of a circuit. The experimental results show that our gate input reconfiguration approach can reduce the circuit SER as much as 77% in some circuits/technologies. As a result of its effectiveness, we use the gate input reconfiguration technique to initialize the layout of all experimental circuits before applying other treatments. Second, a nonlinear optimization and heuristic based sizing techniques are introduced to minimize SER under tight area overhead budget. Although the

optimal solution guarantees the best area distribution, it requires high CPU demand. To reduce large CPU runtime required for solving nonlinear optimization problems, the heuristic based sizing approach is developed. This approach employs two algorithms: the weighted area distribution algorithm which provides the fairness of area budget distribution based on the SER of each gate, and the gate SER saturation consideration algorithm which avoids any increase in the gate SER due to oversizing. Among all investigated sizing scenarios, the methods utilizing the heuristic based algorithms outperform other methods for most circuits and technologies with nearly optimal results. Notably, our heuristics require very small CPU runtime, in the order of 100s of seconds, compared to several hours/days required for solving the nonlinear optimization problem in large circuits.

The effects of leakage and SEs on combinational circuit reliability are also investigated. It is found through simulation that larger RBB assigned to a gate reduces the leakage but causes SER to increase, whereas larger FBB has the opposite consequence. We make use of this observation to develop an optimization and heuristic based technique for maximizing combined reliability due to leakage and SEs by adjusting the BB. In our methods, we use a common reliability metric, MTTF, to associate with both long term-related temperature failures due to elevated leakage and transient failures due to SER. Our optimization based technique maximizes the combined MTTF over BB voltage of each gate while strictly conserving the circuit performance. This technique accomplishes a large increase in MTTF but may not be scalable for large circuits due to high CPU runtime requirement. For this reason, the heuristic driven approach is developed to improve runtime efficacy and scalability for large circuits. The experimental results show that the heuristic solutions for most circuits are close to the solutions obtained by the optimization based technique while requiring very short CPU runtime compared to solving the optimization

problem.

For sequential circuits, this thesis proposes a soft error simulator and two novel techniques to mitigate SEs in sequential circuits. First, we introduce a BILP formulation for SE coverage maximization by selecting some flip-flops to be as SE detectors. This technique guarantees that for a given number of selected flip-flops, maximum SE detection is accomplished, though it requires large CPU runtime and memory usage. Second, a combined technique of flip-flop selection and gate sizing is developed. In this method, we propose a heuristic approach for flip-flop selection to reduce computational resource consumption. Experimental results show that our heuristic flip-flop selection outperforms a previously proposed method and reaches the optimality for all experimental circuits. In addition, we define a re-execution penalty index which represents the re-execution penalty required to recover a SE. We use the re-execution penalty index to guide our simulator to assign appropriate area overhead to flip-flops and logic gates. Our experimental results show that each circuit receives the best area overhead distribution when its re-execution penalty index is minimized.

7.2 Future Directions

As SE vulnerability of digital circuits continuously deteriorates, development of SE reduction techniques is increasingly challenging. The approaches that have been proposed require significant improvement to reconcile the impact of SE in state-of-the-art processors. In this last section of the dissertation, we propose future directions of our work. We believe that following guidelines can help improve the performance of SE mitigation techniques for leading edge or future technologies.

7.2.1 Hierarchical approach

Most SEs, appearing at gate level of a processor, tend to be handled by other fault tolerant mechanisms at higher levels. Lack of this consideration may overestimate the gate SER and results in unnecessary SE hardening. Although it is clear that managing SEs or other types of errors at higher levels causes large timing penalty, if any fault recovery task can be complete within latency cycles between two data-dependent instructions, there is no timing penalty for SE mitigation. Therefore, to identify more realistic SE reduction performance in gate level, we should consider that SEs, which are subsequently captured by other fault tolerant schemes with no additional timing penalty, should be excluded from gate level of the design. This consideration may require system integration as well as collaboration among design communities to obtain integrative solutions for overall system. However, different specialties among designers make this more challenging. For example, while circuit and logic gate designers help harden more vulnerable parts near the “source” of SEs, system and architecture designers exploit chip or microarchitecture level redundancy to enhance system reliability near the “sink” of SEs. In addition to good design collaboration, it is still challenging to fairly weight the importance of different types of SE management on the overall chip. However, we believe that good design procedures and well-defined interface between any adjacent levels of abstractions can lessen the design complexity for SE reduction in the whole system.

7.2.2 Soft error models for next generation devices

As gate leakage is reaching the limitation in conventional MOS devices, many manufacturers are actively researching and developing new devices which significantly gain leakage reduction. Such devices include high-k/metal gate (HKMG) and multi-gate transistors [60]. The trend of leakage in new generation processor is likely to decrease. On the other hand, SE issues have not

been well addressed for these new technologies. Unfortunately, for this recent transition from bulk CMOS to the next generation of transistor structure, few studies have investigated the impact of SEs in new generation circuits. For this reason, in our next step, we should explore SEs and develop SE models for circuits implemented with such devices. After that, we will use these accurate SE models to further investigate methods to improve reliability against SEs. We believe that good SE models will later help develop efficient SE mitigation approaches for leading edge processors.

7.2.3 Power constraint optimization

In some parts of a design where power constraint is strictly conserved, increased power as a result of particle strikes may permanently damage the system. To protect these parts, power profiles of all victim nodes for each strike are required to define node sensitivity due to SET-induced power. In this protection scheme, instead of applying SER sensitivity based gate hardening as mainly focused in this dissertation, we should use power sensitivity of each node to guide the simulator to protect most sensitive nodes. However, an integrated reliability model should be identified as power and SEs have different types of reliability degradations (related to permanent and transient failures, respectively). The challenge of solving this problem also relies on the interdependent of these two phenomena.

7.2.4 Multiple-strike effect

In this study, we focus on the single neutron strike effect such that a particle hit occurs, in a given timing window, at only one transistor. However, with node sizes shrinking to a point that even a very low energy particle can cause a SE, in the future devices multiple strikes at different devices are likely to happen more frequently. This is because low energy neutrons typically have high strike rate. Multiple-strike effect impacts several traditional fault detection mechanisms that

can handle single upset at a time. These techniques include some hardware and information (coding) redundancy methods. For this reason, first, accurate gate level SE models which take multiple-strike effect into account should be developed. After that, hardening techniques against multiple-strike SEs should be further investigated.

Bibliography

[1] D. G. Mavis and P. H. Eaton, "Soft Error Rate Mitigation Techniques for Modern Microcircuits," in *Proc. of the 40th International Reliability Physics Symposium*, Dallas, Texas, 2002, pp. 216-225.

[2] Y. Tosaka et al., "Cosmic Ray Neutron-Induced Soft Errors in Sub-Half Micron CMOS Circuits," *IEEE Electron Device Letters*, pp. 99-101, March 1997.

[3] S. Mitra, M. Zhang, N. Seifert, T.M. Mak, and K. S. Kim, "Built-In Soft Error Resilience for Robust System Design," in *Proc. of the IEEE International Conference on Integrated Circuit Design and Technology*, Austin, TX, 2007, pp. 1-6.

[4] A. H. Johnston, "Scaling and Technology Issues for Soft Error Rates," in *Proc. of the 4th Annual Conference on Reliability*, Standford University, CA, 2000.

[5] F. Wang and V. D. Agrawal, "Single Event Upset: An Embedded Tutorial," in *Proc. of the 21st International Conference on VLSI Design*, Hyderabad, India, 2008, pp. 429-434.

[6] E. Sicard et al., "A Cooperative Research for Experimental Characterization of Signal Integrity in Deep Submicron Integrated Circuits," in *Proc. of the IEEE International Symposium on Electromagnetic Compatibility*, Seattle, WA, 1999, pp. 361-364 vol.1.

[7] K. Bhattacharya, "A Unified Gate Sizing Formulation for Optimizing Soft Error Rate, Cross-Talk Noise and Power under Process Variations," in *Proc. of the 10th International Symposium on Quality Electronic Design (ISQED)*, San Jose, CA, 2009, pp. 388-393.

[8] T. Karnik, "Characterization of Soft Errors Caused by Single Event Upsets in CMOS Processes," *IEEE Transactions on Dependable and Secure Computing* , vol. 1, no. 2, pp. 128-143, April-June 2004.

[9] W. Sootkaneung and K. K. Saluja, "Impact of Body Bias Based Leakage Power Reduction on Soft Error Rate," in *Proc. of the 25th International Conference on VLSI Design*, Hyderabad, India, 2012, pp. 74-79.

[10] A. Biswas et al., "Explaining Cache SER Anomaly Using DUE AVF Measurement," in *Proc. of the 16th IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, Bangalore, India, 2010, pp. 1-12.

[11] S. Paul, F. Cai, X. Zhang, and S. Bhunia, "Reliability-Driven ECC Allocation for Multiple Bit Error Resilience in Processor Cache," *IEEE Transactions on Computers*, vol. 60, no. 1, pp. 20-34, January 2011.

[12] H. Amrouch and J. Henkel, "Self-Immunity Technique to Improve Register File Integrity against Soft Errors," in *Proc. of the 24th International Conference on VLSI Design*, Chennai, India, 2011, pp. 189-194.

[13] P. Reviriego, M. Flanagan, and J. A. Maestro, "A (64,45) Triple Error Correction Code for Memory Applications," *IEEE Transactions on Device and Material Reliability*, vol. 12, no. 1, pp. 101-106, March 2012.

[14] H. R. Zarandi and S. G. Miremadi, "Soft Error Mitigation in Cache Memories of Embedded Systems by Means of a Protected Scheme," *Lecture Notes in Computer Science, Springer Berlin / Heidelberg*, vol. 3747, pp. 121-130, 2005.

[15] V. Gherman, S. Evain, M. Cartron, N. Seymour, and Y. Bonhomme, "System-Level Hardware-Based Protection of Memories against Soft-Errors," in *Proc. of the Design, Automation and Test in Europe (DATE)*, Nice, France, 2009, pp. 1222-1225.

[16] J. Hu, S. Wang, and S. G. Ziavras, "On the Exploitation of Narrow-Width Values for Improving Register File Reliability," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 17, no. 7, pp. 953-963, July 2009.

[17] J. Lee and A. Shrivastava, "A Compiler-Microarchitecture Hybrid Approach to Soft Error Reduction for Register Files," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 7, pp. 1018-1027, July 2010.

[18] M. Chen and A. Orailoglu, "Flip-flop Hardening and Selection for Soft Error and Delay Fault Resilience," in *Proc. of the 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, Chicago, IL, 2009, pp. 49-57.

[19] S. Mitra, "Robust System Design," in *Proc. of the 23rd International Conference on VLSI Design*, Bangalore, India, 2010, pp. 434-439.

[20] R. R. Rao, D. Blaauw, and D. Sylvester, "Soft Error Reduction in Combinational Logic Using Gate Resizing and Flipflop Selection," in *Proc. of the 2006 IEEE/ACM international Conference on Computer-Aided Design*, San Jose, CA, 2006, pp. 502-509.

[21] E. L. Hill, M. H. Lipasti, and K. K. Saluja, "An Accurate Flip-Flop Selection Technique for Reducing Logic SER," in *Proc. of the International Conference on Dependable Systems and Networks (DSN)*, Anchorage, AK, 2008, pp. 128-136.

[22] J. W. Choi, B. Shim, A. C. Singer, and N. I. Cho, "Low-Power Filtering via Minimum Power Soft Error Cancellation," *IEEE Transactions on Signal Processing*, vol. 55, no. 10, pp. 5084-5096, October 2007.

[23] M. Singh and I. Koren, "Fault-Sensitivity Analysis and Reliability Enhancement of Analog-to-Digital Converters," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 5, pp. 839-852, October 2003.

[24] H. S. Deogun, D. Sylvester, and D. Blaauw, "Gate-Level Mitigation Techniques for Neutron-Induced Soft

Error Rate," in *Proc. of the 6th International Symposium on Quality of Electronic Design (ISQED)*, San Jose, CA, 2005, pp. 175-180.

[25] Q. Zhou and K. Mohanram, "Gate Sizing to Radiation Harden Combinational Logic," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and System*, vol. 25, no. 1, pp. 155-166, January 2006.

[26] K. Bhattacharya and N. Ranganathan, "Reliability-centric Gate Sizing with Simultaneous Optimization of Soft Error Rate, Delay and Power," in *Proc. of the International Symposium on Low Power Electronics and Design (ISLPED)*, Bangalore, India, 2008, pp. 99-104.

[27] N. Miskov-Zivanov and D. Marculescu, "Modeling and Analysis of SER in Combinational Circuits," in *Proc. of the IEEE Workshop on Silicon Errors in Logic-System Effects (SELSE 6)*, Standford University, CA, 2010, pp. 1-6.

[28] W. Sootkaneung and K. K. Saluja, "Sizing Techniques for Improving Soft Error Immunity in Digital Circuits," in *Proc. of the International Conference on VLSI Design and Communication Systems (ICVLSICOM)*, Chennai, India, 2010, pp. 87-92.

[29] W. Sootkaneung and K. K. Saluja, "Gate Input Reconfiguration for Combating Soft Errors in Combinational Circuits," in *Proc. of the International Conference on Dependable Systems and Networks Workshops (DSN-W)*, Chicago, IL, 2010, pp. 107-112.

[30] W. Sootkaneung and K. K. Saluja, "On Techniques for Handling Soft Errors in Digital Circuits," in *Proc. of the International Test Conference (ITC)*, Austin, TX, 2010, pp. 1-9, paper 25.2.

[31] W. Sootkaneung and K. K. Saluja, "Soft Error Reduction through Gate Input Dependent Weighted Sizing in Combinational Circuits," in *Proc. of the 12th International Symposium on Quality Electronic Design (ISQED)*, Santa Clara, CA, 2011, pp. 603-610.

[32] W. Sheng, L. Xiao, and Z. Mao, "Soft Error Optimization of Standard Cell Circuits Based on Gate Sizing and Multi-Objective Genetic Algorithm," in *Proc. of the 46th Annual Design Automation Conference (DAC)*, San Francisco, CA, 2009, pp. 502-507.

[33] W. Sootkaneung and K. K. Saluja, "Optimizing Device Size for Soft Error Resilience in Sub-Micron Logic Circuits," in *Proc. of the 2nd Asia Symposium on Quality Electronic Design (ASQED)*, Penang, Malaysia, 2010, pp. 235-242.

[34] R. R. Rao, K. Chopra, D. T. Blaauw, and D. M. Sylvester, "Computing the Soft Error Rate of a Combinational Logic Circuit Using Parameterized Descriptors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and System*, vol. 26, no. 3, pp. 468-479, March 2007.

[35] B. Zhang, W. Wang, and M. Orshansky, "FASER: Fast Analysis of Soft Error Susceptibility for Cell-Based

Designs," in *Proc. of the 7th International Symposium on Quality Electronic Design (ISQED)*, Washington, DC, 2006, pp. 755-760.

[36] A. Agarwal, S. Mukhopadhyay, A. Raychowdhury, K. Roy, and C. H. Kim, "Leakage Power Analysis and Reduction for Nanoscale Circuits," *IEEE Micro*, vol. 26, no. 2, pp. 68-80, March-April 2006.

[37] D. Brooks, R. P. Dick, R. Joseph, and L. Shang, "Power, Thermal, and Reliability Modeling in Nanometer-Scale Microprocessors," *IEEE Micro*, vol. 27, no. 3, pp. 49-62, May-June 2007.

[38] D. Arumi, R. R. Montanes, and J. Figueras, "Gate Leakage Impact on Full Open Defects in Interconect Lines," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 19, no. 12, pp. 2209-2219, December 2011.

[39] A. Sathanur, A. Pullini, L. Benini, G. De Micheli, and E. Macii, "Physically Clustered Forward Body Biasing for Variability Compensation in Nanometer CMOS Design," in *Proc. of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Nice, France, 2009, pp. 154-159.

[40] S. H. Kulkarni, D. M. Sylvester, and D. T. Blaauw, "Design-Time Optimization of Post-Silicon Tuned Circuits Using Adaptive Body Bias," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 3, pp. 481-494, March 2008.

[41] A. Ghosh, R. M. Rao, and R. B. Brown, "A Centralized Supply Voltage and Local Body Bias-Based Compensation Approach to Mitigate Within-Die Process Variation," in *Proc. of the 14th ACM/IEEE international symposium on Low power electronics and design (ISLPED)*, New York, NY, 2009, pp. 45-50.

[42] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "Adaptive Techniques for Overcoming Performance Degradation Due to Aging in CMOS Circuits," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 19, no. 4, pp. 603-614, April 2011.

[43] H. Xu, W. B. Jone, and R. Vemuri, "Aggressive Runtime Leakage Control through Adaptive Light-Weight Vth Hopping with Temperature and Process Variation," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 19, no. 7, pp. 1319-1323, July 2011.

[44] H. Mostafa, M. Anis, and M. Elmasry, "A Novel Low Area Overhead Direct Adaptive Body Bias (D-ABB) Circuit for Die-to-Die and Within-Die Variations Compensation," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 19, no. 10, pp. 1848-1860, October 2011.

[45] M. Meijer and J. P. de Gyvez, "Body-Bias-Driven Design Strategy for Area- and Performance-Efficient CMOS Circuits," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 20, no. 1, pp. 42-51, January 2012.

[46] P. Mangalagiri, S. Bae, R. Krishnan, Y. Xie, and V. Narayanan, "Thermal-Aware Reliability Analysis for Platform FPGAs," in *Proc. of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, 2008, pp. 722-727.

[47] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "Lifetime Reliability: Toward An Architectural Solution," *IEEE Micro*, vol. 25, no. 3, pp. 70-80, May-June 2005.

[48] B. Greskamp, S. R. Sarangi, and J. Torrellas, "Threshold Voltage Variation Effects on Aging-Related Hard Failure Rates," in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS)*, New Orleans, LA, 2007, pp. 1261-1264.

[49] H. Fuketa, M. Hashimoto, Y. Mitsuyama, and T. Onoye, "Alpha-Particle-Induced Soft Errors and Multiple Cell Upsets in 65-nm 10T Subthreshold SRAM," in *Proc. of the IEEE International Reliability Physics Symposium (IRPS)*, Garden Grove (Anaheim), CA, 2010, pp. 213-217.

[50] D. Ernst et al., "Razor: Circuit-Level Correction of Timing Errors for Low-Power Operation," *IEEE Micro*, vol. 24, no. 6, pp. 10-20, November-December 2004.

[51] P. Hazucha and C. Svensson, "Impact of CMOS Technology Scaling on the Atmospheric Neutron Soft Error Rate," *IEEE Transactions on Nuclear Science*, vol. 47, no. 6, pp. 2586-2594, December 2000.

[52] 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. of the International Conference on Dependable Systems and Networks (DSN)*, Bethesda, MD, 2002, pp. 389-398.

[53] G. C. Messenger, "Collection of Charge on Junction Nodes from Ion Tracks," *IEEE Transactions on Nuclear Science*, vol. 29, no. 6, pp. 2024–2031, 1982.

[54] C. Hungse, E. M. Rudnick, J. H. Patel, R. K. Iyer, and G. S. Choi, "A Gate-Level Simulation Environment for Alpha-Particle-Induced Transient Faults," *IEEE Transactions on Computers*, vol. 45, no. 11, pp. 1248-1256, November 1996.

[55] P. Jain and V. Zhu, "Judicious Choice of Waveform Parameters and Accurate Estimation of Critical Charge for Logic SER," in *Proc. of the Workshop on Dependable and Secure Nanocomputing*, Edinburgh, UK, 2007.

[56] HSPICE PTM website. [Online]. Available: http://www.eas.asu.edu/~ptm.

[57] D. Rossi, J. M. Cazeaux, M. Omana, C. Metra, and A. Chatterjee, "Accurate Linear Model for SET Critical Charge Estimation," *IEEE Transactions on VLSI Systems*, vol. 17, no. 8, pp. 1161–1166, August 2009.

[58] JEDEC89A Standard, "Measurement and Reporting of Alpha Particles and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices," Joint Electron Device Engineering Council, Solid State Technology Association, 2006.

[59] T. H. Wu, L. Xie, and A. Davoodi, "A Parallel and Randomized Algorithm for Large-Scale Discrete Dual-Vt Assignment and Continuous Gate Sizing," in *Proc. of the ACM/IEEE International Symposium on Low Power*

Electronics and Design (ISLPED), Bangalore, India, 2008, pp. 45-50.

[60] D. James, "High-k/Metal Gates in Leading Edge Silicon Devices," in *Proc. of the 23rd Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, Saratoga Springs, NY, 2012, pp. 346-353.