The Pennsylvania State University

The Graduate School

College of Engineering

SOFT ERRORS IN LOGIC CIRCUITS:
ANALYSIS AND MODELING

A Thesis in

Electrical Engineering

by

Rajaraman Ramanarayanan

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2007

The thesis of Rajaraman Ramanarayanan was reviewed and approved* by the following:

Vijaykrishnan Narayanan
Associate Professor of Computer Science Engineering
Thesis Advisor
Co-Chair of Committee

Mary Jane Irwin
Evan Pugh Professor and A. Robert Noll Chair of Engineering
Computer Science Engineering

Charles L. Croskey
Professor Emeritus of Electrical Engineering

Yuan Xie
Assistant Professor of Computer Science Engineering

Kenan Unlu
Special Signatory
Professor of Mechanical and Nuclear Engineering

W. Kenneth Jenkins
Professor and Head of the Department of Electrical Engineering
Co-Chair of Committee

*Signatures are on file in the Graduate School

ABSTRACT

Soft errors in data paths are gaining importance as technology scales down, because of increased speed, the number of stages in pipelines, and the decrease in device sizes and supply voltages. In this work, first we consider flip-flop and adder circuits, representing the data-path elements in present day processors for detailed circuit level soft error analyses. We perform simulations to analyze the effect of increasing threshold voltage in flip-flops and dynamic voltage and frequency scaling on the soft error rate in the adder circuits. Second, we also experiment with techniques to improve the soft error rate in these circuits. Next, we propose a new approach to model soft errors in larger circuits, which can be applied to designs that use cell libraries. These cell libraries are characterized for soft error analysis and utilize analytical equations to model the propagation of a voltage pulse to the input of a state element. The average error of the SER estimates, using our approach compared to the estimates obtained using circuit level simulations, is 6.5% while providing an average speed up of 15000. We have demonstrated the scalability of our approach using designs from the ISCAS-85 benchmarks.

Increasing variability in device parameters not only affects the behavior of contemporary ICs, but also their vulnerability to transient error phenomenon, especially soft errors. Such variations in device parameters are caused by static process variations, dynamic variations in power supply and temperature values and slow degradation of individual devices due to phenomena like Hot Carrier Injection (HCI) and Negative Bias Temperature Instability (NBTI). In the next part of our work, we analyze the impact of such variations on the Soft Error Rates (SER) of combinational logic circuits. Tools that model threshold degradation of NMOS devices due to HCI and PMOS devices due to NBTI in logic circuits were built based on existing methodologies. Results were obtained for custom designed circuits and ISCAS-85 benchmarks. A detailed analysis of the effects of threshold variations on SER is also presented with interesting observations. Finally, the results of experimental measurements of SER in different set-ups and different devices including a microcontroller and a microprocessor are presented in the Appendix. While the experiments on the microcontroller showed no errors, the microprocessor had many failures, thus clearly indicating the trend in soft errors with technology scaling. The results presented here show the increase in error rates with voltage scaling as well.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

My stay at Penn State has taught me many things and I am indebted to several people for the wonderful times I have had here. It will be impossible to acknowledge all who have contributed not only towards my thesis but also towards my life in general. I will mention just a handful of people here.

I am grateful and indebted to my thesis advisor Professor Vijaykrishnan Narayanan for the guidance and encouragement he has shown me throughout my stay at Penn State and also for the motivating discussions he enjoyed. During many phases when I had struggled, his belief in my capabilities and me exceeded my own. This motivated me to strive for better research. I would also like to thank Professor Kenneth Jenkins for accepting the task of being co-chair of my committee and also taking time from his busy schedule to give his feedback at various stages of my research.

I am also very grateful to Professor Mary Jane Irwin for providing invaluable suggestions throughout the course of my Masters as well as my Ph.D. It is truly amazing to see her grasp of even the most intricate details of different topics and provide some insightful pointers towards future work, which have provided direction for my research at Penn State.

I would also like to thank Professor Yuan Xie, who has always been available to give his suggestions on different research topics I have worked on. I was able to go to him for advice whenever Dr. Vijay was out of his office on travel or work. My work at the reactor would have never happened without the help of Professor Kenan Unlu. Apart from making it possible for me to perform soft error testing at the Penn State Nuclear reactor, he provided great suggestions on how to perform the tests effectively and safely. I would also like to thank the staff at the nuclear facility at Penn State for helping me in various ways with the accelerated soft error testing. I would also like to thank Professor Charles Croskey for agreeing to be part of my thesis committee and for providing very crucial suggestions towards my thesis organization and content, which has resulted in significant improvement to my thesis.

Being part of a large research group made my work easier, especially when I needed any help with regards to my research. While every student in the group has helped me at some point of time, I would like to give special thanks to Vijaysai, Jungsub and Ramki for helping me at very crucial stages of my research.

The five and a half years of my stay at Penn State have been memorable and I owe my thanks to innumerable friends some of whom include Aravind, Balaji, Juliet, Krishna, Kunal, Prashanth, Sachin, Shiva, Suresh, Vicky, Vidhyadhar, members of the Association for India's Development, Vedic Society and the Penn State Cricket Club. I would also like to thank my friends: Kalyan, Rajeswari, Sampath and Srinath for having been so supportive of whatever I have done and for keeping me motivated.

Finally, I would like to thank my parents and extended family members without whose help I would have never been able to make it at Penn State. I am indeed fortunate to have such a big circle of family and friends, which have kept me focused throughout my life.

Chapter 1

Introduction

Soft errors in combinational circuits are becoming as important as errors in memory elements. Continuous device scaling and increasing pipeline lengths contribute to the increase in soft error rates in data-path structures. The masking effects that prohibit the increase in soft error rates in logic circuits are not sufficient to prevent further rise in error rates.

Soft errors are temporary errors caused mainly because of external radiations. Such radiations directly or indirectly induce localized ionization capable of upsetting internal data states. The kinetic energy of the particles hitting silicon substrate generates electron hole pairs as they pass through the p-n junctions. These electron hole pairs generate short duration current pulses that cause soft errors. In memory circuits and latches, these errors are just a flip in the stored values, which result in temporary changes in the output of combinational circuits. In logic circuits, errors occur when the temporary changes in the output are latched [1], [2].

1.1 Sources of Radiation Induced Soft Errors

The energy particles that cause significant increase in the number of electron hole pairs include 1) alpha particles, 2) high energy cosmic ray induced neutrons and 3) neutron induced boron (^{10}B) fission. The first type consists of alpha particles emitted from packaging materials and from the interaction of cosmic ray thermal neutron with boron present in the P-type regions of the devices [3]. A single alpha particle can generate anywhere from 4 to 16fC/m over its entire range.

The second type of particles result from high-energy cosmic ray induced neutron flux and is strongly dependent on altitude. The intensity of the cosmic ray neutron flux increase with increase in altitude. The primary reaction by which cosmic ray induced neutrons cause SER is by silicon recoil. The impinging neutrons knock off the silicon from its lattice. The displaced silicon nucleus breaks down into smaller fragments each of which generates some charge. The charge density for silicon recoils is about 25 to 150fC/m, which is more than that from alpha particle strike. So cosmic ray neutrons have a higher potential to upset the circuit. Thus, currently, cosmic ray induced neutrons seem to be the major source of soft errors in circuits.

The third significant source of ionizing particles is from the neutron induced ^{10}B fission. ^{10}B, an isotope of p-type dopant (about 19.9%), is unstable and on impact from neutron, it absorbs the neutrons and breaks apart with the release of an alpha particle and ^{7}Li (Lithium). Both these byproducts are capable of inducing soft errors. To reduce SER due to alpha particle induced soft errors, one can use pure materials and shield the circuit so that components with higher alpha emission rates are physically isolated from the sensitive circuits. But such solutions are generally not effective against the neutrons as they are highly penetrative. The intensity of these neutron radiations depends on altitude, geomagnetic region and solar cycles [4]. Recent works [5], [6] and [7] have shown the effect of technology scaling on soft errors. In [4], a study on radiation flux noted that particles of lower energy occur far more frequently than particles of higher energy. Thus as CMOS device sizes decrease, they are more easily affected by these lower energy particles, potentially leading to a much higher rate of soft errors.

In sum, these high energy particles striking the silicon substrate generate electron hole pairs as they pass through the p-n junctions. These electron hole pairs generate short duration current pulses that cause soft errors. In memory circuits and latches, these errors flip the stored values while they cause transient glitches at the output of combinational circuits. In combinational circuits, errors occur when these transient glitches are latched by state elements [8].

1.2 Soft Errors in Logic Circuits

As mentioned earlier, memory elements are the first to be affected by such radiations as these memory elements store logic values either in capacitors or as feedback circuits that could be easily upset. Errors resulting from direct strikes on state elements have been the focus of research for many years [6], [9]. As a result of reducing pipeline depth and downscaling of nodal capacitance and supply voltages, radiation induced soft errors in combinational logic is gaining increasing attention. These logic errors are expected to become as important as directly induced errors on state elements [10]. Thus a detailed evaluation and modeling of soft errors in various components of data paths is essential.

In combinational circuits, there are three inherent masking mechanisms that prevent the propagation of any given pulse along a path towards the input of a state element. The first mechanism is logical masking, which is determined by the logic structure, and directly depends on the input patterns. The second mechanism is electrical masking, a function of the inherent delays of logic gates in the circuit. Finally, latching-window or time-window masking is a function of the flip-flop set-up and hold time.

The masking effects can be better understood with the help of Figure 1.1, which shows an example of a pipeline stage having a NAND based structure with their states. Assuming a particle strike causes a negative pulse at node B, the inherent delay (i.e. the finite bandwidth) in the gates before the next set of registers result in the progressive attenuation of the pulses as shown in the figure. This represents electrical masking where the pulse is much smaller when it reaches O1. Also, note that the pulse at B does not cause any change on node E. This is due to the logical property of the NAND gate preventing any pulse propagation and hence called logical masking. Latching-window or time-window masking is also illustrated in Figure 1.1. The pulse should occur within the set-up and hold time of the latching element to cause an error. Pulses that occur outside this window, as in two of the cases illustrated, do not result in a soft error.

Although the three masking mechanisms have been the reasons for fewer errors in logic circuits, with scaling, the effect of both electrical and latch-window masking is reduced. This is due to lower supply voltages and capacitances, higher frequencies and decreasing pipeline depths. In fact the number of errors in logic circuits is predicted to overtake the number of errors in unprotected memory [10]. Further, our results from experimental measurements of soft errors on various devices show an increased number of failures with scaling. Thus it has become important to analyze and to model soft errors in combinational logic circuits.

In this work, we first begin with a circuit level analysis of soft errors in common data path elements. Here, at first, a detailed analysis of threshold variation effects on Soft Error Rates (SER) in flip-flops is presented. Next, an analysis of SER in adder circuits is presented. A detailed analysis of the various masking effects in combinational circuits is studied in the case of prefix adders. Further, the effect of voltage and frequency scaling is studied and the results are presented. Next a couple of techniques to improve the SER in adder circuits, based on the results are presented. The methodology used to model soft errors in detailed circuit analysis is not scalable for predicting soft errors in larger logic blocks. This has necessitated techniques to model and hence predict soft errors in larger data path circuits efficiently and accurately.

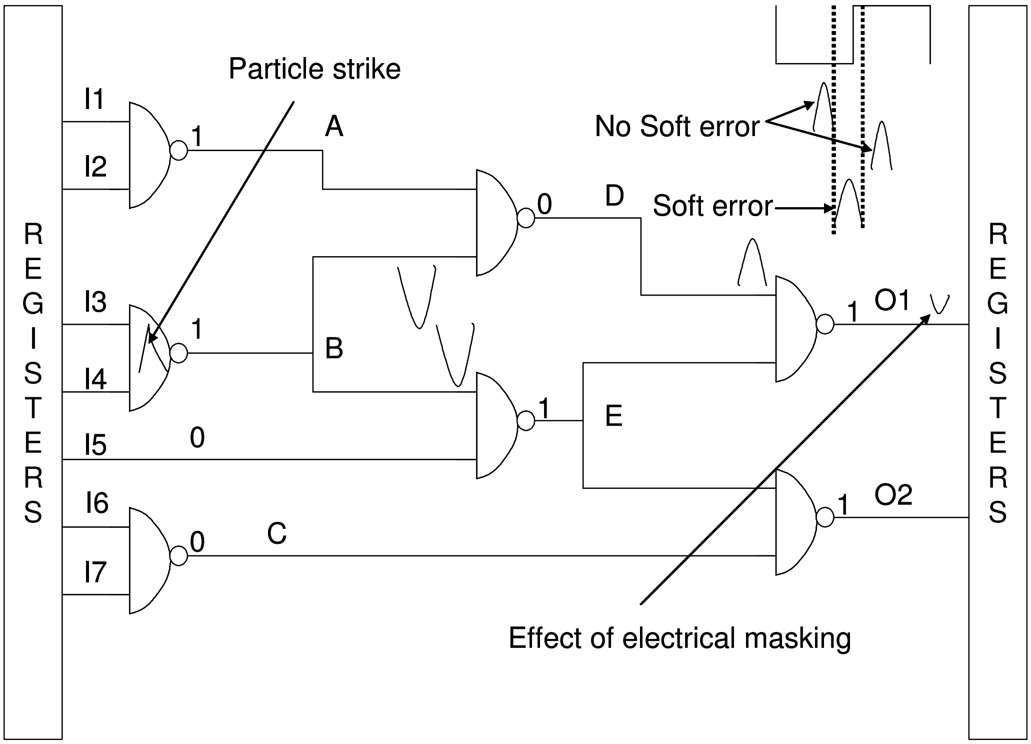


Figure 1.1: An example pipeline stage

Recently, there have been various approaches to estimate SER in logic circuits efficiently [11], [12], [13], [14], [15], [16], [17] and [18]. In this work, a new approach to estimate SER of logic circuits is presented. This method attempts to capture electrical, logic and latch window masking concurrently. The approach is applied to designs that use cell libraries characterized for soft error analysis, and utilizes analytical equations to model the propagation of a current pulse to the input of a state element.

Soft error is only one of the major sources of reliability problems that are affecting modern day circuits. These reliability problems are caused by the increasing variability in device parameters, which not only affect the performance and power of circuits but also their vulnerability to soft errors. Such variations in device parameters are caused by static process variations, dynamic variations in power supply and temperature values, and slow degradation of individual devices due to phenomena like Hot Carrier Injection (HCI) and Negative Bias Temperature Instability (NBTI). Here, the impact of such variations on the SER of combinational logic circuits is analyzed. Since the analysis involves changing individual parameters, detailed circuit level simulations were used. A detailed analysis of effect of threshold variations on SER is also presented with interesting observations.

1.3 Organization of the Thesis

The remainder of this thesis is organized as follows. In Chapter 2, a detailed analysis of data path components for soft errors is presented along with results for a few optimization techniques. Chapter 3 presents the Soft Error Analysis Tool for Logic

Analysis (SEAT-LA), a fast and accurate tool to measure SER in logic circuits. Chapter 4 presents the results of effects of other reliability issues and variations on soft errors. Finally, Chapter 5 concludes the thesis summarizing the results and contributions of this thesis.

Appendix 1 presents the experimental measurement of soft errors in different devices, emphasizing the importance of technology scaling from the results presented. Additionally, Appendix 2 provides additional information on the details of different designs used for soft error evaluation in this thesis.

Chapter 2

Soft Error Analysis in Data-path Components

In this chapter, analyses of two important components of data paths are presented. First, a brief review of the preliminary work done on different flop-flop designs is given after presenting a methodology to measure SER at the circuit level. Next, the effects of changing the threshold voltages on the soft error vulnerability of different flip-flop designs are presented. Different adder circuits are also analyzed for soft error rates in terms of critical charge ($Q_{critical}$) and the results are presented here. A few optimization techniques are proposed, based on the above analyses in each case.

2.1 Measuring SER

Soft errors occur when the collected energy Q at a particular node is greater than a critical charge $Q_{critical}$, which results in a bit flip at that node. This concept of critical charge is used for the estimation of soft error rate (SER) in this chapter. According to the model proposed in [6] (Eq. 2.1):

$$SER \propto Nflux \times CS \times \exp\left(-\frac{Q_{critical}}{Q_s}\right) \qquad 2.1$$

Where Nflux is the intensity of the Neutron Flux, CS is the area of the cross section of the node and Q_s is the charge collection efficiency which strongly depends on doping. $Q_{critical}$ is proportional to the node capacitance and the supply voltage. Thus, $Q_{critical}$ can be defined as the minimum charge collected due to a particle strike that can cause a soft error. If the charge generated by a particle strike at a node is more than $Q_{critical}$, the generated pulse is latched on, resulting in a bit flip. This concept of critical charge is generally used to estimate the sensitivity of SER in memories. The value of $Q_{critical}$ can be found by measuring the current required to flip a memory cell and can then be derived using Eq. 2.1.

The particle strike itself is modeled as a piece-wise linear current waveform, where the waveform's peak accounts for funneling a charge collection and the waveform's tail accounts for a diffusion charge collection. By changing the magnitude of the peak of the waveform and appropriately scaling the waveform, we try to find the minimum height for which the wrong value is stored in the memory element including flip-flops. A similar approach has been used in prior work [7]. However, a transient change in the value of a logic circuit does not affect the results of a computation unless it is captured in a memory element like a flip-flop. Therefore, to measure $Q_{critical}$ of a combinational logic, we inject a current pulse and try to latch the wrong value at the output of the logic chain. Logical masking, electrical masking and latching-window masking can mask a logic error. Such masking reduces the de-rating effects of soft errors. Thus another terminology called the timing window (*tw*) is introduced. This timing window can be defined as the fraction of the clock period for which a given particle strikes at a node would cause a bit flip to occur at the flip-flop connected to the output of the logic circuit. Timing window (*tw*) captures both electrical and time-window masking effects in logic circuits. Similar definitions have been used in recent works as well [14], [17].

While analyzing data path circuits in this work, we inject the current pulse only to those nodes that produce the change in the output, thus ignoring logical masking, as the effect of logical masking does not remain the same across different technologies. For flip-flops, the internal nodes, where the logical value is stored as charge, were chosen. For logic circuits the input nodes were chosen to inject the current pulse.

2.2 Soft Errors in Flip-Flops

2.2.1 Review of SER Analysis in Flip-Flops

In [19], we have presented a detailed analysis of soft errors in flip-flops along with the implementation of a couple of optimization techniques. Here a brief review of the work is presented. Flip-flops, important component of pipelined architectures, are becoming more susceptible to soft errors. Thus, this work analyzed soft error rates on a variety of flip-flops. The flip-flop designs studied here encompass many types of designs which include master-slave, pulse triggered, sense amplifier based flip-flops and two scannable latches. These designs were implemented and simulated in HSPICE [24], using the 70 nm Predictive Technology Model (PTM) [23].

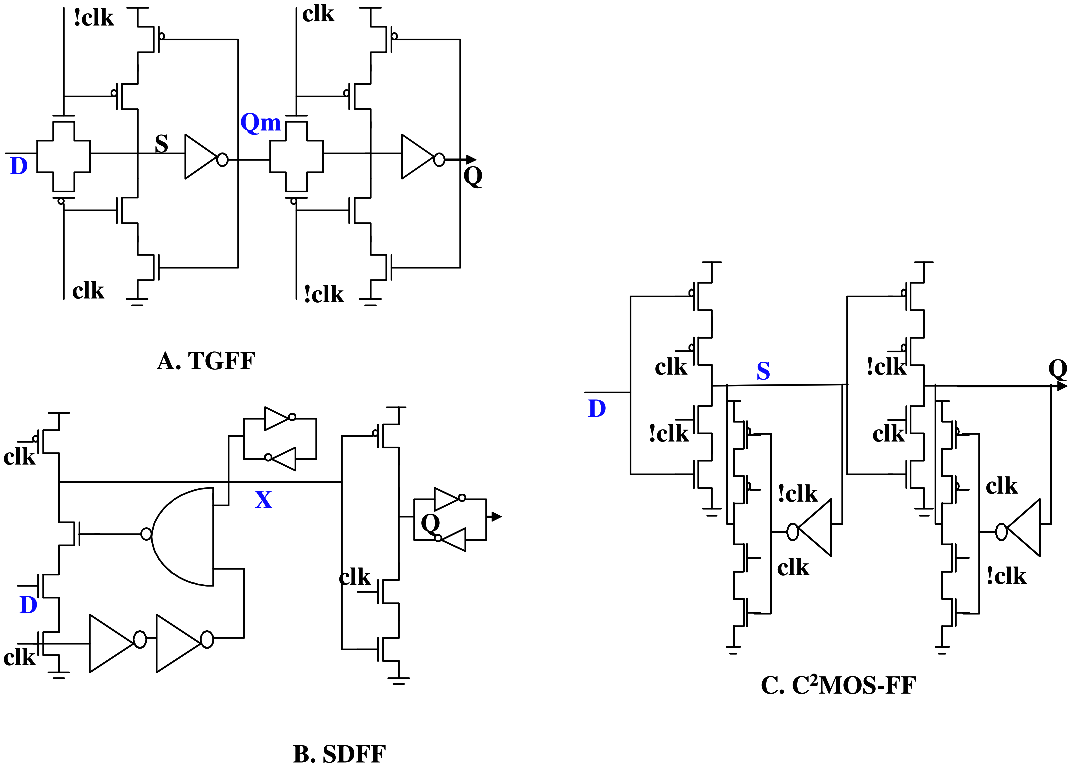
First, the critical charge for the susceptible nodes in each design was evaluated. From the results, it was observed that, in general, the pulse triggered flip-flops had a higher $Q_{critical,}$ while some designs, like the semi-dynamic and sense amplifier based flip-flops, had a higher $Q_{critical}$ for one type of bit flip. Also, the double edge triggered design was found to have the least $Q_{critical}$ of all the designs considered. Further, two hardening techniques were implemented and these results were presented. One attempts to increase the critical charge by increasing the gate capacitance, while the other improves the overall robustness of the circuit by replicating the master stage of the master slave flip-flops, which leads to reduced power and area overhead.

2.2.2 Effect of Threshold Voltage on Flip-Flop SER

We characterize three different flip-flops: the transmission gate flip flop (TGFF), the semi-dynamic flip-flop (SDFF) and the C^2MOS flip-flop (C^2MOS), for estimating the effect of increasing threshold voltages (V_{th}) on $Q_{critical}$. Please refer to Figure 2.1 for detailed schematics of these designs. There are two different effects of the change in threshold voltages on flip-flops. Firstly, the soft error rate of the flip-flop itself could change. This is found by evaluating $Q_{critical}$ at the most susceptible node [19]. Secondly the ability of the flip-flop to latch onto an error at the input could change. This effect will be useful in analyzing its behavior in a data path.

Since we focus on data paths, we list the $Q_{critical}$, at the input of flip-flop in Table 2.1. The results for 1 to 0 transitions for the node S and the input D, for TGFF are presented here. The $Q_{critical}$ of the internal node S remains almost constant. In fact there is a slight reduction in the value, which is not evident from the values given in the table. There are two factors that could affect the change in $Q_{critical}$. The gain of the inverter, for which the node S is an input, increases with an increase in threshold voltage [20]. This should result in a significant reduction in $Q_{critical}$. Also the transmission gate present at the slave stage would lead to a greater $Q_{critical}$ for the node as the threshold voltages increase.

These two factors effectively cancel out each other; hence the $Q_{critical}$ remains almost constant. At the input D, the presence of the transmission gate results in a large increase in $Q_{critical}$. Similar simulations were done on a C^2MOS flip-flop, which also has master-slave stages similar to that of the transmission gate flip-flop. In this case, since there is no inverter in the path to the output, $Q_{critical}$ increases for both the nodes S and D.



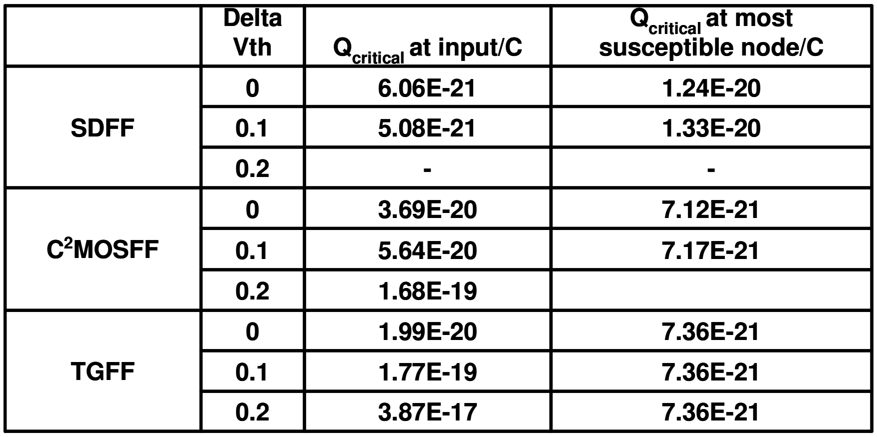
A. TGFF

B. SDFF

C. C²MOS-FF

Figure 2.1: Flip-Flop designs

Table 2.1: $Q_{critical}$ for different Flip-Flops



	Delta Vth	$Q_{critical}$ at input/C	$Q_{critical}$ at most susceptible node/C
SDFF	0	6.06E-21	1.24E-20
	0.1	5.08E-21	1.33E-20
	0.2	-	-
C^2MOSFF	0	3.69E-20	7.12E-21
	0.1	5.64E-20	7.17E-21
	0.2	1.68E-19	
TGFF	0	1.99E-20	7.36E-21
	0.1	1.77E-19	7.36E-21
	0.2	3.87E-17	7.36E-21

One of the pulse triggered designs, a semi-dynamic flip-flop, is also tested for changes in $Q_{critical}$. This design has few large sized devices resulting in a much higher $Q_{critical}$. Here, the node X is the most susceptible node. Since this node feeds back into a NAND gate, when the threshold increases, due to the increase in delay of the NAND gate and two inverters, $Q_{critical}$ also increases. Thus the flip-flop by itself has a higher $Q_{critical}$ as threshold voltage increases. At the input the greater overlap time helps pull down voltage at node X more and thus reduces the $Q_{critical}$. When V_{th} increases by 0.2V, the flip-flop fails to latch the input data. Therefore, $Q_{critical}$ is not listed in the table.

What is to be noted from the results is that the $Q_{critical}$ at the input nodes of flip-flops increased rapidly with V_{th}. This rapid increase in $Q_{critical}$ offsets the possible reduction in $Q_{critical}$, due to the increased gain of static logic gates with increase in V_{th}. In sum, in data paths with unbalanced delays, delay balancing can be done by increasing the threshold voltage of the paths with smaller delays including that of the flip-flops in the path. This technique results in reduced leakage in these paths and also increases the robustness of the paths [20].

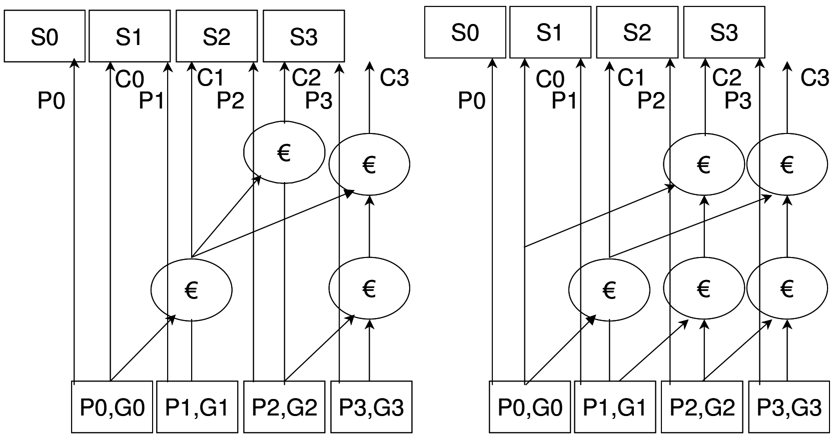
2.3 Evaluation of SER in Adders

Adder circuits are an integral part of data path circuits. A detailed analysis and comparison of soft errors will give a clearer insight on the tradeoffs in selecting the adder structures for fault tolerance. Here, two of the common prefix adders, which are the most commonly used adders in present day designs, are analyzed for soft errors. The prefix adders are analyzed for soft errors to find the $Q_{critical}$ and timing window (*tw*) values. The Brent-Kung (B-K) [21] and Kogge-Stone (K-S) [22] adder structures used here for $Q_{critical}$ analysis are shown in Figure 2.2. The P and G represent “Propagate” and “Generate” signals, which are functions of the adder inputs. The € cell outputs the “Block Propagate” and “Block Generate” values to be used by the other € cells and also to generate the carry. S are the sum values obtained from propagate and carry-in into the bit. Intuitively, we can see that the K-S adder has a greater number of cells and shortest path to the carry. On the other hand, B-K tends to have a higher fan-out and could potentially have more multi-bit errors. These adders are compared with a ripple carry adder here.

In this work, we start with analyzing for possible multiple bit errors caused by single particle strikes at different nodes for a B-K adder and present the results. Next we compare the B-K adder and the K-S adder to Ripple Carry (RC) adder with respect to their $Q_{critical}$ values. All simulations were done with 1V supply voltage with the adders working at 0.5GHz. Next, we analyze the effect of voltage and frequency scaling on the $Q_{critical}$ and timing window (*tw*) and present the results here. Voltage scaling simulations were done at 1 V, 0.8 V, and 0.6 V while the frequencies used in frequency scaling were

1, 0.75 and 0.5 GHz. Then, we run the adder circuits at the maximum frequencies in our voltage scaling simulations.

Based on the above results, we try to improve either the $Q_{critical}$ or the timing window of the adder circuits. To increase the timing window, we use a Semi-Dynamic Flip-Flop (SDFF) instead of the Transmission Gate Flip-Flop (TGFF) that we used in our initial simulations. Since the SDFF has a shorter set-up time than a TGFF, the timing window should decrease. It has been shown that increasing the threshold voltage increases $Q_{critical}$. Thus, we simulate the B-K adder to observe the $Q_{critical}$ and *tw* with increases in threshold voltages. All circuits in our simulations were custom designed and laid out in 70 nm Predictive technology model [23] and simulated using HSPICE [24]. For better abstraction, the nodes presented in our analysis are those in which a particle hit here would affect the whole carry chain and are outputs or inputs of the basic cells.



(a) Brent-kung

(b) Kogge-stone

Figure 2.2: Four-bit Prefix Adder Structures

2.3.1 Comparison of Adder Structures

Figure 2.3 shows the $Q_{critical}$ (Log scale) values for the three types of adders. First we analyze the B-K adder and compare the values evaluated at various nodes. As clearly seen from the results, the $Q_{critical}$s for a node to cause a flip at all the sum outputs are similar. For example it shows that the $Q_{critical}$ values for a pulse at the node C0 affecting the outputs S1, S2 & S3 are comparable. Thus the node C0 fans out almost equally to the sum outputs. However, the $Q_{critical}$ for all above outputs flipping together is much higher. This is due to the difference in the time at which the particle strike has to occur at C0 to cause a flip in each output in the same clock cycle. This concept is better explained using Figure 2.4.

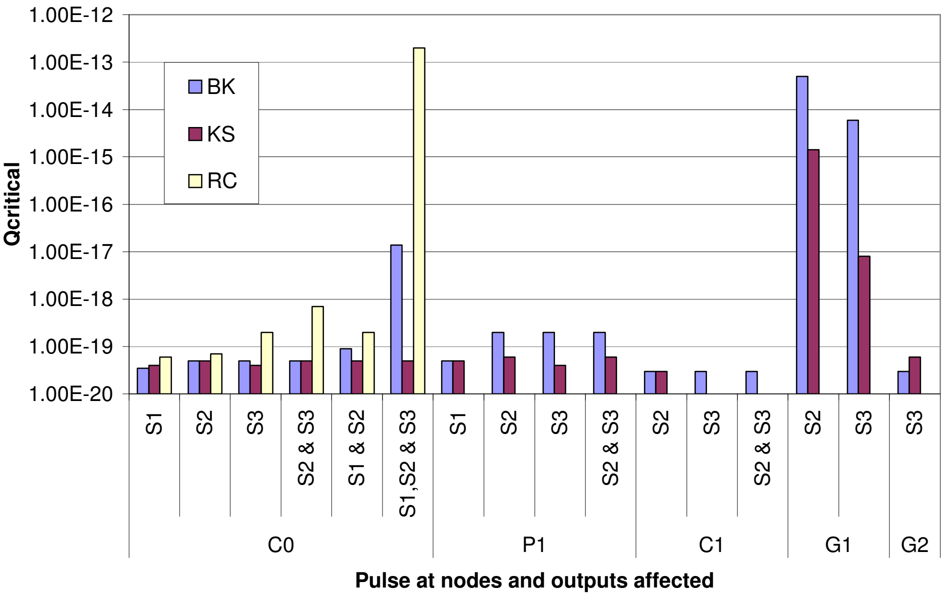


Figure 2.3: Comparison of B-K, K-S and RC Adders

Figure 2.4 shows that the same current pulse caused by a particle strike (at node C0) at different points of time induces flip in different outputs. But to flip all the outputs together, a much larger pulse is required, as shown. From this we could infer some interesting facts. Firstly, due to difference in paths to the various outputs, the time at which the pulse needs to occur at a node to cause a bit flip (and also the timing window) may be different even if the $Q_{critical}$s are comparable. Another subtle fact to be noted, demonstrated by the increased $Q_{critical}$ for multi-bit errors, is that as the collected charge (Q_{coll}) at the node increases the timing window of the node also increases, since there is a greater chance for that pulse to be latched on at the outputs. Another interesting trend was noted in some gates closer to the output latch. Here, capacitance effects mitigate the flipping nature of the output, thus increasing the $Q_{critical}$ of the nodes. One example is pulses occurring at node C0 directly affected the output S1, hence increasing the $Q_{critical}$ for some input patterns. Since these capacitance effects are functions of input patterns, they could be modeled as special cases of logical masking.

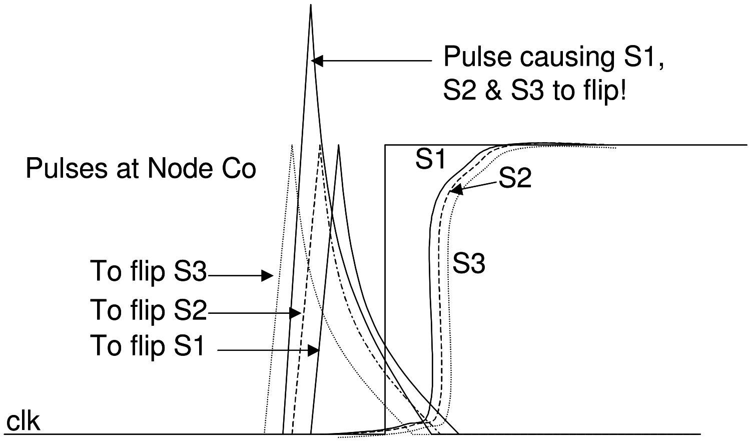


Figure 2.4: Multi-bit errors caused due to pulse at C0

In Figure 2.3, B-K, K-S and RC adders are compared with respect to their $Q_{critical}$ at the node C0. In the B-K and K-S, the $Q_{critical}$ required to flip the different sum outputs are comparable. The RC on the other hand intuitively has a progressively increasing $Q_{critical}$ as shown. Also, the K-S adder has comparatively lower $Q_{critical}$ values. This is because of the short carry chains in the case of K-S adders. But the K-S adder has comparatively greater area susceptible to soft errors due to the larger number of cells. However, because of the structural property of the adders there are fewer multi-bit errors occurring in a few nodes in the K-S adders (Node C1 in Figure 3). Thus the B-K adder, which has more nodes that fan out equally to many outputs, could have a greater number of nodes that cause multiple-bit errors on a single particle strike.

2.3.2 Effect of Voltage and Frequency Scaling

Figure 2.5 shows the result of scaling both voltage and speed together in a B-K adder. Here, the adder was run at 1 GHz, 0.833GHz and 0.5 GHz with 1V, 0.8 V and 0.6V as supply voltages, respectively. The adders were run at maximum possible speeds at each supply voltage. As seen from the results, for most of the nodes as the voltage and frequency are scaled, $Q_{critical}$ reduces slightly. Thus, the $Q_{critical}$ reduction is not drastic with supply scaling as expected. Next, we kept the frequency at 0.5 GHz and increased the voltage from 0.6 V to 1 V and observed the change in $Q_{critical}$ and the timing window, *tw*.

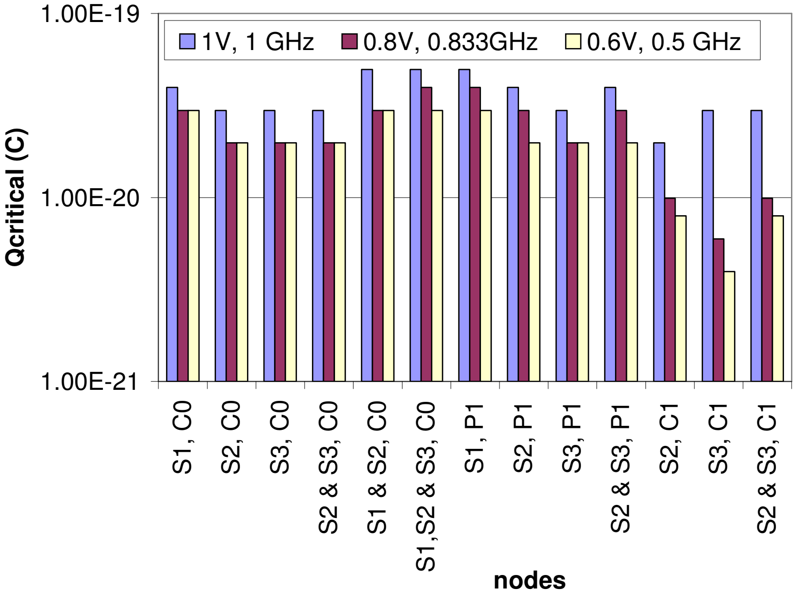


Figure 2.5: Effect of Voltage and Frequency Scaling

The effect of voltage scaling on the $Q_{critical}$ is studied in Figure 2.6. Clearly, for most of the nodes, $Q_{critical}$ increases by a small amount as voltage increases, but for some nodes the increase is more evident (Node C0 in Figure 2.6). This is because, the ripple effect, which results in a glitch, covers a larger portion of the clock at the smaller voltage. This overlaps partly with the set-up and hold time of the flip-flop and a smaller current pulse (which adds on to the glitch), and hence a smaller $Q_{critical}$ is needed to flip the output for lower voltages. Also, sometimes, especially in the case of multiple bit errors, $Q_{critical}$ decreases as voltage increases. This might be due to the change in set-up and hold time constraints of the flip-flop, resulting in the change in the timing window of the outputs. For example, for a pulse at C0 to flip all of S1, S2 and S3 together, $Q_{critical}$ decreases as the voltage is increased from 0.8 V to 1 V. This is because there is less overlap time between the timing windows of each output node for a pulse at C0.

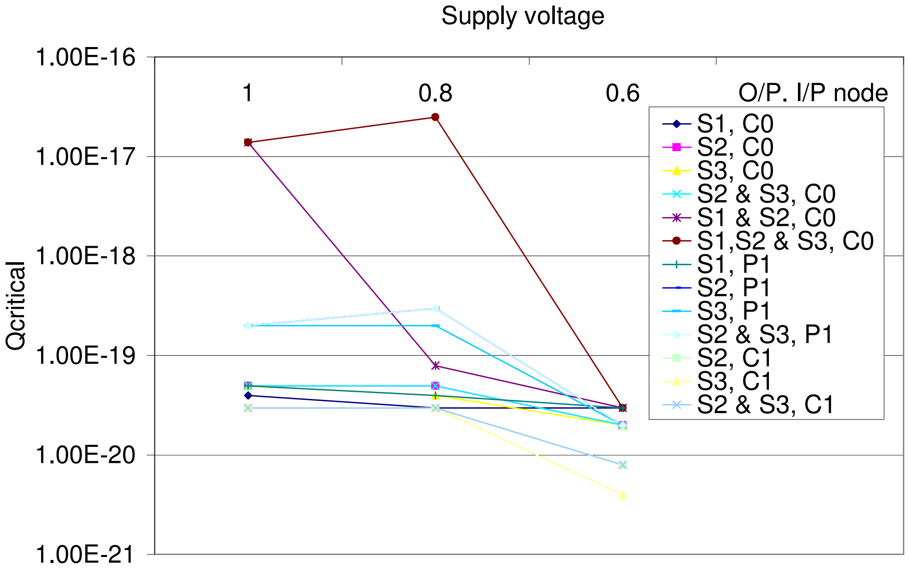


Figure 2.6: Effect of scaling voltage on $Q_{critical}$

Figure 2.7 shows the effect of voltage scaling on the timing window, *tw*. Clearly, the timing window also reduces as voltage is increased. Thus, apart from an increase in the $Q_{critical}$ values, as voltage increases, there is reduction in the value of *tw*, which will reduce the soft error rate significantly.

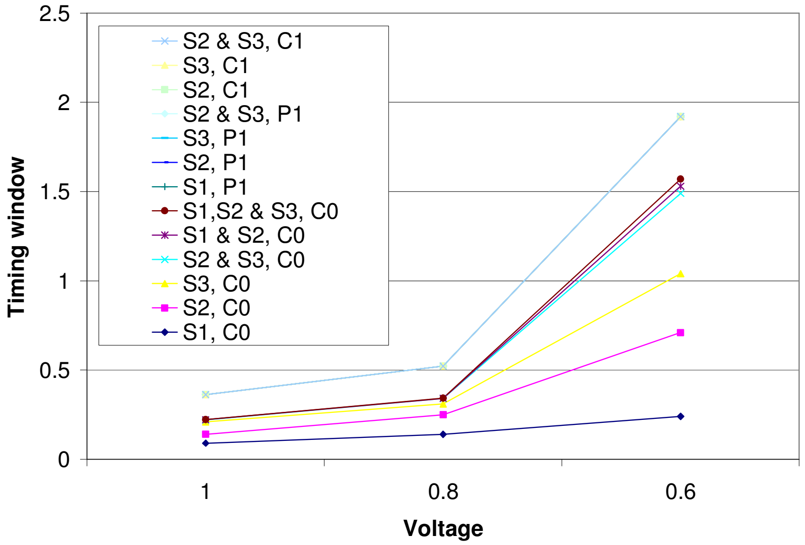


Figure 2.7: Effect of scaling voltage on *tw*

Now, at a supply voltage of 1V, a separate simulation was conducted in which the frequency was increased from 0.5 GHz to 1 GHz. Figure 2.8 presents the result of increasing frequency on $Q_{critical}$ values. In most cases a very slight decrease in $Q_{critical}$ with increase speed is noticed. But in some cases (Node C0), the $Q_{critical}$ values decreases drastically as speed increases. This is because the glitches formed due to the ripples created in the adder occupy a larger portion of the clock cycle with increased speed and hence only a small pulse is required to aid this glitch to flip an output. Also, the timing window increases as the speed scales, which is intuitive since set-up and hold time become a bigger percentage of the clock period with increase in speed. Thus by analyzing the voltage and frequency scaling separately, we can see that either increasing voltage or decreasing speed will result in a higher $Q_{critical}$ and a shorter timing window, thus making the circuits more robust. Also, scaling down both frequency and voltage at the same time does not affect the soft error rate as much; however, a small reduction in $Q_{critical}$ was observed.

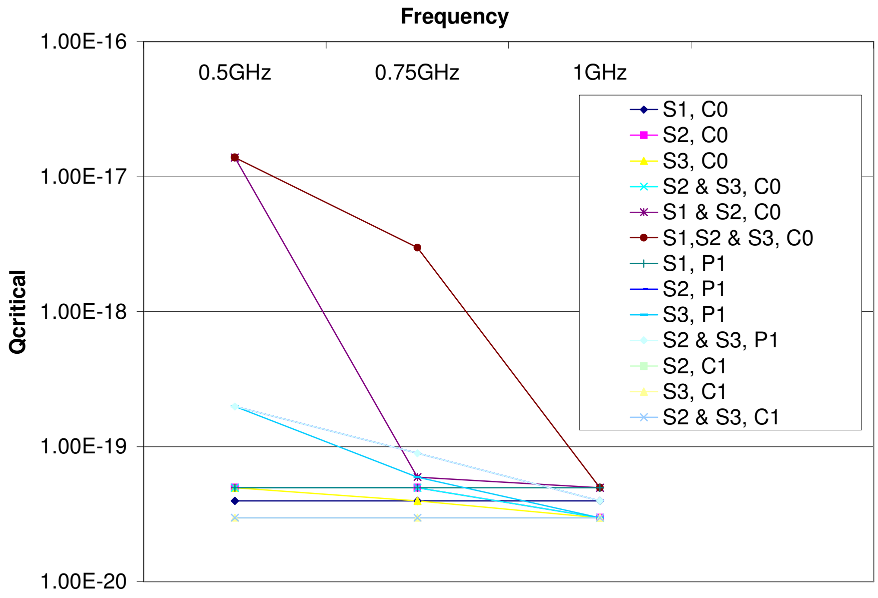


Figure 2.8: Effect of scaling frequency on $Q_{critical}$

2.3.3 Changing *tw* and $Q_{critical}$

Apart from the above simulations, other simulations were performed to improve either the timing window or $Q_{critical}$ of the B-K adder. Firstly, a different flip-flop, the Semi-Dynamic Flip-Flop (SDFF), having smaller set-up time, was used to measure both $Q_{critical}$ and *tw*. When compared to a transmission gate flip-flop initially used for simulations, the $Q_{critical}$ did not change much for most of the nodes and increased slightly for multiple bit errors. More importantly, the timing window of the adder reduced, especially for multiple bit errors as shown in Figure 2.9.

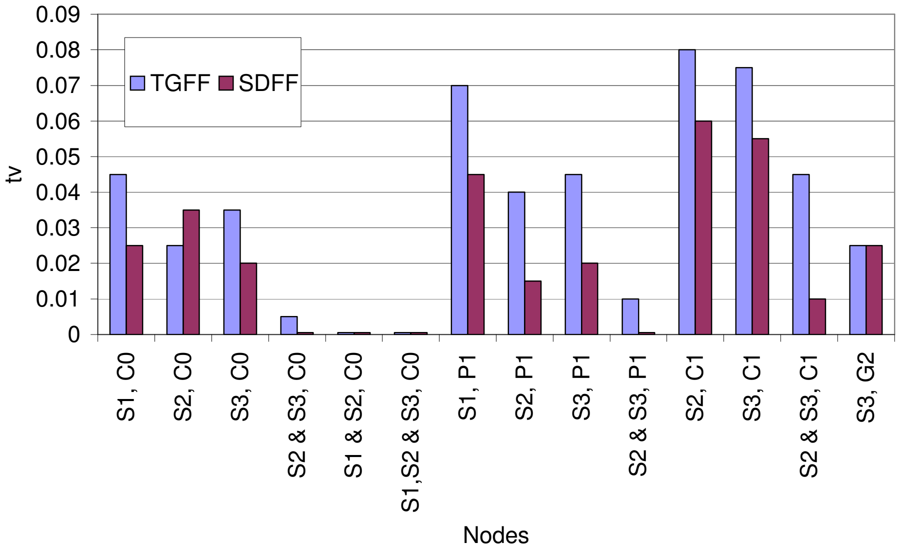


Figure 2.9: Comparing *tw* of TGFF and SDFF

Secondly, the threshold voltage of the circuit along with the flip-flops was increased to see the effect on both $Q_{critical}$ and *tw*. Though, as expected, there was a marginal increase in the $Q_{critical}$, the effect on the timing window was more pronounced. As shown in Figure 2.10, the timing window of the circuit increased for almost every node as the threshold voltage was increased.

To summarize, in this chapter, detailed circuit-level analyses of different components of data-paths and the effect of different optimization techniques were presented. The insights obtained from the results presented here could be used for making important design decisions while choosing architectures and optimization techniques for a given application. The accurate circuit techniques used in this chapter to measure SER cannot be used to model SER in larger data path circuits, as the procedure is not very scalable. Thus, a faster tool that could model SER for larger circuits is necessary for predicting soft errors in logic circuits. In the next chapter, such a tool built on a new methodology is proposed and presented. This tool is verified using the HSPICE circuit simulation and is shown to be scalable for larger benchmarks circuits.

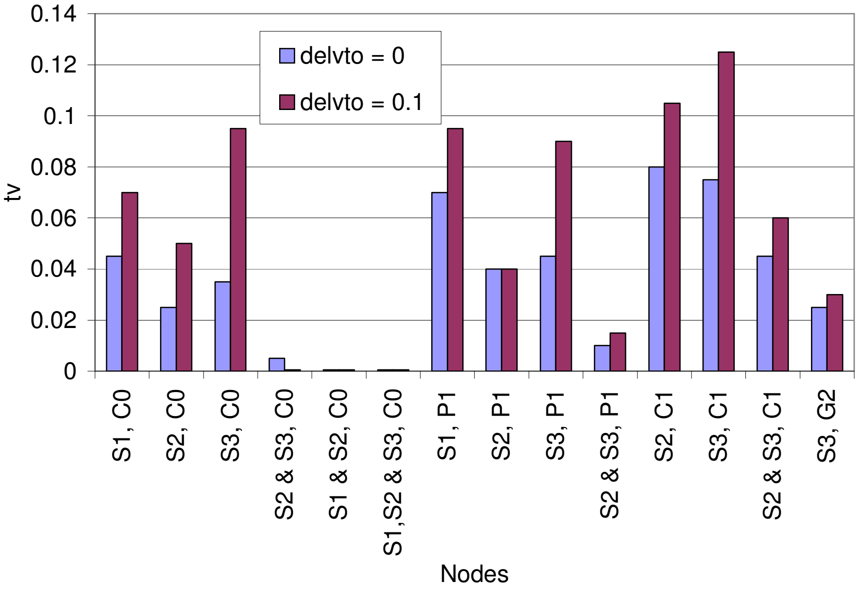


Figure 2.10: Effect of increasing threshold voltage on *tw*

Chapter 3

SEAT-LA: A Soft Error Analysis Tool for Logic Analysis

In this chapter, we propose a new approach to estimate SER for logic circuits that attempts to capture concurrently the three masking effects. A tool known as SEAT-LA (Soft Error Analysis Tool – Logic Analyzer), has been developed using the above methodology. This tool is a part of the hierarchical Soft Error Analysis Toolset (SEAT) that supports different levels of abstractions. The tool is verified for accuracy with simulations using HSPICE, with the assumption that HSPICE is the most accurate standard available.

The rest of the chapter is organized as follows. In Section 3.1, a basic introduction to soft errors in logic circuits and related works is presented. Section 3.2 describes the methodology used in SEAT-LA for soft error analysis. The tool implementation is discussed in Section 3.3. Section 3.4 presents verification of the tool by HSPICE and soft error estimates for some ISCAS circuits.

3.1 Modeling Soft Errors in Logic Circuits

In logic circuits, as mentioned earlier, there are three inherent masking mechanisms that prevent the propagation of any given pulse along a path towards the input of a state element. Logical, electrical, and latching window or time window masking are the three masking effects that are to be modeled for SER estimation in logic circuits. Recent works have proposed different methods to model the above masking effects [10], [5], [12], [13], [14], [15], [16], [17] and [18]. In [10] electrical masking is categorized into two further effects: an increase in rise and fall time and delay degradation. It models these two categories separately and then combines them to model electrical masking. However, they do not verify their methodology using device or circuit level analysis results. The SERA tool [13] combines probability theory, circuit simulation, graph theory, and fault simulation to estimate SER. This tool characterizes inverter chains and extends the results to all gates, which can result in inaccuracies.

In contrast, our approach characterizes each cell in the library. In [14], a mathematical model, based on set-up and hold time, was used for timing window (*tw*) estimation, while the electrical masking effect was determined using noise rejection curves on various gates. Instead of noise rejection, we actually model the transfer of a glitch across the combinational logic. The tool ASERTA used in [17] models soft errors by modeling electrical masking using mathematical equations for pulse propagation. However, they do not consider the effect of pulse heights on the electrical masking as we do in our work. Also, our approach to re-convergent nodes is very different from the methodologies presented above. Thus, our approach is unique from all the previous approaches that have been proposed to date [10], [5], [12], [13], [14], [15], [16], [17] and [18]. There have been various works on glitch modeling for power consumption [8], [25]. These models, however, do not calculate all glitch properties such as pulse amplitude and width required for modeling SER. A mathematical expression to model glitch amplitudes as it propagates through logic gates is proposed in [26]. In this work, we propose a mathematical expression for output amplitude based on geometrical calculation on approximated input pulse widths. We also use a mathematical expression for the output pulse width based on approximating the output voltage to a trapezoidal or a triangular pulse.

3.2 Characterization and Methodology

3.2.1 Logic Cell Characterization

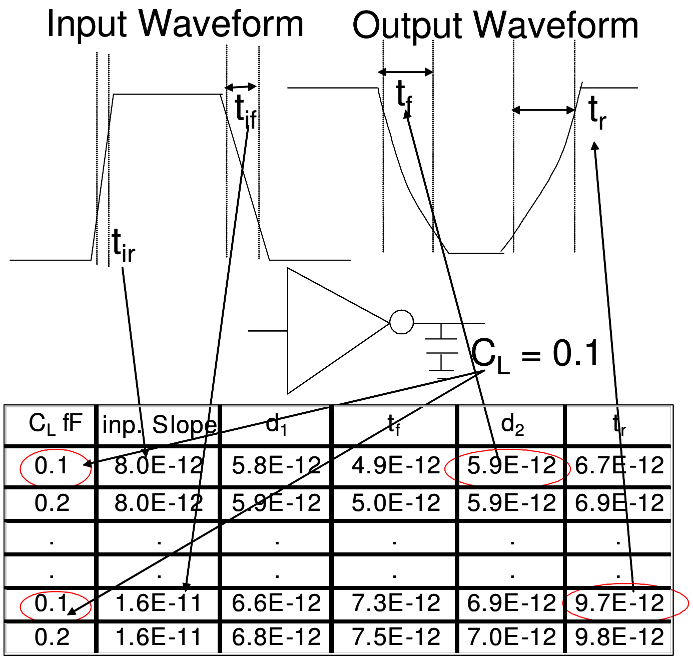
Our methodology assumes that a soft error upset is modeled by the injection of a current pulse. The first part of our characterization involves capturing the current-voltage transfer characteristics for the logic cells in our library for different current pulses occurring at the input nodes. The output of this characteristic table provides the output voltage pulse parameters that include the pulse magnitude, width, and the rise and fall times of the output voltage pulse. The characterization is performed for different input and output capacitances and current pulses. The device level version of SEAT was used to determine the dominant type of current pulses for which this characterization needs to be done. One disadvantage of this approach is that for a large library, the number of tables will also be large.

The second pre-characterization involves delay characterization for all the cells in our library. This characterization may already be available for the target cell library. Our analytical models for calculating glitch amplitude and width propagation require these delay values. It is well known that the delay of a gate is a function of both the slope and the load capacitance. Hence, in our work, we have characterized the delay and slopes of the output for each of the basic cells for different input slopes and for different input combinations by varying the load capacitance.

Figure 3.1 shows a portion of the delay characterization table for an inverter. Thus, for each load capacitance (C_L) and input slope, the delays (d_1 for 1-> 0 transition and d_2 for 0-> 1 transition of the output) and the output slopes (rise time t_r and fall time t_f of the output) are listed. As shown in Figure 3.1, the output delay and slopes can be obtained by indexing, using the input slopes (t_{if} and t_{ir}) and C_L. An important note is that the delay and slope values are a function of the input states of various gates. For example, Table 3.1 shows the delay and slope values for two different states of a XOR gate. Consequently, it is necessary to know the input state of the gate for finding the delay in addition to the load capacitance and input slope.

3.2.2 Flip-Flop Characterization

This characterization is used to determine the *tw* of the flip-flops used in our designs. Our characterization involves sweeping a voltage pulse of a specific width and height at the input of the flip-flop and finding the time for which this pulse is latched by the flip-flop using HSPICE simulation. This time is then used to calculate the timing window (*tw*) by dividing this over the clock period that the circuit is operating at. This characterization is repeated for different pulse widths and heights. The fact that the height of the pulse is also used to characterize the *tw* makes this method more accurate. For example, two pulses of same width, but different heights, might have different *tw*s, making our approach more accurate than approximations such used in [17], which assume that only a pulse completely encapsulating the latching window can cause an error.



C_L fF	inp. Slope	d_1	t_f	d_2	t_r
0.1	8.0E-12	5.8E-12	4.9E-12	5.9E-12	6.7E-12
0.2	8.0E-12	5.9E-12	5.0E-12	5.9E-12	6.9E-12
.	.	.	.	.	.
.	.	.	.	.	.
0.1	1.6E-11	6.6E-12	7.3E-12	6.9E-12	9.7E-12
0.2	1.6E-11	6.8E-12	7.5E-12	7.0E-12	9.8E-12

Figure 3.1: Slope-Delay characteristics of an inverter

Table 3.1: Slope-Delay characteristics of XOR gate

Cload(fF)	inputslope	State a =0 , b 0 -> 1		State a =0 , b 1 -> 0	
		d_1	t_r	d_2	t_f
0.1	8.00E-13	2.68E-12	1.00E-11	1.63E-12	1.12E-11
0.2	8.00E-13	2.77E-12	1.03E-11	1.72E-12	1.14E-11
.	.	.	.	.	.
.	.	.	.	.	.
0.1	1.60E-12	2.68E-12	1.02E-11	1.66E-12	1.15E-11
0.2	1.60E-12	2.78E-12	1.05E-11	1.75E-12	1.18E-11

Figure 3.2 shows the variation of the timing window with increasing pulse width for a positive voltage pulse of magnitude of 1V for two different flip-flops, Transmission Gate (TGFF) and a Hybrid Latch Flip-Flop (HLFF). Clearly, although the variation is

linear at different portions, there are lower slopes for lower pulse widths and higher slopes at larger pulse widths. This difference will not be captured by the linear equations as represented in [10], [14]. Also, note that at very high pulse widths, it does not matter if either of the flip-flops is used, as the timing window for both is the same.

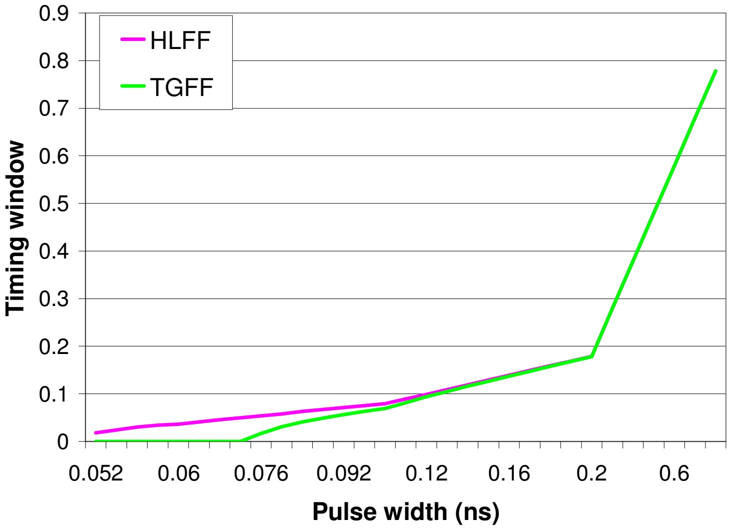


Figure 3.2: Flip-Flop characterization

3.2.3 Modeling Voltage Glitch Propagation

Next, we propose a set of mathematical equations assuming a triangular or trapezoidal pulse for determining how the voltage pulse amplitude and width vary as they propagate through logic gates towards the flip-flop input.

First, we focus on estimating the amplitude of the output voltage pulse given the input pulse width (PW_i) and the slopes of the output pulse (t_f and t_r). Assuming a linear output slope, for a 1->0 output pulse of an inverter, the minimum output voltage V_{omin} can be calculated as follows (Eq. 3.1):

$$Vo_{\min} = \begin{cases} 1 - \dfrac{PW_i}{t_f \times 1.25}, PWi < t_f \times 1.25 \\ 0, PW_i > t_f \times 1.25 \end{cases} \qquad 3.1$$

Here, t_f is the output fall time which was found from the delay and slope table discussed in section 3.2.1. t_f is the time required for the output changing from 90% to 10% of supply voltage, V_{dd}. Consequently, we use a scaling factor of 1.25 to mimic a complete swing from V_{dd} to zero. Similarly, for a 0->1 output pulse of an inverter, the maximum output voltage V_{omax} can be given as follows (Eq. 3.2):

$$Vo_{\max} = \begin{cases} \dfrac{PW_i}{t_r \times 1.25}, PWi < t_r \times 1.25 \\ V_{dd}, PW_i > t_r \times 1.25 \end{cases} \qquad 3.2$$

Where, t_r is the rise time of the output pulse (from 10% to 90% of V_{dd}). The accuracy of the model is directly dependent on the accuracies of the pulse width and the slope values from the pre-characterized tables. Figure 3.3 shows the verification of this model for a simple inverter where Vmin is the minimum output voltage observed using circuit simulation and Vmincalc is the voltage calculated using the above model. Next, we focus on estimating the width of the output voltage pulse. The width of the output pulse can be modeled as a function of delay of the gate. Approximating the output pulse to be a triangular pulse, we model the output pulse width (PW_o) using the following equation (Eq. 3.3):

$$\mathrm{PWo} = (\mathrm{PWi} - \mathrm{d}_1) + \mathrm{X} * \mathrm{d}_2 \qquad 3.3$$

Where the delays d_1 and d_2 are the first and second transition delays of the output waveform and as in Eq. 3.4

$$X = \frac{\left(\frac{V_{dd}}{2} - Vo_{\min}\right)}{\frac{V_{dd}}{2}} \qquad 3.4$$

for a 0 →1 input pulse and as in Eq. 3.5

$$X = \frac{\left(Vo_{\max} - \frac{V_{dd}}{2}\right)}{\frac{V_{dd}}{2}} \qquad 3.5$$

for a 1→0 input pulse

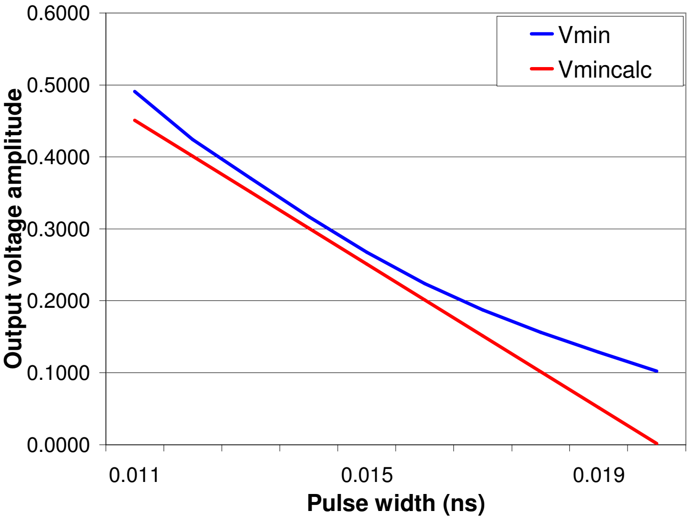


Figure 3.3: Calculated voltage amplitude for an inverter

This scaling using X is performed, because delays (d_1 and d_2) in the characterization table are determined at 50% switching point ($V_{dd}/2$) assuming full voltage swing, while actual voltages are swinging only to V_{omin} or V_{omax}. Figure 3.4 can be used to explain the value of X for positive input pulse. As can be seen the distance between the negative edge of input pulse and the positive edge of output pulse is not the delay d_2; rather, it is just a fraction of it given by the expression for X.

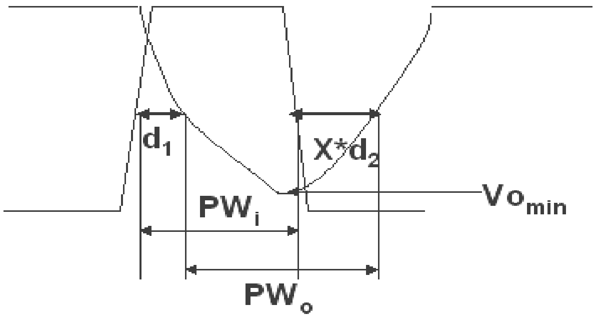


Figure 3.4: Modeling pulse propagation

The pulse width estimation method was verified with basic gates, using a trapezoidal pulse as input of varying pulse width. Figure 3.5 shows the results for a NAND gate where "pwidtho" is the observed pulse width from circuit simulation while "pwidthcalc" is the calculated pulse width. As can be seen, for larger pulse widths, the mathematical model matches circuit simulation results exactly. But for lower pulse widths, the model underestimates the output pulse widths. This does not affect the accuracy of our methodology significantly, because the timing window of smaller pulses is very close to zero. Hence, the equations proposed above can accurately model the pulse propagation, provided the characteristics are accurate.

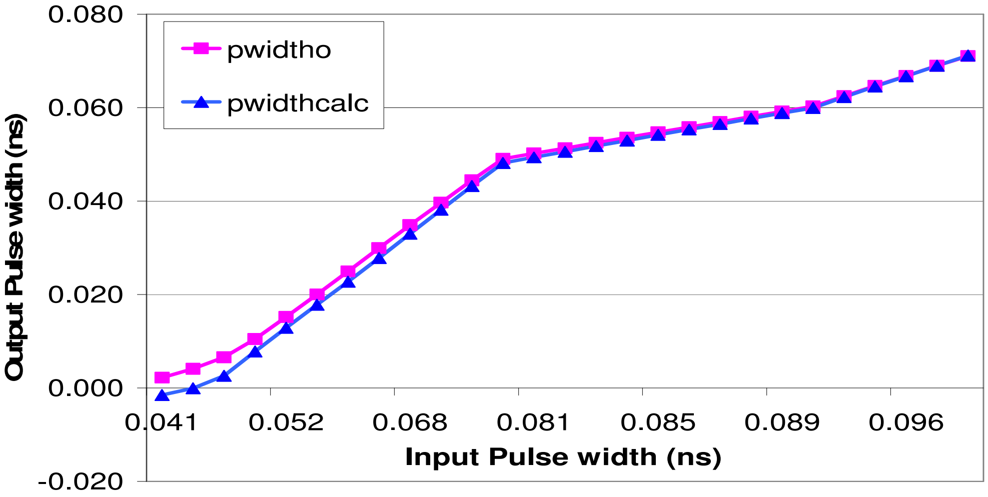


Figure 3.5: Verification of pulse width calculation for a NAND gate

3.2.4 Soft Error Estimation Methodology

Figure 3.6 shows our methodology as applied to a logic chain for a specified set of primary inputs. Here, current pulses are injected into each node. The corresponding voltage pulse is obtained by using the values from a current-voltage (I-V) transfer table. Once a corresponding output voltage is obtained, the propagated pulse width and amplitude at the output of each gate along the path are calculated, using the equations presented in the previous sections and the pre-characterized delay models. Since we also account for the state of each node when propagating the pulse, logical masking is accounted for inherently. Once the voltage pulse propagates to the flip-flop, the pulse-width and amplitude values are used to obtain the corresponding *tw*, using the flip-flop characterization table explained in section 3.2.2.

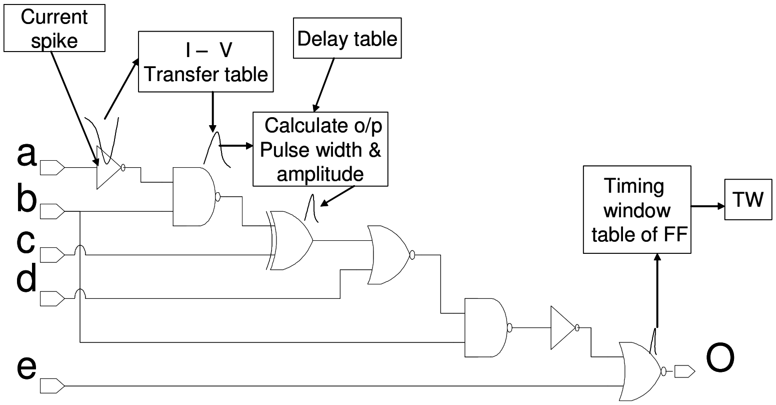

Current spike
I – V Transfer table
Delay table
Calculate o/p Pulse width & amplitude
Timing window table of FF
TW
a
b
c
d
e
O

Figure 3.6: Estimation methodology

Once the *tw* for a node to one output is known, assuming the probability of a pulse hitting a node N to be P_N, which is a factor of area occupied by the node, the pulse size, etc., the soft error rate for the output (for example O in Figure 3.6), SER_O, can be calculated as follows (Eq. 3.6):

$$SER_o = \sum_N P_N \times tw \qquad 3.6$$

Thus if the circuit has m outputs, the overall SER is (Eq. 3.7):

$$SER = \sum_m SER_o \qquad 3.7$$

3.2.5 Re-convergent Paths

The soft error estimation methodology discussed above requires enhancements for handling re-convergent paths. Here, the pulse propagation has to be considered as a special case. There are two important factors that affect the pulse propagation through such paths. The first factor is the input conditions at the re-convergent node, which can either result in a magnified or a mitigated pulse at the output re-convergent node. Another important factor is the delay difference between the two paths that lead to the re-convergent gate. This determines the delay of the gate and also determines whether the output pulse can be considered a singe pulse or two different pulses. In our methodology, both these factors can be taken into account by characterizing the cell delays and the slopes by varying the time difference between glitches occurring at multiple inputs of the re-convergent gate. After a certain time difference between the arrivals of the glitches, these edges can be considered as separate pulses and hence, the input pulses at the re-convergent gate propagate to the output as two separate pulses. Note that in both cases, the equations presented in previous sections can be used to obtain the pulse characteristics.

3.3 SEAT-LA Tool Flow - Implementation

Figure 3.7 shows the SEAT-LA tool implemented as a part of the bigger tool flow. The tool was implemented using perl and Tcl scripts to work in conjunction with other required tools. As can be seen from Figure 3.7, the back annotated gate level net-list is taken as input. The tool, design compiler from Synopsys, is used to extract the paths from each node to the output. The tool also requires knowledge of the capacitance at each node, using which the delay and slope tables are to be indexed. These capacitances were obtained from the back annotated net-lists. The state of each node was obtained for a given input vector using the model-sim simulator. Once the state of every node is obtained, SEAT-LA (in Figure 3.7) computes the pulse propagation from each node to the output and also calculates the value of *tw* as explained in the Sections 3.2.3 and 3.2.4. This analysis is done for each path of every node. Thus, as described in Section 3.2.4, the SER is obtained by summing up the *tw* values for all nodes in the path.

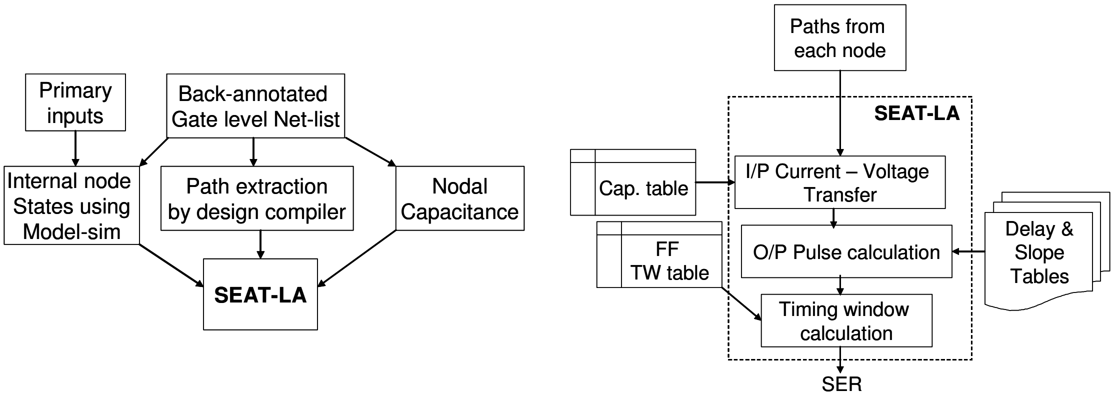

Primary inputs
Back-annotated Gate level Net-list
Internal node States using Model-sim
Path extraction by design compiler
Nodal Capacitance
SEAT-LA
Paths from each node
SEAT-LA
Cap. table
I/P Current – Voltage Transfer
FF TW table
O/P Pulse calculation
Delay & Slope Tables
Timing window calculation
SER

Figure 3.7: SEAT-LA tool flow

3.4 Results and Discussion

In this section, we present the validation results. First we present the *tw* results of simulations using small designs from the ISCAS benchmark: c17, a 4-bit ripple carry adder, a 2x4 decoder and the logic chain shown in Figure 3.6. All our designs were mapped using the following four pre-characterized cells: an inverter, a 2-input NAND, a 2-input NOR and a 2-input XOR gate. Also, each of the outputs was connected to a Transmission Gate Flip-Flop (TGFF). All our pre-characterizations were performed for the 70 nm Predictive Technology Model using HSPICE circuit level simulations. We compare the *tw* obtained from our tool with those observed by HSPICE. Next we calculate the error rates based on *tw*s and compare them with HSPICE results. Thus we also assume that the results obtained from HSPICE as accurate. Finally, we present the results of running the tool on bigger ISCAS benchmarks.

3.4.1 Timing Window Verification

In this section, we present the timing window (*tw*) of the various designs obtained using SEAT-LA and compare them to observed timing window from HSPICE simulations. Each of the smaller designs was implemented using Micro Magic MAX, a VLSI layout tool with 70 nm PTM technology transistor models. These designs were simulated using HSPICE and the extracted gate-level net list was also given as an input for the tool. Next, the timing window was measured using HSPICE simulation by moving the current pulse over a clock period at every node to obtain "tw HSPICE" (See Figures **3.9**-**3.11**) for a given input. The timing window, "tw SEAT-LA", was calculated by the tool SEAT-LA for the same input for each node using the extracted net-list.

The timing windows for the ISCAS benchmark c17 (schematic in Figure **3.8**) are presented in Figure **3.9**. Here, the observed and calculated timing windows differ by a mean error margin of 9.8% and a maximum error of 14.4%. The c17 circuit does have a re-convergent node at o23 as shown in Figure **3.8**. Thus, at certain input states, a pulse at G11 can propagate through both nodes G16 and G19 to o23. Since the delays to both the nodes are same, the delay difference between the two pulses is zero. Thus, the delay corresponding to both inputs switching is used to calculate the output voltage characteristics by our tool, instead of treating them as two separate glitches at the two different inputs of G16. The timing window evaluation for node G11 for two different cases illustrates the importance of re-convergent modeling. In Figure 3.8, the timing window for a pulse at G11, when it propagates through both G16 and G19, results in a timing window of 0.127. On the other hand, if the timing window is calculated assuming that the pulse propagates only through G19, the resulting value is 0.094.

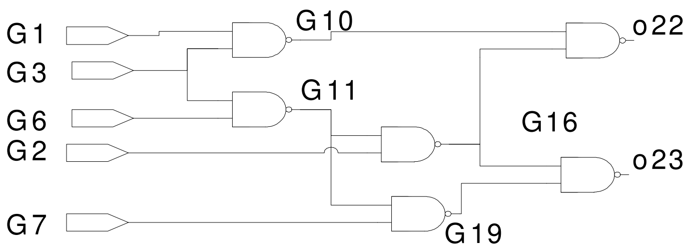


Figure 3.8: C17 schematic

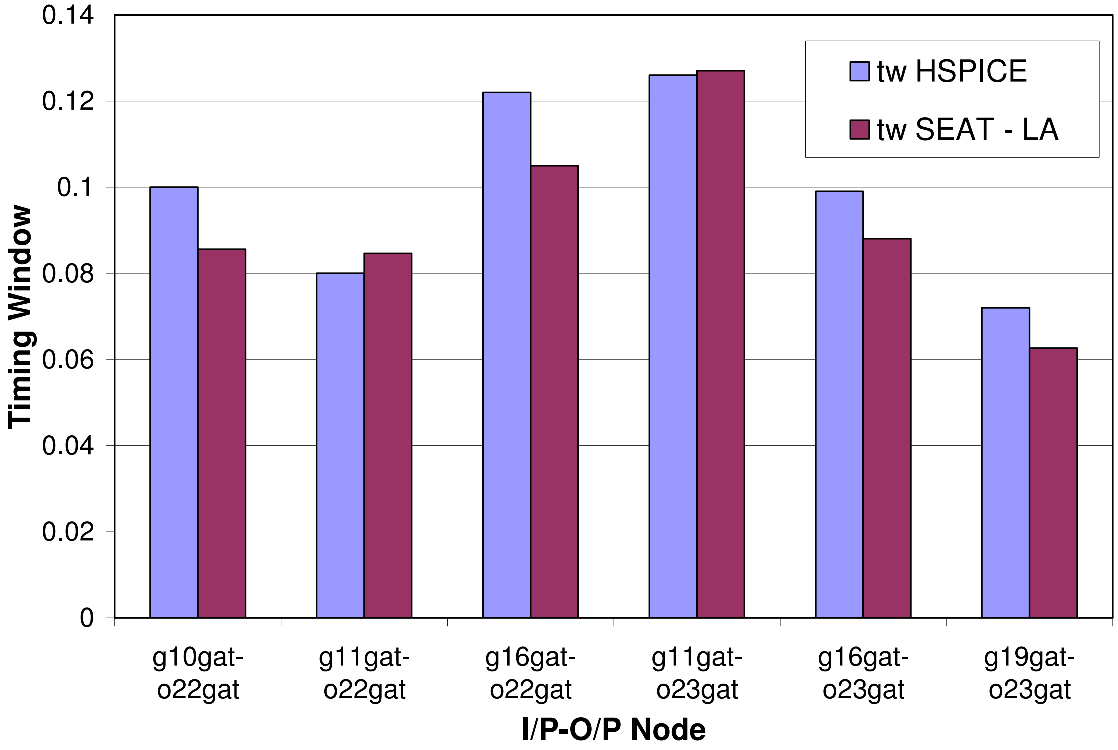


Figure 3.9: C17 timing window results

Figure 3.10 presents the results for a 4-bit ripple carry adder. The adder has four sum and one carry outputs and 13 internal nodes. The states of the various nodes were such that transients caused at only 4 of the 13 nodes propagated to the output with a non-zero probability; thus, only these results are presented here. As seen from the calculated results, apart from one value, there is a good match (mean error of 3.6%) between the observed and calculated values. It was found that this mismatch was due to the difference in the actual slope and the characterized slope, which affected the accuracy of the pulse magnitude.

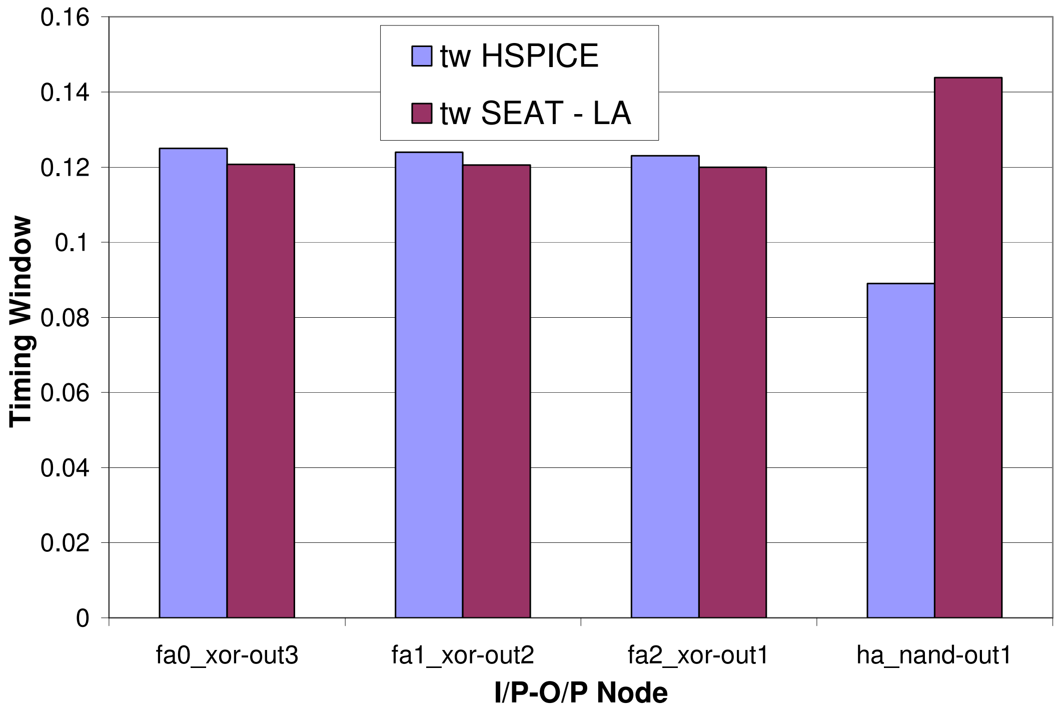


Figure 3.10: Timing window for an adder

Figure 3.11 shows the calculated and observed timing windows for the decoder and the logic chain. The observed and calculated values match for most of the nodes, except for node “ab” in Figure 3.11. The mean error was calculated to be 13.8% for the decoder and logic chain in Figure 3.6. In the case of node "ab", the tool underestimated the timing window: the reason for that is explained below.

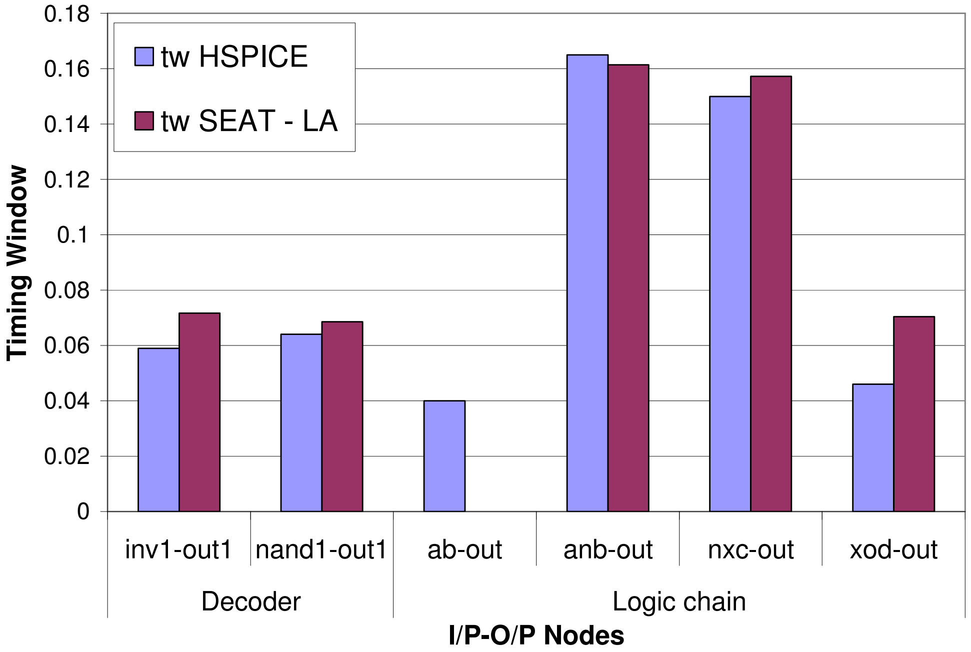


Figure 3.11: Timing window for decoder and logic chain

We observed that our tool slightly underestimates the timing window values for nodes that are farther away from the outputs while for the nodes closer to the output, the tool overestimates the timing window. For example, in the c17 results presented in Figure 3.9, the timing window for G11, which was farther away from the output, is underestimated while for the nodes closer to the output (G10, G16 and G19) it is overestimated. This is because, for current pulses occurring at nodes farther away from the output, the error in the result is mainly due to the approximations in reading the delay and slope values from the table. This results in the underestimation of the timing window.

For a current pulse occurring at nodes closer to the output, the approximations in the flip-flop characterizing plays a greater role, resulting in overestimation.

3.4.2 Soft Error Rate Verification

In this section, we present our results in verifying the SER from the observed and the calculated *tw* presented in the previous section, using equations in Section 3.2.4. The soft error numbers obtained from the equation are compared with values obtained from errors observed at the flip-flop, when injecting random errors at the nodes using HSPICE circuit simulation. For this purpose, a small test bench was set up to inject random pulses in the nodes of each design. The injection site (node) and time (within one clock period) were chosen randomly by the test bench. A current pulse of given width and height were subsequently injected. The test bench observed the state of the flip-flop and checked for the occurrence of an error. This procedure was iterated 5000 times for each design and the corresponding errors obtained to calculate the "SER Hspice" entries in Table 3.2. The times required for 5000 iterations are reported in column 6.

Table 3.2 presents the results for all the above designs as well as an inverter chain design. Here, "SER tw" is the SER calculated from the "tw HSPICE" in Figure 3.9, "SER SEAT-LA" is the SER calculated by the tool and "SER HSPICE" is the SER as obtained from the test bench explained above. The error percentage between "SER SEAT-LA" and "SER HSPICE" is presented in the last column. This error percentage was calculated by using the following formula (Eq. 3.8):

$$\% Error = \frac{SER\,SEAT - LA \;\; \sim \;\; SER\;\; HSPICE}{SER\,SEAT - LA} * 100 \qquad 3.8$$

The SEAT-LA results match well with the HSPICE values with an average error margin of 6.5% (mean). They also match well with the “SER tw” with a mean error of 7.3%. SEAT-LA also has a maximum speed up of 27000 times over HSPICE simulation while the average speed up is 15000.

Table 3.2: Soft error rate comparison

Design	**SER *tw***	**SER SEAT-LA**	**time (min)**	**SER HSPICE**	**Time (min)**	**% Error**
Invchain	0.0733	0.0765	0.01	0.0756	270	1.2
c17	0.1498	0.1382	0.04	0.1410	426	2.0
Decoder	0.0205	0.0234	0.02	0.0210	670	10.1
Logicchain	0.0668	0.0649	0.22	0.0620	719	4.4
Adder	0.0355	0.0388	1.4	0.0332	872	14.4

We also used SEAT-LA to estimate SER when a HLFF was used to latch the output. The results are shown in Figure 3.12. There was an increase in the SER for all the designs, as seen here as compared to using a TGFF. Because the HLFF is a pulse triggered flip-flop, it has a bigger timing window than the TGFF, especially at lower pulse widths (See Figure 3.2).

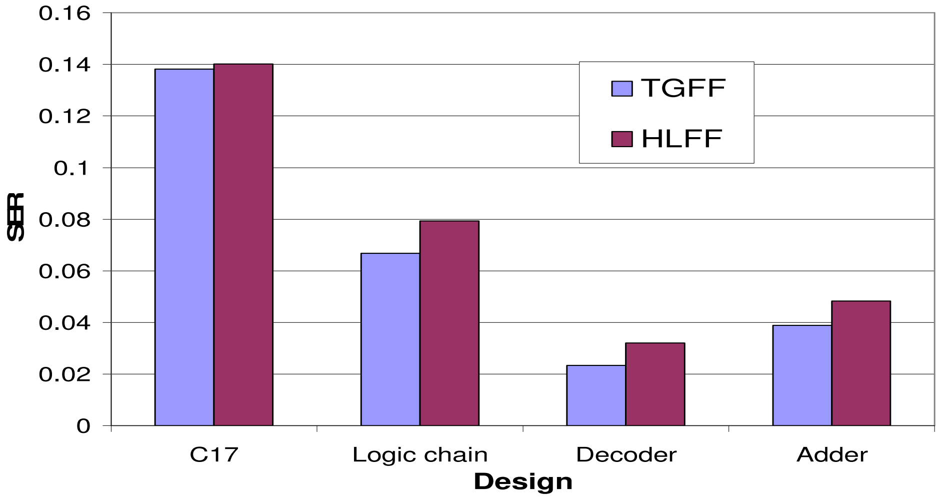


Figure 3.12: Effect of using different Flip-Flops

3.4.3 Simulations on ISCAS Benchmarks

In this section, we present the scalability of our approach using larger ISCAS benchmarks. All the benchmarks used were much bigger than the small designs. Therefore, verification by HSPICE could not be done as in the case of previous designs, due to very long simulation times. All the simulations were run on Sun Fire V210 work stations with SOLARIS Unix operating system and 4GB RAM. Table 3.3 gives the error rate and also the time taken in minutes. We observe that the time required by our tool increases with the number of paths in the design. However, soft errors in combinational logic become more important with shallower pipeline stages. Consequently, the number of paths to be analyzed by our tool for a single combinational logic stage is expected to be reduced, due to reduced logic depth in a pipeline stage in future. Note that since this tool was written using perl and tcl scripts, there can be many optimizations to make the tool work much faster and efficiently than it presently does.

Table 3.3: SER for ISCAS benchmarks

Circuit Name	Circuit Function	Total Gates	# of I/Ps	# of O/Ps	SER	Time in min
C432	Priority Decoder	160	36	7	0.0725	108
C499	ECAT	202	41	32	0.0041	216
C880	ALU and Control	383	60	26	0.0188	102
C1355	ECAT	546	41	32	0.0070	162
C1908	ECAT	880	33	25	0.0011	1073
C2670	ALU and control	1193	233	140	0.0034	547

Thus in this chapter a new tool to model soft errors in logic circuits is proposed. This tool can form an integral part of the synthesis tool flow and can be used to predict SER at the design stage along with other performance evaluation techniques like static timing analysis. The proposed tool is verified with HSPICE simulations with the assumption that HSPICE being the most accurate available standard. It is also shown to be much faster than HSPICE simulations while the scalability of the tool is verified by simulations on larger ISCAS benchmarks. But, as technology scales, there are other variability issues that affect the current technology circuits apart from soft errors. It is important to analyze the effect of such variability issues on soft errors, which is dealt with in detail in the next chapter.

Chapter 4

Reliability Effects on Soft Errors

4.1 Introduction

Sub-90 nm circuits currently face an unmanageable problem of unpredictability in process parameters of their individual devices [27]. Such uncertainties not only affect the circuit power and performance characteristics [28], but also the resilience to transient faults like soft errors. Since the soft error resilience of any device has a strong dependence on the process parameters such as device length, threshold voltage etc., the effect of process variability on SER of the circuits cannot be neglected. One of the main causes of such variations in process parameters is manufacturing parameter fluctuations, due to increasingly challenging fabrication requirements [29], [30]. Such effects are of static nature, which may be characterized and detected immediately after manufacturing to some extent. Another reason for changes in device behavior is the dynamic variations of the individual device due to runtime phenomena, such as device degradation, power supply noise, and temperature imbalances. Each of these phenomenon impact the device characteristics and thereby the resilience of the devices to soft errors.

One of the biggest problems faced by current circuit designers is manufacturing parameter fluctuations. The aggravation of such defects is attributed to technology scaling, which poses very difficult fabrication challenges. The expected device parameters vary across different dies and within a die itself, namely inter-die and intra-die variations respectively [31]. Such variations may significantly affect power and delay characteristics of the circuit and thereby impose challenges in meeting the necessary budgets. Apart from these variations being a reliability concern, such variations introduce newer reliability concerns in the form of modulating the susceptibility and resilience of the circuits to transient errors like radiation induced soft errors. Due to reduced feature sizes, any slight change in the process parameters contributes to a significant percentage of variations; thereby, their impact is quite significant. Consequently, dynamically natural variations due to power supply variations and increased temperature across chips have also been one of the important causes for variations [32], [33], [34], [35]. Also, variations due to device degradation because of Hot Carrier Effects (HCE) and Negative Bias Temperature Instability (NBTI), which are attributed to the circuit usage over a period of time, may lead to run time degradation and uncertainties of the circuits [36], [37], [38]. Although such changes may contribute to more permanent failures [39], the degradation may once again affect the transient error vulnerability of the circuit, since it changes the operating conditions and the device parameters of the circuits, both of which affect the SER of a circuit significantly.

Although there have been many methodologies that propose modeling and optimizations for soft errors [10], [5], [12], [13], [14], [15], [16], [17], [18] and [40], it is important to address soft errors in the presence of other reliability issues as well. To the best of the author's knowledge, this is first work that has looked at effects of other reliability issues on SER. Thus, the primary contribution of this chapter is to analyze of SER in combinational logic circuits under static and dynamic variations. This is done through the use of various tools to estimate the state of individual devices and their process parameters due to HCE and NBTI induced degradation over a period of time. The chapter is organized as follows: Section 4.2 presents the variation models used by in this work. The details of SER estimation tools are discussed in Section 4.3. Finally, the simulation results are presented in Section 4.4.

4.2 Modeling Variation

In this chapter, we considered three different categories of variations: static, dynamic and aging related variations. The methodologies used to model these types of variations are discussed in detail in this section.

4.2.1 Static Variations

Static variations are primarily due to manufacturing uncertainties such as variations in channel length, channel width, thickness of gate oxide, and threshold voltage. Inter-die variations (such as V_{th}) change the value of the parameters in all the transistors in a die in the same direction. Processing temperatures, equipment quality, wafer polishing and placement causes these variations. Examples are channel length, channel width and variations between individual metal layers used for routing [29]. These variations result mainly in differences in power and delay.

In contrast, intra-die variations can be either systematic or random. These variations arise due to misalignment of wafer, Random Dopant Fluctuations (RDF) and uneven planarization steps. Transistor parameter shifts resulting from systematic variations are correlated and are dependent on the neighboring transistor parameters. Random variations shift the transistor parameters independent of the locality. In particular, RDF leads to non-uniform distribution in transistor threshold voltages (V_{th}) in the circuit and is a key cause of random variations [32]. These static variations can be translated into changes in the effective threshold voltage (V_{th}) [33]. Hence the effect of change of V_{th} due to process variations on the SER of various circuits is studied in this work.

4.2.2 Dynamic Variations in Power Supply and Temperature

The dynamic variations considered by us in this work are the variations due to temperature changes in circuits and power supply variations. In this section we discuss briefly the impact of such variations on circuits.

4.2.2.1 Power supply variations

The variations in power supply voltages have been one of the most important challenges as technology scales down. This is because of the decreased supply voltage, which results in a much larger ratio of the peak noise voltage to the ideal supply voltage [41]. This power supply noise is primarily a voltage drop in the power distribution networks, resulting in different voltages at different parts of the same chip. These variations are mainly due to resistive and inductive voltage drops across power supply networks. A power supply noise analysis methodology for circuits and microprocessors has been discussed in [42]. A minimum power supply fluctuation of 10% is acceptable [41]. Thus, in our work we have varied the power supply for the benchmarks by 10% and presented the corresponding variation of SER. We discuss these results in section 4.4.

4.2.2.2 Variations in temperature

Heat generation in chips has increased rapidly with recent scaling trends and increased transistor density. This has led to non-uniform substrate temperature profiles, affecting both interconnect and transistor delays. In the case of interconnects, the rise in temperature increases the resistivity of metals thus resulting in increased delays. In devices, temperature affects both the mobility and the threshold voltage. A rise in substrate temperature reduces the mobility of electrons/holes in MOSFETs because of increased scattering at higher temperature [34]. The threshold voltage also decreases with an increase in temperature because of the change in Fermi-potential ($Ø_f$) [35]. These two effects determine the trends in delay in logic circuits; thus, they affect the electrical masking capability of the logic circuits. The change in transistor delays also affects flip-flop characteristics, such as the set-up and hold times. This in turn results in a change in latch window masking capability of logic circuits. Analyzing these effects in detail is important, yet no work has contributed to such an analysis thus far.

4.2.3 Variations Due to Aging

The dynamic variations considered by us in this work are the variations due to device degradation over a period of time because of HCE, NBTI and power supply variations. To analyze the effect on SER due to such variations, we first modeled the impact of such variations on the circuits.

4.2.3.1 NMOS degradation due to HCE

The variations in threshold voltages of different devices over a period of time are considered for dynamic variations in the process parameters. Such changes in process parameters are primarily due to factors like temperature and logic activity of the device, which impact the basic I-V characteristics of the devices.

One such phenomenon, which leads to the degradation of the device due to the aforementioned factors, is Hot Carrier Effect (HCE). The Hot Carrier Effect is the phenomenon of trapping high-energy charge carriers at the gate oxide or creating new traps due to impact ionization effect. This trapping of charges increases the transistor threshold, thereby affecting the power and performance of the device. Such variations are more predominant in the NMOS transistors compared to PMOS transistors, primarily due to the negligible degradation rate of the saturation current of PMOS when compared to NMOS [43]. In this work we developed a tool for individual devices using the analytical models presented in [44], to observe the aging impact on devices. Eq. 4.1, 4.2 and 4.3 are the prime equations governing the degree of degradation of a device.

$$\frac{dN_{it}}{dt}\left[1+AN_{it}\right]=KI_{bb} \qquad 4.1$$

$$I_{bb}=\frac{C_1}{W}I_{DS}\exp(-\frac{B_i}{\mathrm{E}_m}) \qquad 4.2$$

$$\mathrm{E}_m=\frac{V_{ds}-V_{dsat}}{l_d} \qquad 4.3$$

Where, N_{it} is the number of trapped charges per cm^2 (for unit deposition depth). A = 5 x 10^{-9} cm^2, K = 5 x 10^{15}, C = 2, B_i = 4.41 x 10^6 V/cm is the ionization coefficient. E_m is related to peak electric field along the channel and is given by Eq. 4.3. I_{DS} is the drain source current that flows through the device during a transition (Figure 4.1). We use these equations in conjunction with the analytical model presented in [44] to estimate the threshold variation of a single NMOS device under constant current conditions. The degradation of V_{th} under such constant stressing conditions is demonstrated in Figure 4.2 for a 70 nm NMOS transistor.

Note that this degradation is modeled with the assumption of constant stressing of the NMOS device, which is essentially continuous current flowing through it for the given period of time. In circuits, however, current flow through the device is occurs only during switching and over a short period of time. Hence the actual age of the circuits can be related to the stressed age using Eq. 4.4.

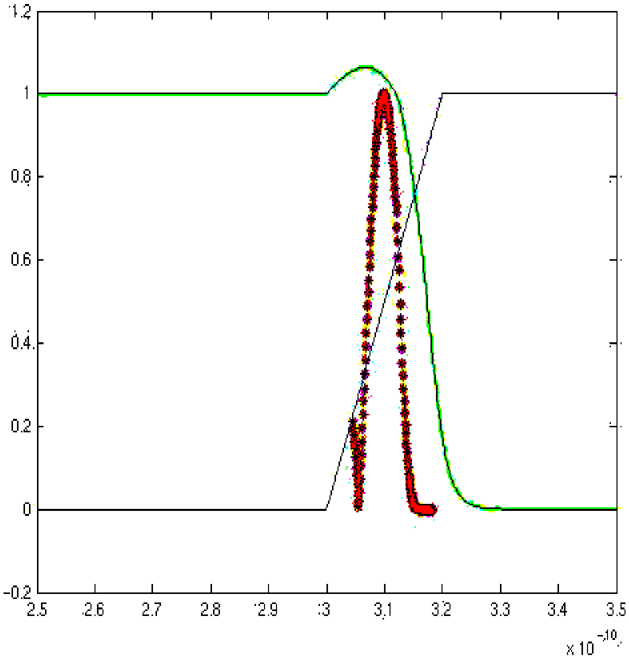


Figure 4.1: Device current during a transition

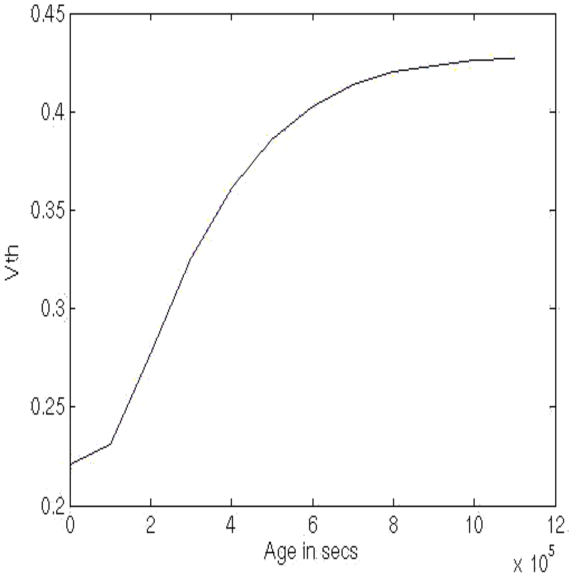


Figure 4.2: Variation of Vth of NMOS with aging

$$\Delta S = \frac{T * P}{CLK} \left[\frac{\int_{t1}^{t2} I_{sub}\, dt}{I_{sub\_dc}} \right] \qquad 4.4$$

Where, ΔS is the stressed age, T is the actual age, P is the switching probability of the gate of the NMOS transistor under consideration, I_{sub} is the substrate current, CLK is the clock frequency and I_{sub_dc} is the total constant current under stressed conditions that may flow during time T. The equation primarily exploits the number of transitions of the gate over a period of time and the current flowing through the device, based upon the exact current estimates obtained from HSPICE. The MATLAB model is integrated with the circuit simulation tool HSPICE for precise estimation of individual device threshold changes over a period of time. We obtain the actual ages of each of the NMOS transistors in a circuit during its operation, based on the actual current flowing through the devices during any transition. These numbers are used to obtain the actual ages of the devices and thereby the new threshold of the devices at the end of a time window of observation, which, in our case, is one day. The thresholds are then used to obtain the new current estimates and the ages at the end of next day. Figure 4.3 demonstrates the flow of the tool developed by us which iteratively estimates the degradation of the devices given a SPICE model of any circuit. Note that unlike the random behavior of static process variations these dynamic variations considered by us are more deterministic due to their strong dependence on the logic activity of the transistors.

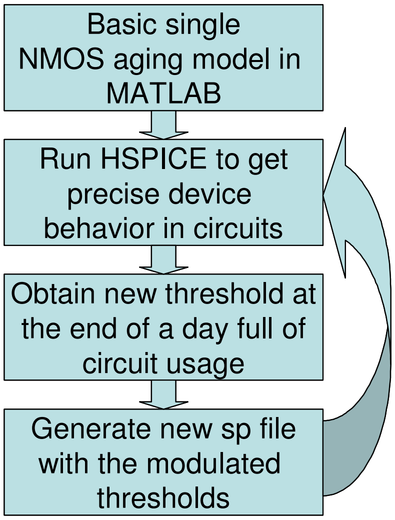

Basic single NMOS aging model in MATLAB
Run HSPICE to get precise device behavior in circuits
Obtain new threshold at the end of a day full of circuit usage
Generate new sp file with the modulated thresholds

Figure 4.3: Tool flow for estimating degradation

4.2.3.2 PMOS degradation due to NBTI

Another phenomenon that leads to slow degradation of especially PMOS devices is NBTI. NBTI in PMOS occurs for a negative gate voltage (V_{gs} = -V_{DD}) and results in increased threshold voltage with time. The main reason for the NBTI effect is found to be the presence of an increased number of positive interface traps caused by the displacement of Si-H bonds, which are induced by positive holes from the channel. The NMOS transistor has a negligible level of holes in its channel and hence does not suffer from NBTI degradation [38].

Recently, a number of works have attempted to model NBTI [37], [38]. In this work, we follow the approach used in [38] to model the change in the threshold voltage due to NBTI. As explained here, when the PMOS device is on, it is in a stressed stage. When it is off, it is in a recovery stage where the threshold voltage attempts to drop back due to the diffusion of the H atoms back to the interface to form the S-H bonds. These two stages are modeled by the Eq. 4.5 and Eq. 4.6 [38]:

During Stress,

$$\Delta V_{th} = \sqrt{K_V{}^2 (t - t_0)^{1/2} + \Delta V_{th0}{}^2} + \delta_v \qquad 4.5$$

During Recovery,

$$\Delta V_{th} = (\Delta V_{th0} - \delta_v) \times \left[1 - \sqrt{\eta (t - t_0)/t}\right] \qquad 4.6$$

Where (Eq. 4.7),

$$K_V \propto \sqrt{C_{ox}(V_{gs} - V_{th})} \exp(E_{ox}/E_0) \exp(-E_a/kT) \qquad 4.7$$

Here, E_{ox} = (V_{gs}- V_{th})/T_{ox}, C_{ox} is the oxide capacitance, k is the Boltzman constant, δ_v = 5 mV, η is 0.35, t and t_0 are the initial and final times, E_0 = 2 MV/cm and E_a = 0.13 eV. From Eq. 4.5 and Eq. 4.6, an expression for long term degradation can be derived after n cycles of recovery for a duty cycle or a stress period of β and a clock period of D. Thus, as in [45] (Eq. 4.8):

$$\Delta V_{th} = K_V \beta^{0.25} T^{0.25} \left[\frac{1 - \left(1 - \sqrt{\eta(1-\beta)/n}\right)^{2n}}{1 - \left(1 - \sqrt{\eta(1-\beta)/n}\right)^{2}} \right] + \delta_v \qquad 4.8$$

Using the above equations, the change in threshold voltages are estimated, taking into consideration the stress and recovery time based on the static probability of the gate voltage for each node in a circuit. With this information, the SER for any given circuit can be estimated with degraded PMOS in these circuits. Results for this analysis are presented in Section 4.4.

4.3 SER Estimation Tools

In this section a brief background on the SER estimation set-up and methodology used for SER analysis in the presence of variations is presented. In our simulations, we analyze soft error rates in small custom benchmark circuits that represent combinational logic in general. We use a circuit level tool built using perl scripts, which invokes HSPICE to estimate SER accurately in these custom benchmark circuits. This circuit level tool operates on a SPICE file containing the design and the flip-flops attached to the outputs. We also estimate the SER variations for bigger ISCAS-85 benchmarks with respect to static threshold changes in circuits using SEAT-LA.

The circuit level estimation tool uses an accurate method for SER estimation. This tool requires a SPICE netlist of the circuit for which the SER needs to be estimated. This tool provides an SER for a given current pulse generated by a pulse strike. A double exponential current pulse similar to those used in previous chapters was used. The circuit level tool then uses HSPICE to calculate the timing window (*tw*) for which this current pulse at each node in the logic circuit causes an error at the output. As defined in Chapter 2, the timing window is the amount of time for which the current pulse at a node causes an error at the flip-flop output divided by the clock period. Thus, *tw* is the probability that an error occurs for a current pulse at a node N. If the probability of the current pulse occurring at this node N is known, then as in Chapter 3, the SER for that circuit (with N nodes and a single output O) can be calculated using Eq. 3.4.

Apart from being accurate, the circuit-level tool also allows the user to change parameters of individual devices that occur due to variations. But the circuit level tool works effectively for only very small circuits with 5-10 gates. When circuits are larger, analysis using HSPICE becomes very tedious and time consuming. Hence, to analyze SER variations in bigger ISCAS benchmarks, we use SEAT-LA.

4.4 Results and Discussion

We used a set of custom designed circuit layouts and gate level designs of ISCAS-85 benchmarks for our simulations. The custom designed circuits were laid out using Micro Magic MAX tool [46] in 70 nm PTM technology [23] and simulated using HSPICE. These circuits include a ten stage inverter chain, a logic chain similar to the one used in Chapter 3, the C17 ISCAS benchmark, a 2X4 decoder and a four bit Ripple Carry Adder (RCA). The gate level designs of the ISCAS-85 benchmarks were tested using SEAT-LA. All the circuits designed drove a Transmission Gate Flip-Flop (TGFF) which captured the outputs.

4.4.1 Static Variations

In our first set of simulations on static variations, we tested the impact of inter-die variations on both the custom-designed benchmarks and ISCAS-85 benchmarks. The results of these tests are shown in Figure 4.4 and show a maximum variation of 16% for a 3σ (maximum) variation of 10% in threshold voltage. More importantly, we observed an increase in the SER as the threshold increases. This trend is the opposite of what is expected and explained in [20].

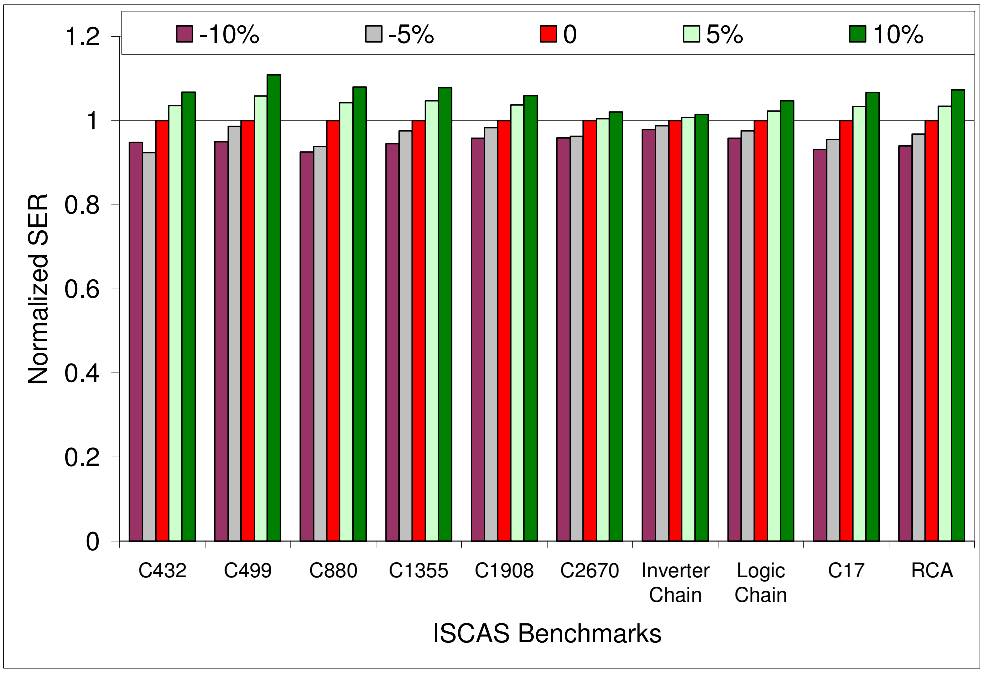


Figure 4.4: Normalized SER due to static variations for ISCAS and custom benchmarks

As mentioned in [20], there are two different phenomena that determine the trend of SER with change in threshold voltage. The first is the increase in the gain of static logic circuits with increase in the threshold voltage. This, in turn, reduces the electrical masking capability of static logic circuits resulting in an increase in SER. The second is the increase in flip-flop set-up and hold time with increased threshold voltage. This increases the (latch-window) masking capability of the flip-flops, as larger pulses are now needed to be latched on by the flip-flop. Thus, the SER trend depends on which of the above factors dominates more. In [20], it was found that the SER decreased with large increase in V_{th}. This was due to the large increase in the flip-flop set-up and hold times, while the increase in gain of the logic circuits had a much smaller effect on the overall SER trend.

Since our initial analysis on SER variation due to variations in V_{th} had an opposite trend to what is expected as in [20], we extended our simulations to estimate the change in SER by increasing V_{th} beyond 10% for two of the smaller benchmarks. These results are presented in Figure 4.5. From the figure it is noted that there is an initial linear increase in SER with small increase in V_{th}. This is clearly due to the dominant influence of CMOS gate gains in the logic circuits. But as V_{th} increases further, the reduction in setup and hold times of the flip-flops at the end of the data-paths starts playing a more important role, thus reducing SER drastically after a point, as seen in Figure 4.5.

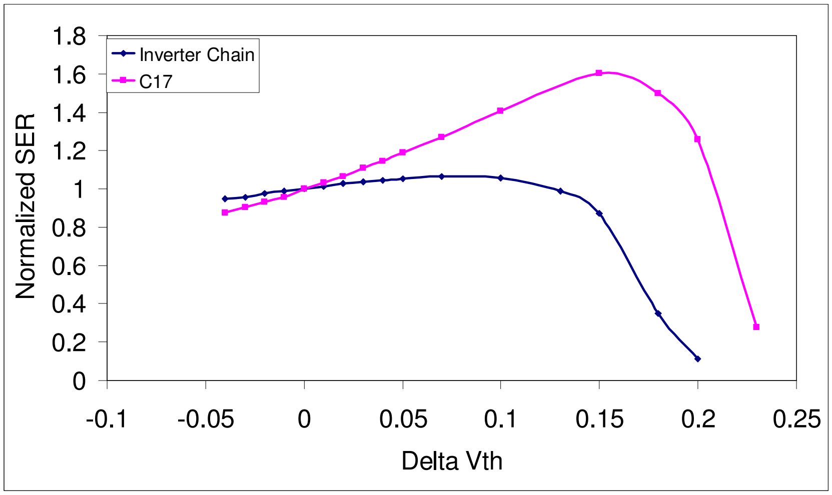


Figure 4.5: SER variation with increase in V_{th}

These curves indicate interesting trade-offs that can be used for design optimizations in circuits for both power and SER mitigation. Therefore, this can be used to estimate the V_{th} for which the circuits can be used to save leakage power and also decrease SER. Also, this technique can be applied for noise mitigation in general rather than just mitigating radiation induced soft errors.

To model the effect of intra-die random variations, we performed SER analysis on two of the custom benchmarks with random V_{th} assignments for the different devices. A Gaussian distribution in threshold voltage with $\mu = 0.2$ and $\sigma = 0.02$ for NMOS devices and $\mu = 0.22$ and $\sigma = 0.02$ for PMOS devices was used to assign V_{th} to each of the devices. Once the V_{th} was assigned for each device in the circuit, the circuit was simulated for SER analysis. Due to long simulation times, SER analyses for 100 different assignments were performed to mimic Monte-Carlo type of simulations as our initial simulations with just 10 different assignments showed a peak-to-peak variation of 41%. Figure 4.6 shows the variation of SER normalized to SER for nominal threshold values.

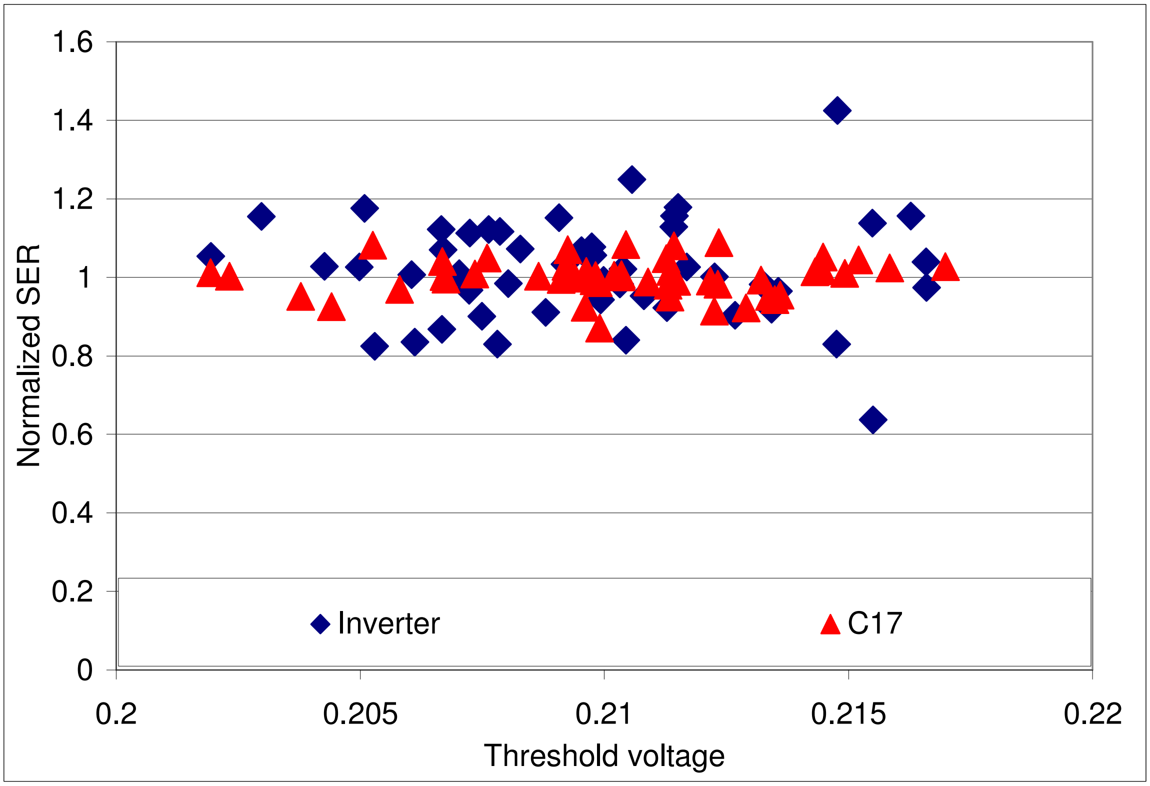


Figure 4.6: Impact of random V_{th} variations

Clearly there was a large variation in the SER with V_{th} variation. Thus, further analysis was done to determine the reason for such a large variation in SER. For this analysis, the flip-flop characteristic variation was studied by determining the variation in the timing window for different voltage pulse widths by sweeping the pulse at the input of the TGFF used in the designs. There was a large variation in the timing window especially for the smaller pulse widths (average variation of 30%). Since only smaller pulse widths reach the input of the flip-flop, the large variation in SER was the result of the large variation in the flip-flop characteristics, which determines the trend in SER.

4.4.2 Dynamic Variations in Power Supply and Temperature

Our next set of simulations studied the effects of power supply variations on the SER on the different benchmark circuits. Figure 4.7 presents these and shows that for a 10% fluctuation of variation in power supply voltage, the SER varied by a maximum of 25% among all the circuits considered. Here, as expected, the SER increased with decrease in voltage. Also, it is interesting to note that the variation in SER is almost linear with respect to the change in voltage.

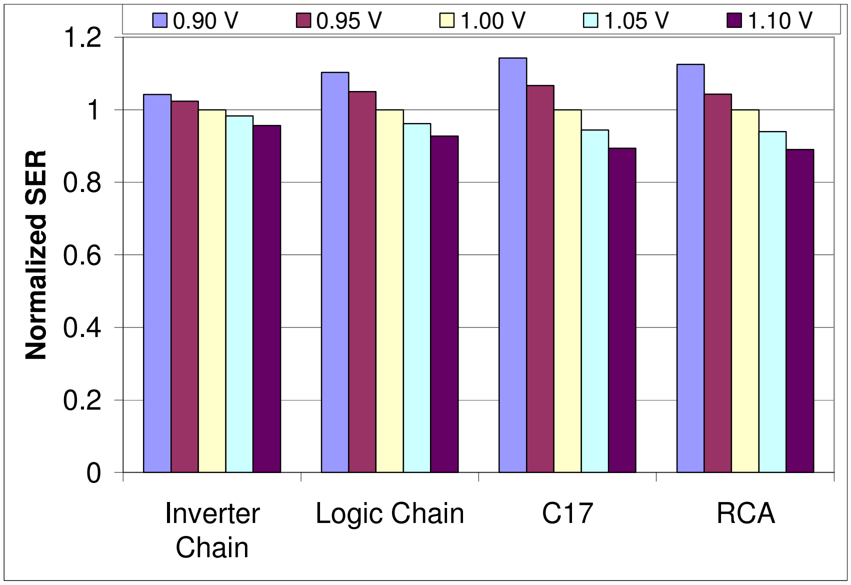


Figure 4.7: Variation of SER with power supply variations

Next, we also studied the effect of increased temperature on SER for the custom benchmark circuits. Figure 4.8 shows the variations in SER with increase in temperature. As seen here, there are vast differences in trends in different designs. To determine the reasons for the different trends, it is important to study the effect of temperature on flip-flop (TGFF) characteristics. For this analysis, the *tw* for trapezoidal pulses with increasing widths were studied.

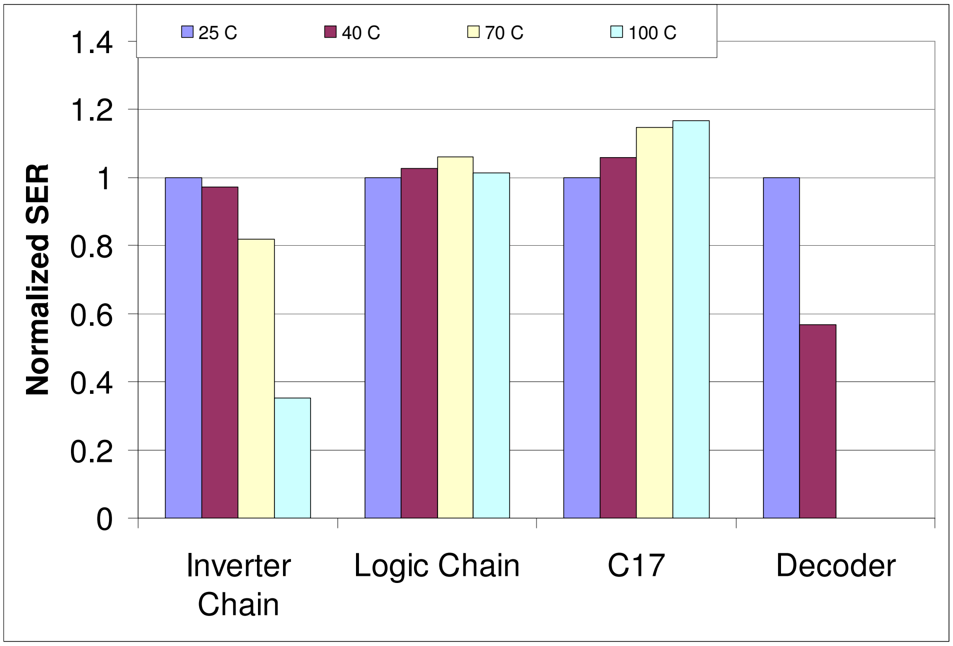
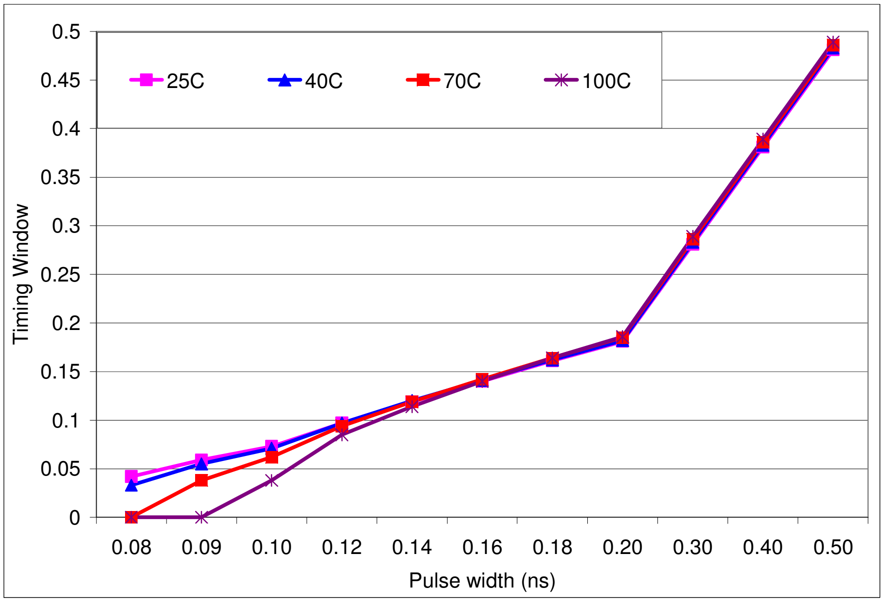


Figure 4.8: SER variations with temperature

Figure 4.9 shows the timing window variation for increasing pulse widths at different temperatures. As seen here, the timing window is much lower at higher temperatures for small pulse widths, while it was similar or, in fact slightly greater (than that for lower temperatures) for large pulse widths. Thus, for designs with longer data paths, the pulse widths that reach the flip-flop are very small, resulting in very small SER at higher temperatures as seen for inverter chain and decoder in Figure 4.8. On the other hand, for designs with very short data paths like C17, the pulse width (due to the same strike) reaching the flip-flop is larger, thus having an opposite trend in SER as seen in Figure 4.8. To confirm this, the SER of a 2 stage inverter (similar stages as c17) was found to have a much smaller reduction in SER with increase in temperature.



S

Figure 4.9: Timing window variation for TGFF with temperature

4.4.3 Variations Due to Aging

Next, we performed simulations on the custom benchmark circuits to find the effect of device degradation due to HCE by using the tool discussed in section 4.2.3.1. Thus we studied the effect of HCE on the threshold voltages of different NMOS transistors in each of the circuits. These threshold variations were obtained by assuming a 50% switching activity at all the inputs of each of the circuit considered. The variation of average V_{th} after every 200 days of operation of these circuits is presented in Figure 4.10. From Figure 4.10, it is clear that the V_{th} variation depends on the type of circuit. For example, in circuits like inverter chain and C17, where the input transitions result in a larger number of transitions of devices in the circuit, there is a large increase in V_{th} with age. For circuits like the logic chain and the RCA, there is not a large increase in the V_{th}.

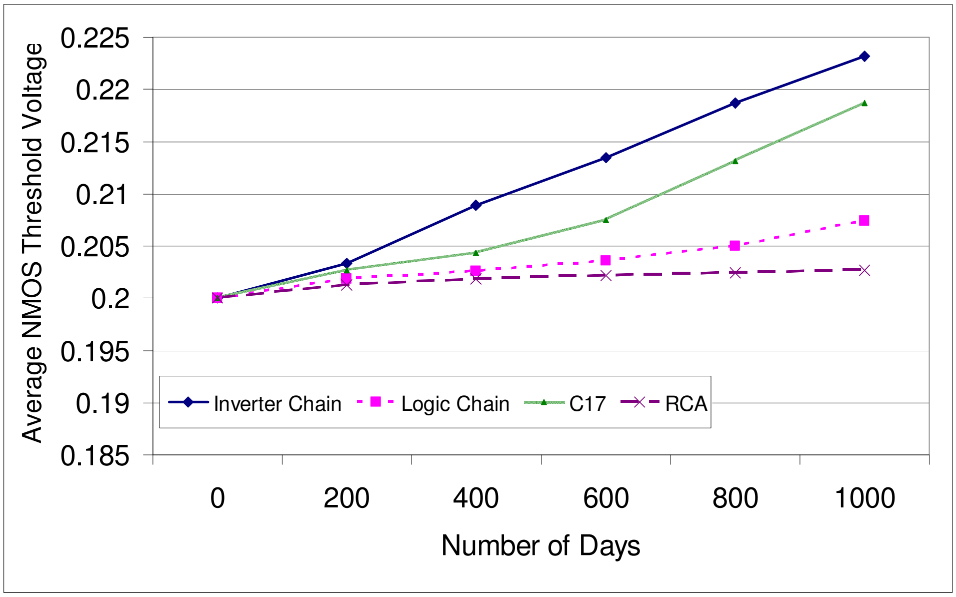


Figure 4.10: V_{th} variations due to HCE

We also studied the degradation of PMOS devices due to NBTI modeled as presented in section 4.2.3.2. For this task, the static probability of all the internal nodes was obtained based on a 0.5 input transition probability for each circuit. Then, using these transition probability values, the change in V_{th} in all the PMOS devices in each circuit was obtained. Figure 4.11 presents the average variation of V_{th} of the PMOS devices in different circuits. As seen from the figure, there is only a small increase in the threshold voltage, which is in fact smaller than those observed for HCE.

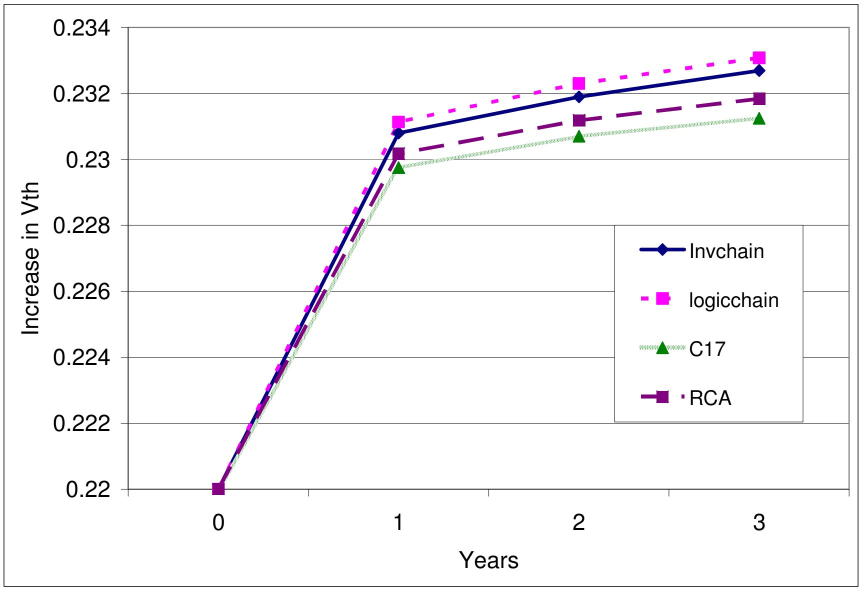


Figure 4.11: V_{th} variations due to NBTI

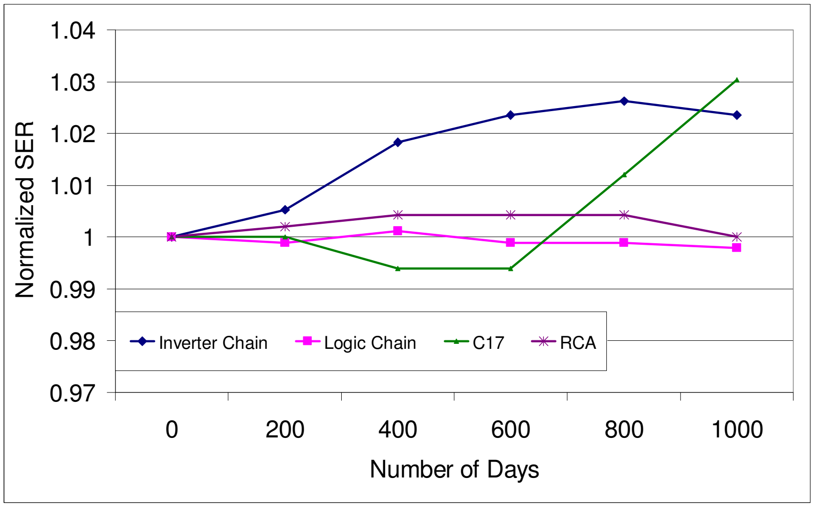


Figure 4.12: SER Variation due to HCE alone

First, the effect of HCE alone on SER was studied. Figure 4.12 shows the variation of SER due to HCE only. Here it is seen that the variation in SER is very low showing no real trend. This is mainly due to the selective increase in some of the NMOS

V_{th} values, while the PMOS device V_{th} remains the same thus making the PMOS device stronger. This actually results in making the positive voltage spikes across some circuits bigger as the NMOS device becomes weaker with increased threshold.

Next, the SER for different bench mark circuits was obtained by incorporating the changes in V_{th} due to the two aging effects, as both effects affect the V_{th} of the corresponding devices simultaneously with age. Figure 4.13 presents the results obtained for aging effects on SER. As expected, the SER variations are not as large because of the very small variations in the threshold voltages due to aging. Also, there is not a very clear trend in these variations mainly because the change in V_{th} is different for NMOS and PMOS devices due to the two effects, which make it difficult to predict the masking effects of different designs.

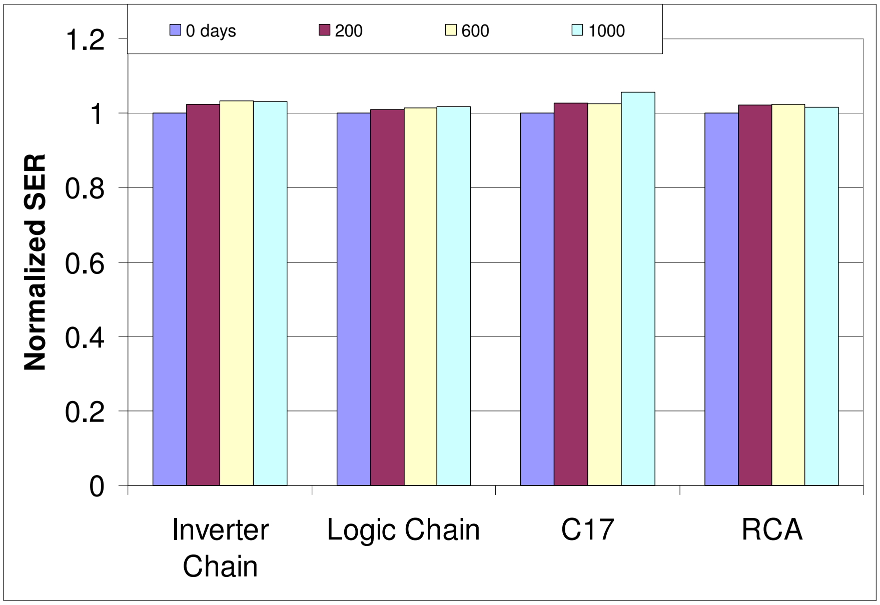


Figure 4.13: SER variations due to NBTI and HCE

Finally, simulations were performed on one of the benchmark circuits (the inverter chain) after incorporating different variations simultaneously to find the overall impact on SER. Table 4.1 shows the overall impact of these variations in SER. Here, the first column (%dvt) is the percentage change in V_{th} due to inter-die variations, the second column (Days) represents the age of the circuit in number of days, the third column (Temp) represents the temperature in Celsius at which the circuit works, and the fourth column presents the normalized SER values for each case. From these results, it can be noted that temperature has the maximum impact on the SER of the inverter, as the results are comparable to its effects on the inverter chain as shown in Figure 4.8. As expected, aging has very little impact when compared to the other variations on SER.

Table 4.1: Overal variation impact on SER of inverter chain

%dvt	**Days**	**Temp (°C)**	**SER**
0	0	25	1
5	999	100	0.42
10	600	100	0.40
10	999	70	0.84
10	999	100	0.40
-5	999	100	0.45
-10	600	100	0.49
-10	999	70	0.88
-10	999	100	0.47

Chapter 5

Conclusions

5.1 Summary of Contributions

This work first presented the circuit-level analysis of soft error rates in different flip-flops and adder structures, which are representative of data-path components. Design decisions on choosing the kind of architecture could be made, using the insights gained from this analysis. Second, the effect of power saving techniques, such as voltage scaling and increasing threshold voltage, on soft error rates was analyzed. Although these techniques were accurate, they cannot be used efficiently to model soft errors in larger circuits.

Thus, in this work, a new methodology to model SER in logic circuits was proposed. A logic level tool was built, which took a Verilog netlist and the parasitic capacitances for the nodes in the netlist as input. It computed the SER for the circuit for any given current pulse. The tool and the methodology were verified using HSPICE simulations and the results were presented. This tool can be used as an integral part of the synthesis tool flow where SER predictions can be made for designs when other performance evaluations, such as timing and power, are predicted. Hence, using this logic level tool could be useful to make informed design decisions.

Another primary contribution of this work involved an analysis of SER in combinational logic circuits under static, dynamic and aging-related variations. For variations due to aging, different tools were used to estimate the state of individual devices and their process parameters, as a result of HCE and NBTI induced degradation over a period of time. The variations that were studied included static variations such as inter- and intra-die process variations, dynamic variations due to power supply voltage variations and temperature- and aging-induced variation due to HCE and NBTI.

First, the methodology that was used to model these variations was presented, based on previous works. Next the circuit level analysis tool to measure SER in various circuits was presented. Finally, the results on the effects of different variations on SER were presented.

Another important contribution of this work, building on the previous results in [47], is for these results to be used as a starting point for experimental measurement of SER in different devices, such as microcontrollers and microprocessors. Although experimental measurements of SER have been performed by many companies in the industry [48], [49], very little has been published in the open literature that could be used to gain insights, as the error rates are usually presented in anonymous units. The results presented in the appendix show an increase in error rates as result of both technology scaling and voltage scaling in microprocessors.

5.2 Summary of Results

The most important results obtained are listed below:

- From the circuit level analysis on the effect of threshold voltage on flip-flops, we have found that the SER of flip-flops decreased dramatically with increase in threshold voltage; therefore modification of the threshold voltage can be used for reducing both power and error rates of data paths.
- Adder circuits and in general combinational circuits were affected by the timing window and capacitance effects, which in turn affected the electrical and latching window masking. These phenomena determined the occurrences of multiple bit errors in adder circuits. Since multiple bit errors are much costlier to correct, different types of adders were analyzed to determine which ones would be preferred for preventing multi-bit errors.
- Voltage and frequency scaling in logic circuits resulted in a slight decrease in $Q_{critical}$ values in general, thus increasing the SER. One way to optimize and improve error rates in logic circuits is to use appropriate flip-flop designs. Using flip-flops with shorter set-up and hold times would result in reducing the number of multiple bit errors occurring in the circuit due to single particle hits. Also, the effect of increasing threshold voltages on the circuit was analyzed. As expected, the $Q_{critical}$ of nodes increased slightly while the timing window of nodes also increased thus resulting in a trade-off.
- The SEAT-LA tool was built using a newly proposed methodology. This was verified by comparing the SER predictions with those obtained using a tool running HSPICE simulations on these designs with random node selection and at random times. The verification results were presented here and the error margins were approximately 6.5% (mean) as compared to circuit level simulations, with

an average calculation speed up of 15000. Running the same tool on bigger ISCAS benchmark netlists also proved the scalability of the tool. Both the SER and the time required for the calculations were presented.

- Variations in threshold voltage due to inter-die variations resulted in SER variation of 16% for a 3σ variation of 10%. Threshold voltage variation due to intra-die variations resulted in a peak-to-peak SER variation of 41% in small circuits based on a small set of simulations. The effect of device degradation was not as large as the effect of static variations. Power supply variations showed a maximum variation in SER of 25% across different circuits, while the effect of temperature on SER variations was the highest especially for logic circuits with short data paths.

While this thesis demonstrated the contribution of a new methodology to model SER in logic circuits, more results could be obtained if further research was performed, using this work as a basis. Some areas that could be explored include a hierarchical methodology to model SER at larger block levels and the development of models from real time testing of microprocessors to gain more insights. This methodology could be applied to arithmetic units like adders, multipliers etc., which use some basic blocks which are small enough to be quickly characterized by the tool presented in this work. Using these characterizations, SER for much larger architectures can be predicted quickly. Accelerated testing is another area where future work is needed to formalize a methodology that can be used to verify all the existing tools that estimate SER.

References

1. Baze M., Buchner S., Attenuation of single event induced pulses in CMOS combinational logic, IEEE Transactions on Nuclear Science, Vol. 44, No. 6, December 1997, Page(s): 2217–2223.

2. Buchner S., Baze M., Brown D., McMorrow D., Melinger J., Comparison of error rates in combinational and sequential logic, IEEE Transactions on Nuclear Science, Vol. 44, No. 6, December 1997, Page(s):2209–2216.

3. Baumann R.C., Soft errors in advanced semiconductor devices-part I: the three radiation sources, IEEE Transactions on Device and Materials Reliability, Vol. 1 No. 1, March 2001, Page(s): 17–22.

4. Ziegler J., Terrestrial cosmic ray intensities, IBM Journal of Research and Development, Vol. 40, No. 1, January 1996, Page(s): 19–39.

5. Hareland S., Maiz J., Alavi M., Mistry K., Walsta S., Dai C., Impact of CMOS process scaling and SOI on the soft error rates of logic processes, Symposium on VLSI Technology Digest of Technical Papers, 2001, Page(s): 73–74.

6. Hazucha P., Svensson C. Impact of CMOS technology scaling on the atmospheric neutron soft error rate, IEEE Transactions on Nuclear Science, Vol. 47, No. 6, Dec. 2000, Page(s): 2586–2594.

7. Seifert N., Moyer D., Leland N., Hokinson R., Historical trend in Alpha-particle induced soft error rates of the Alpha microprocessor, Proceedings of 39th IEEE Annual International Reliability Physics Symposium, 2001, Page(s): 259–265.

8. Chung K. S, Kim T, Lin C.L., G-vector: A new model for glitch analysis, Proceedings of Twelfth Annual IEEE International ASIC/SOC Conference, Sept. 1999, Page(s): 159–162.

9. May T.C., Woods M. H., Alpha-particle induced soft errors in dynamic memories, IEEE Transactions in Electronic Devices, Vol. 26, No.1, 1979, Page(s): 2–9.

10. Shivakumar P., Kistler M., Keckler S., Burger D., and Alvisi L., Modeling the effect of technology trends on the soft error rate of combinational logic, Proceedings of International Conference on Dependable Systems and Networks, June 2002, Page(s): 389–398.

11. Massengill L. W., Baranski A. E., Nort D. O. V, Meng J, and Bhuva B. L. Analysis of Single-Event Effects in Combinational Logic – Simulation of the

AM2901 Bitslice Processor, IEEE Transactions on Nuclear Science, Vol. 47, No. 6, December 2000, Page(s): 2609–2615.

12. Nguyen H. T., Yagil Y, A Systematic approach to SER estimation and solutions, 41st Annual International Reliability Physics Symposium, April 2003, Page(s): 60–70.

13. Zhang M., Shanbhag N. R, A soft error rate analysis (SERA) methodology, Proceedings of International Conference on Computer Aided Design, Nov. 2004, Page(s): 111–118.

14. Zhao C., Bai X., Dey S., A scalable soft spot analysis methodology for compound noise effects in nano-meter circuits, Proceedings of 41st Design Automation Conference, June 2004, Page(s): 894–899.

15. Rajaraman R., Kim J. S., Vijaykrishnan N., Xie Y., Irwin M. J., SEAT-LA: A soft error analysis tool for combinational logic, Proceedings of the International Conference on VLSI Design, Jan. 2006, Page(s): 4.

16. Krishnaswamy S., Viamontes G. F., Markov I. L. and Hayes J. P., Accurate reliability evaluation and enhancement via probabilistic transfer matrices, Proceedings of Design Automation and Test in Europe (DATE), March 2005, Page(s) 282–287.

17. Dhillon Y. S., Diril A. U., Chatterjee A, Soft-error tolerance analysis and optimization of nanometer circuits, Proceedings of Design, Automation and Test in Europe, March 2005, Page(s): 288–293.

18. Zhang B., Wang W., Orshansky M., FASER: Fast Analysis of Soft Error Susceptibility for Cell-Based Designs, Proceedings of 7th International Symposium on Quality Electronic Design, March 2006, Page(s): 755–760.

19. Ramanarayanan R., Degalahal V., Vijaykrishnan N., Irwin M. J. and Duarte D., Analysis of soft error rate in flip-flops and scannable latches, Proceedings of the 16th Annual IEEE International ASIC/SOC Conference, September, 2003, Page(s): 231–234.

20. Degalahal V., Ramanarayanan R., Vijaykrishan N., Xie Y., Irwin M. J., The effect of threshold voltages on soft error rate, Proceedings of the 5th International Symposium on Quality Electronic Design, March 22-24, 2004, Page(s): 503–508.

21. Brent R. P., Kung H. T., A regular layout for parallel adders, IEEE Transactions on Computers, Vol. 21, 1982, Page(s): 260–264.

22. Kogge P. M., Stone H. S., A parallel algorithm for the efficient solution of recurrence equations, IEEE Transactions on Computers, Vol. 22, No. 8, 1973, Page(s): 260−264.

23. Predictive Tech. Model - http://www.eas.asu.edu/~ptm/.

24. Avant! Star-Hspice, User Manual, Release 2003.

25. Liu X., Papaefthymiou M.C., A statistical model of input glitch propagation and its application in power macromodeling, Proceedings of the 45th Midwest Symposium on Circuits and Systems, Vol. 1, Aug. 2002, Page(s): 380−383.

26. Omana M., Papasso G., Rossi D., Metra C., A model for transient fault propagation in combinatorial logic, Proceedings of 9th IEEE On-Line Testing Symposium, July 2003, Page(s): 111−115.

27 ITRS–2003, Critical reliability challenges for the International Technology Roadmap for Semiconductors, International Sematech Technology transfer 03024377A-TR, 2003.

28. Borkar S., Karnik T., Narendra S., Tschanz J., Keshavarzi A., De V., Parameter variations and impact on circuits and micro-architecture, Proceedings of Design Automation Conference, June 2003, Page(s): 338−342.

29. Bowman K. A., Duvall S. G., Meindl J. D., Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration, IEEE Journal of Solid-State Circuits, Vol. 37, No. 2, Feb. 2002, Page(s):183−190.

30. Borkar S., Parameter Variations and Impact on Circuits & Micro-architecture, C2S2 MARC0 review, March 2003.

31. Chandrakasan A., Bowhill W., F. Fox (Editors), Design of High-Performance Microprocessor Circuits, IEEE Press, 2000.

32. Tang X., De V., Meindl J., Intrinsic MOSFET parameter fluctuations due to random dopant placement, IEEE Transactions on Very Large Scale Integration (VLSI), Vol. 5, No. 4, Dec. 1997, Page(s): 369−376.

33. Agarwal A., Paul B. C., Mukhopadhyay S. and Roy K., Process variation in embedded memories: failure analysis and variation aware architecture, IEEE Journal of Solid-state Circuits, Vol. 40, September 2005, Page(s): 1804−1814.

34. Cheng Y., Rosenbaum E., Kang S., Improvement on chip-level electro-thermal simulator-ILLIADS-T, Proceedings of IEEE International Symposium on Circuits and Systems, Vol. 4, May 1996, Page(s): 432−435.

35. Power J. A., Clancy R., Wall W. A., Mathewson A., Lane W. A., An investigation of MOSFET statistical and temperature effects, Proceedings of the 1992 International Conference on Microelectronic Test Structures, March 1992, Page(s): 202−207.

36. Xuan X., Chatterjee A., Singh A. D., Local redesign for reliability of CMOS digital circuits under device degradation, Proceedings of International Reliability Physics Symposium, 2004, Page(s): 651−652.

37. Kumar S. V., Kim C. H., Sapatnekar S. S., Impact of NBTI on SRAM read stability and design for reliability, Proceedings of International Symposium on Quality Electronic Design, Mar. 2006, Page(s): 210−218.

38. Vattikonda R., Wang W., Cao Y., Modeling and minimization of PMOS NBTI effect for robust nanometer design, Proceedings of Design Automation Conference, Jul. 2006, Page(s): 1047−1052.

39. Srinivasan J., Adve S. V., Bose P., Rivers J. A., The impact of technology scaling on lifetime reliability, Proceedings of International Conference on Dependable Systems and Networks, 2004, Page(s): 177−186.

40. Zhou Q., Mohanram K., Gate sizing to radiation harden combinational logic, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 25, No. 1, Jan. 2006, Page(s):155−166.

41. Ajami A. H., Banerjee K., Pedram M., Scaling analysis of on-chip power grid voltage variations in nanometer scale ULSI, International Journal of Analog Integrated Circuits and Signal Processing, Vol. 42, No. 3, 2005, Page(s): 277−1290.

42. Chen H. H., Ling D.D., Power supply noise analysis methodology for deep-submicron VLSI chip design, Proceedings of Design Automation Conference, 1997, Page(s): 638.

43. Fang P., Tao J., Chen J. F., Hu C., Design in hot-carrier reliability for high-performance applications, Proceedings of IEEE Custom Integrated Circuits Conference, 1998, Page(s): 525−531.

44. Jean Y. S., Wu C. Y., The threshold-voltage model of MOSFET devices with localized interface charge, IEEE Transactions on Electron Devices, 1997, Page(s): 441−447.

45. http://www.eas.asu.edu/~ptm/reliability/PTM_NBTI_Beta.pdf.

46. http://www.micromagic.com/mmi_tools.html.

47. Degalahal V., Soft errors: Modeling and interactions with power optimizations, Thesis (Ph.D.), Computer Science and Engineering, Pennsylvania State University, 2005.

48. Lesea A., Drimer S., Fabula J., Carmichael C., Alfke P., The Rosetta experiment: Atmospheric soft error rate testing in differing technology FPGAs, IEEE Transactions on Device and Materials Reliability, Vol. 5, No. 3, Sept. 2005, Page(s): 301–302.

49. Hazucha P., Karnik T., Walstra S., Bloechel B., Tschanz J., Maiz J., Soumyanath K., Dermer G., Narendra S., De V., Borkar S., Measurements and analysis of SER tolerant latch in a 90 nm dual-Vt CMOS process, Proceedings of the IEEE 2003 Custom Integrated Circuits Conference, Sept. 2003, Page(s): 617–620

50. Daniels R.G., A participant's perspective, IEEE Micro, Vol. 16, No. 6, Dec. 1996, Page(s): 21–31.

51. JEDEC. Standard jesd89-measurement and reporting of alphaparticles and terrestrial cosmic ray-induced soft errors in semiconductordevices. (http://www.jedec.org/download/default.cfm).

52. Maiz J., Hareland S., Zhang K., Armstrong P., Characterization of multi-bit soft error events in advanced SRAMs, International Electron Devices Meeting, Technical Digest, 2003, Page(s): 21.4.1–21.4.4.

53. Hazucha P., Karnik T., Maiz J., Walstra S., Bloechel B., Tschanz J., Dermer G., Hareland S., Armstrong P., Borkar S., Neutron soft error rate measurements in a 90-nm CMOS process and scaling trends in SRAM from 0.25-micron to 90-nm generation, International Electron Devices Meeting, Technical Digest, 2003, Page(s): 21.5.1–21.5.4.

54. M68HC11E Family Data Sheet at: http://freescale.com/.

55. www.applieddata.net.

56. www.HVWTech.com.

57. http://www.intel.com/design/embeddedpca/products/pxa270/techdocs.htm.

Appendix A

Accelerated Testing for Measuring Neutron Induced Soft Errors

A.1 Introduction

Two of the known sources of radiation that cause soft errors are alpha particles and neutrons. Alpha particles can easily be shielded by using appropriate packaging materials. On the other hand, neutrons cannot be shielded, even with a few feet of concrete. Thus, our accelerated testing effort has focused mainly on neutron induced soft errors in memories, microcontrollers, and microprocessors.

There are different ways in which accelerated soft error rates can be measured. One set of experiments has used multiple set of devices at different altitudes to study the effects of atmospheric neutron on these devices [50]. However this becomes very costly, as a large number of devices are required to make enough measurements that are numerically significant to predict SER. Another way of measuring neutron induced SER is to expose the device to accelerated neutron radiation. Beam 30L for Weapon Neutron Research at the Los Alamos National Science Center (LANSCE) is a JEDEC prescribed test beam for soft errors [51] and it is the only one of its kind. This beam is highly stable and closely replicates the energy spectrum of terrestrial neutrons in the 2-800 MeV, range while providing a very high neutron flux. Most SER testing reported in recent literature was performed at this facility [52], [49], [6].

In our testing facility, we have two different setups for measuring soft error rates due to neutrons. This appendix mainly focuses on the soft error measurements done at this test set-up. We first present the different setups in which our experiments were done. Next, we discuss our experimental results and measurements for different devices, which included a memory chip, a microcontroller and a microprocessor.

A.2 Test Facility and Setup

The Penn State Breazeale Nuclear Reactor (BNR) in the Radiation Science and Engineering Center (RSEC) is the test facility at which the testing was done. The BNR is a 1 MW, TRIGA (Training, Research, Isotopes and General Atomics) nuclear reactor with moveable core in a large pool with pulsing capabilities. The maximum rated power of the reactor is 1 MW in the continuous mode and can be increased to 2000 MW in the pulse mode. However, for the soft error testing, the reactor was used in the continuous mode only. The reactor power can also be stepped from 10 W to 1 MW to observe the soft error rate dependence on neutron flux. The BNR has a variety of dry tubes called beam ports and fixtures available in or near the core. A pneumatic transfer system is also available for irradiation of samples. When the reactor core is placed next to the Deuterium Oxide (D_2O) or heavy water tank and the graphite reflector assembly is placed near the beam port locations, neutron beams become available from two of the seven existing beam ports.

Neutrons are classified based on their energies. Neutrons with an energy more than 0.1 MeV are called fast neutrons and those with energy less than that are called epithermal and thermal neutrons. The atmospheric neutron flux has both of these components, though neutrons at lower energies are far numerous than higher energy neutrons. The neutron flux emitted from the reactor core has both fast and thermal components. Near the reactor core, the neutron flux is dominated by fast neutron. The fast neutron flux near the core can be accessed through long tubes (standpipes). Since this fast neutron flux from the reactor core passes through the heavy water tank, the resultant neutron flux at the output of the beam port is that of thermal neutrons.

In a steady state of operation at 1 MW, the thermal neutron flux is 1x10^{13} n/cm^{2}s at the edge of the core and 3x10^{13} n/cm^{2}s at the central thimble. The average thermal flux at the exit of the beam port is 3x10^7 n/cm^{2}s. The neutron spectrum of this beam port is measured with a slow neutron chopper as shown in Figure **A.1**. The spectrum measurement is compared with a corresponding Maxwell Boltzmann distribution. Since the generation of the neutrons is a statistical phenomenon, isolating the neutrons of a given energy is usually difficult. However, the facilities available at RSEC, namely, the ability to operate at multiple power levels, pulse the reactor to produce an extremely high power range and isolate the thermal and fast neutron flux. This permits effective characterization of the soft error rate.

The fast neutron spectrum is currently being calibrated. The upper energy limit of fast neutron flux from the reactor is less than that of the atmosphere's neutrons spectrum, but it corresponds to the dominant portion of the atmosphere neutron flux.

Initial measurements of memory circuits SERs were previously completed and a detailed description of the facility and the test set-up can be found in [**47**]. Also, in [**47**], it is shown that the memory error rate increases linearly with an increase in the reactor power, thus confirming the validity of accelerated tests, especially in high capacity chips. The following two sub-sections briefly describe the two different test setups that were used for this thesis.

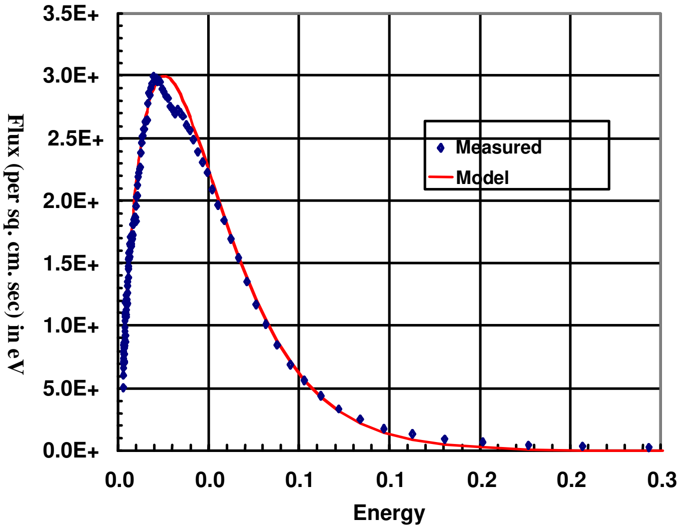


Figure A.1: Thermal neutron energy distribution at the exit of a horizontal neutron beam port at BNR [47]

A.2.1 Test setup at the outer neutron port

The first setup in which the soft error measurements were done was at the outer neutron beam port. Figure A.2 shows both the simplified schematic and the test set-up at the beam port respectively. As shown, an epoxy shield is used to protect the other portions of the board thus exposing only the test chip to thermal neutron radiation. Using this test setup, SER measurements for a microcontroller and a microprocessor were separately done. The microcontroller that was tested was Motorola-based M68HC11 [54]. Also, a PXA270, a strong arm based processor, on a BitsyXb embedded system from

Applied Data Systems [55], was tested for SER measurements using this setup. The next few sections present the results for these experimental measurements.

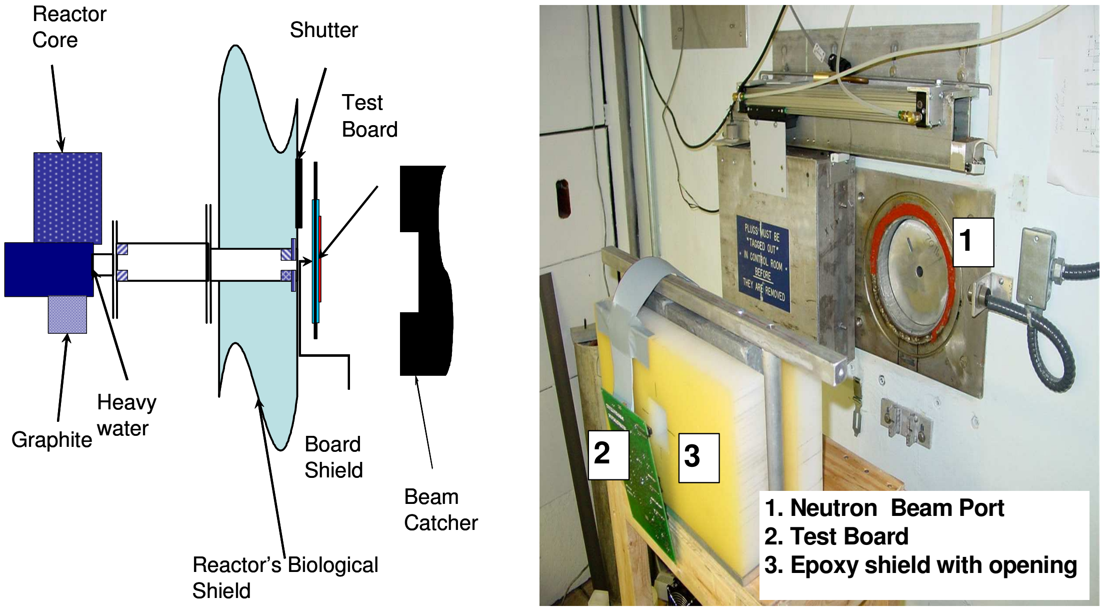


Figure A.2: Simplified schematics of beam port and setup for testing the board

A.2.2 Test setup at the reactor core

It was already known that the neutron flux close to the reactor core is much higher than that at the neutron beam ports. Therefore, the circuit board for this setup was placed near the periphery of the reactor core via a vertical standpipe, in order to observe the effect of fast neutrons on soft error rate. A schematic drawing of fast neutron irradiation facility at BNR is shown in Figure A.3. At 1 MW steady state of reactor operation, the fast neutron flux at the core boundary is 5x10^{12} n/cm^{2}s. The reactor can be pulsed for a very short duration of time, around 10 milli-seconds at its Full-Width Half-Maximum, during which it generates a fast flux of about 1X10^{16} n/cm^{2}s at the core periphery. This amounts to about a ten order of magnitude increase in the fast flux. The time duration is very limited, yet the amount of fast flux is immense. The DUT is put inside the standpipe and the reading is taken. In addition, the walls of the pipe are covered with cadmium. Cadmium absorbs the thermal component of the flux, so that only fast neutrons affect the board. The memory boards tested at the outer beam port in [47] were tested for SER measurements using this setup, and the results are presented in the following section.

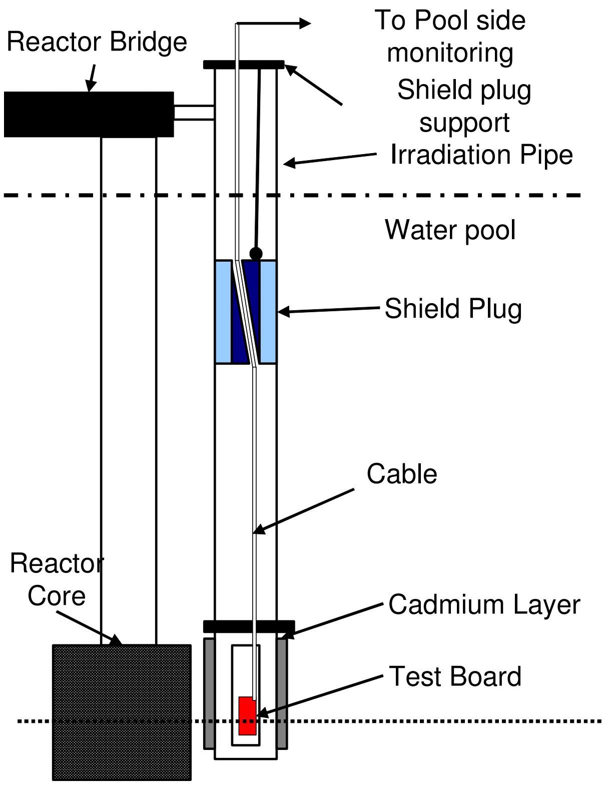


Figure A.3: Test setup at the reactor core

A.3 Experimental Results

Several types of test boards were used to measure SER using the two different setups mentioned earlier. This section presents the results for the various SER measurements done at the two different setups. This includes the testing of M68HC11 microcontroller and the PXA270 based BitsyXb system at the outer beam port and also the memory inside the core of the reactor.

A.3.1 Testing M68HC11 microcontroller

The M68HC11 microcontroller was evaluated with a prototyping circuit board from HVW technologies [56]. As a prototyping system it is a quick and efficient means of testing code and external circuitry. Here, the setup shown in Figure A.2 was used wherein only the microcontroller chip was directly exposed to the neutron radiation.

The 68HC11 is 8-bit data, 16-bit address microcontroller from Motorola, with an instruction set that is similar to the older 68xx (6801, 6805, 6809) parts. Depending on the particular model number, the 68HC11 has built-in EEPROM/OTPROM, RAM, digital I/O, timers, A/D converter, PWM generator, and synchronous and asynchronous communications channels (RS232 and SPI). For these tests, the E9 series was used. The M68HC11 is optimized for low power consumption and high-performance operation at bus frequencies up to 4 MHz. The CPU has two 8-bit accumulators (A & B) that can be concatenated to provide a 16-bit double accumulator (D). Two 16-bit index registers are available (X & Y) to provide indexing to anywhere in the memory map. More details on the architecture and programming of the microcontroller can be found in [54].

A RS232 serial cable was used along with the Windows HyperTerminal program for communication with the Buffalo 3.4 operating system loaded on the microcontroller. The microcontroller has an on-chip SRAM. The program instructions for the microcontroller that were used to control the microcontroller were stored in the portion of the SRAM available for the user (address space 0000-0050 Hex). The data used by the program were stored in the 512-byte on-chip EEPROM. After some initial tests, two different testing scenarios were devised by using the following approach: (1) Provide repeated use of a single mathematical operation for a number of times. (2) Provide sufficient delay for the result of the computation to be stored in the registers. (3) Perform the operations on data that was stored in the on-chip memory, so that any errors in the memory would be apparent in the output of the program. (4) Transmit the data through the serial port. (5) Provide delay cycles to make sure that the read-write time was negligible compared to the time for which the data was operated/stored in the microcontroller.

The pseudo-codes for these testing scenarios are presented in Table A.1. The first one was used to read data from the EEPROM and perform additions before outputting the data to the serial port to be captured by the HyperTerminal program. The second code was used to perform similar operations on the SRAM memory. Both versions had a delay loop which was used to make the read-write time insignificant. Note that both memory types were exhaustively read from and the data was manipulated by the program. No errors were found while each program was run for 150 minutes each. The reason for this could be due to the high voltage (5V) and capacitances of the chip, leading to no memory flips (or pulse strong enough to be latched in the accumulator) since the microcontroller was fabricated in 2 micron HCMOS technology [50]. Also, since the accumulators and data path occupy only a very small area in the microcontroller, the neutron hit probability is also small.

Table A.1: Psuedo-codes for Testing M68HC11 microcontroller

Addition of EEPROM Memory	Testing the SRAM data
scdr EQU $102f	scdr EQU $102f
scsr EQU $102e	scsr EQU $102e
base EQU $1000	base EQU $1000
eepr EQU $b600	sram EQU $0000
org $0000	org $0000
loop: ldx #base	loop: ldx #base
ldy #eepr	ldy #sram
lop3: ldaa $0,y	lop5: ldaa $0,y
ldab #0	anda $0F
lop4: adda $0,y	adda #65
iny	staa <scdr,x
incb	her1: brclr <scsr,x $40 her1
cmpb #9	iny
bne lop4	pshy
pshy	ldy #0
ldy #0	lop1: iny
lop1: iny	cpy #$FFFF
cpy #$FFFF	bne lop1
bne lop1	puly
puly	cpy #$FF
adda #47	bne lop5
xmi1: staa <scdr,x	bra loop
her1: brclr <scsr,x $40 her1	
suba #47	
cpy #$b7f8	
bne lop3	
bra loop	

A.3.2 Testing PXA270 processor

The accelerated testing of the microcontroller was just a steppingstone for testing more advanced and complicated processors. Thus, our next step was to test a PXA270 microprocessor at the outer port of the reactor. The PXA270 is an Arm-based microprocessor with voltage scaling capabilities to reduce the power requirements and is fabricated at or below 180 nanometer technology. Hence, the probability of errors occurring in this type of processors is much higher, as our experiments on this processor confirmed. More information on the PXA270 processor architecture is in [57].

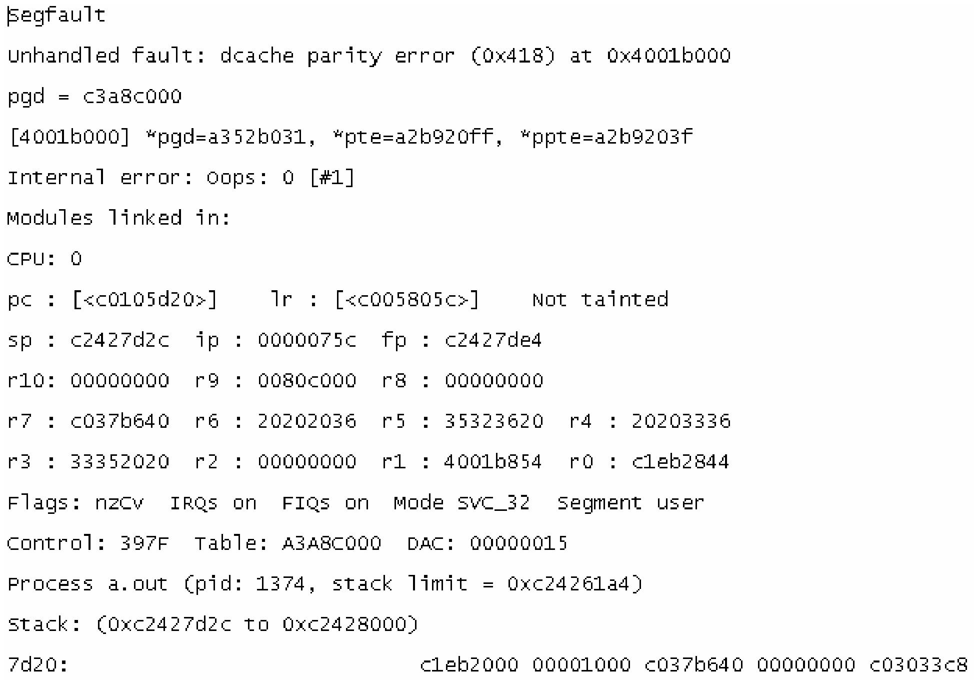
The processor that was tested was part of a BitsyXb Single Board Computer (SBC) from Applied Data Systems [55]. The specification details for the SBC can be obtained from [55]. Since the on-board memory was very small (~64 MB), an elementary version of Debian Linux OS was installed. Hence, an external memory to install the full root system of the operating system was required, so that C programs, which were used to test the processor, could be compiled and run on the system. Therefore, the complete version of the Debian Linux OS was installed on a 1GB USB flash memory and the initial exposure tests were done. Even though the PXA270 chip was the only portion of the board directly exposed to the neutron radiation, the flash memory was in the vicinity of the beam and could also be affected. To avoid this problem, a USB based hard drive was later procured and the full root system of Debian Linux OS was installed on it. In the following paragraphs, the details and the results of the experiments are presented.

The microprocessor architecture is much more complicated than the micro-controller that was tested earlier, making it far more difficult to write programs that could utilize the memory. Thus, a simple matrix multiplication program was compiled and run on the processor, and SER measurements were made during the exposure to neutron radiation. In these programs, first the source matrices were created, then a suitable delay was introduced, after which the actual multiplication of the two source matrices was done. The source and the result matrices were then stored in a file. Two identical matrix multiplications were performed one after the other, and then the results were compared for errors. The delay values were chosen such that the read/write time of the file transfers was negligible. Four preliminary experimental runs were performed and the results are presented in Table A.2. Here a USB flash drive was used for Test 1 and Test 2, while a hard drive was used for Tests 3 and 4. Both Tests 2 and 4 were predominantly run at scaled down voltage and speed.

Table A.2: Preliminary Test Results

	Test 1	Test 2	Test 3	Test 4
Run Time (minutes)	106	154	87	143
Total Errors	8	16	10	25
Segmentation + bus errors	1 + 0	5 + 0	2 + 0	9 + 5
Crashes during boot	1	2	1	2
Irrecoverable crashes	6	5	4	9
Other errors	0	4	3	0
Other errors include connection lost, debug port crashes only, file system corruptions etc				

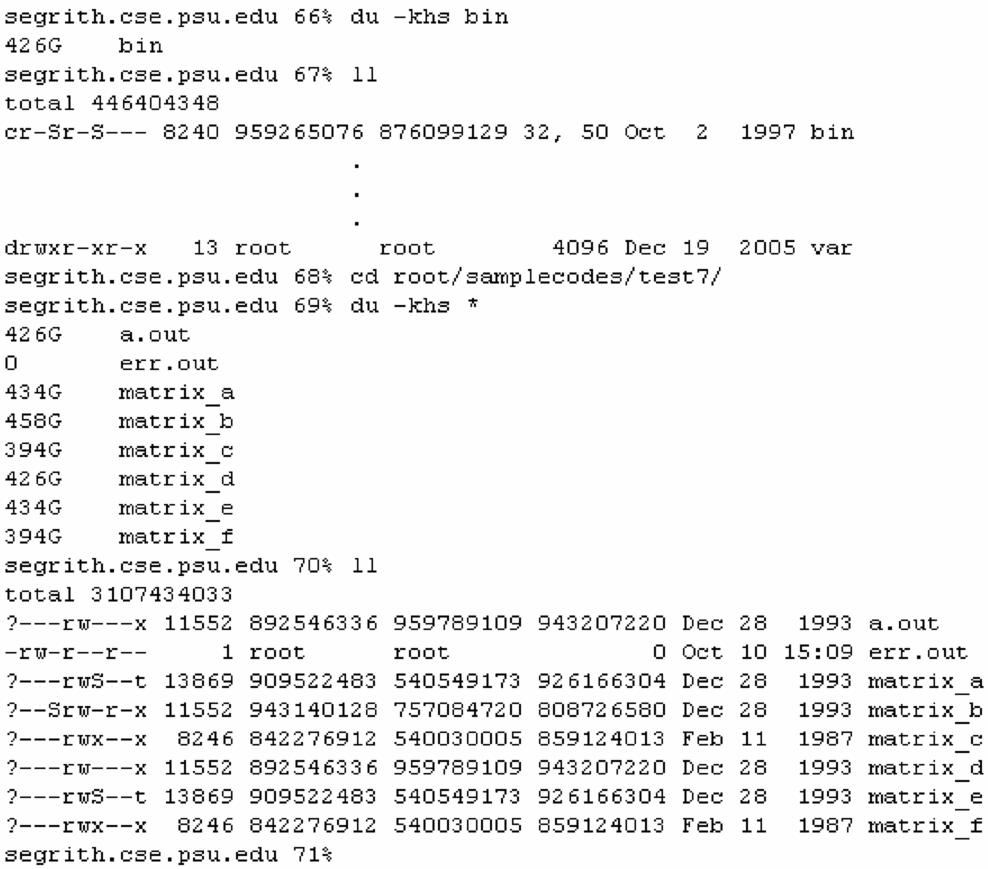
As seen from the table, apart from the irrecoverable crashes, segmentation faults are the most common types of failures that occur. These segmentation faults and also many of the irrecoverable crashes are caused due to an error called d-cache parity error. Figure A.4 shows the debug port message when this error occurs. As the name indicates, this error occurs due to a difference in parity while reading back the information from the memory, indicating that a bit flip that could have occurred in the data cache. The corruption of the data is thus prevented, because of the inherent property of the kernel to issue segmentation faults when this error occurs.



```
Segfault
Unhandled fault: dcache parity error (0x418) at 0x4001b000
pgd = c3a8c000
[4001b000] *pgd=a352b031, *pte=a2b920ff, *ppte=a2b9203f
Internal error: Oops: 0 [#1]
Modules linked in:
CPU: 0
pc : [<c0105d20>]    lr : [<c005805c>]    Not tainted
sp : c2427d2c  ip : 0000075c  fp : c2427de4
r10: 00000000  r9 : 0080c000  r8 : 00000000
r7 : c037b640  r6 : 20202036  r5 : 35323620  r4 : 20203336
r3 : 33352020  r2 : 00000000  r1 : 4001b854  r0 : c1eb2844
Flags: nzCv  IRQs on  FIQs on  Mode SVC_32  Segment user
Control: 397F  Table: A3A8C000  DAC: 00000015
Process a.out (pid: 1374, stack limit = 0xc24261a4)
Stack: (0xc2427d2c to 0xc2428000)
7d20:                            c1eb2000 00001000 c037b640 00000000 c03033c8
```

Figure A.4: D-cache parity error

Another type of failure which could have occurred, due to a hardware logic error, resulted in file system corruption as shown in Figure A.5. This kind of failure occurs when the ‘inode’ fields for different files are updated on to the USB memory. These ‘inode’ fields store file information such as size, type, date of creation etc. As each file in the USB memory is being modified when a process is running, the processor updates the information in these ‘inode’ files. Thus, the error could have occurred in the ‘inode’ fields, which could be due to a logic error, as the errors in the cache lead to segmentation faults and hence are prevented. It was also noted that these file system corruptions occurred when hard drives were used as the USB memory, which could be due mainly to a relatively larger number of write backs/updates of ‘inode’ fields in the case of hard drives.



```
segrith.cse.psu.edu 66% du -khs bin
426G    bin
segrith.cse.psu.edu 67% ll
total 446404348
cr-Sr-S--- 8240 959265076 876099129 32, 50 Oct  2  1997 bin
                        .
                        .
                        .
drwxr-xr-x   13 root      root        4096 Dec 19  2005 var
segrith.cse.psu.edu 68% cd root/samplecodes/test7/
segrith.cse.psu.edu 69% du -khs *
426G    a.out
0       err.out
434G    matrix_a
458G    matrix_b
394G    matrix_c
426G    matrix_d
434G    matrix_e
394G    matrix_f
segrith.cse.psu.edu 70% ll
total 3107434033
?---rw---x 11552 892546336 959789109 943207220 Dec 28  1993 a.out
-rw-r--r--     1 root      root              0 Oct 10 15:09 err.out
?---rwS--t 13869 909522483 540549173 926166304 Dec 28  1993 matrix_a
?--Srw-r-x 11552 943140128 757084720 808726580 Dec 28  1993 matrix_b
?---rwx--x  8246 842276912 540030005 859124013 Feb 11  1987 matrix_c
?---rw---x 11552 892546336 959789109 943207220 Dec 28  1993 matrix_d
?---rwS--t 13869 909522483 540549173 926166304 Dec 28  1993 matrix_e
?---rwx--x  8246 842276912 540030005 859124013 Feb 11  1987 matrix_f
segrith.cse.psu.edu 71%
```

Figure A.5: File system corruption

A single test run was also performed for the memory board that was used in [47] at the reactor core using the standpipe setup described earlier (Figure A.3). After a 10-minute exposure at 900 MW of reactor power, data was read back from the memory. The data read back from all the memory locations were zeros. The errors were persistent even after repeated read-writes following the exposure. Thus permanent damage was caused by the thermal neutrons present in the radiation, and precautions are being taken to avoid such damages for the future.

To summarize, the work presented in this appendix is a preliminary work to study the effect of technology and voltage scaling on processors. Further effort is needed to use the results obtained from such accelerated testing for enhancement and the verification of the various models that are currently used to predict soft errors including the tool presented in this thesis.

Appendix B

Circuit Designs

In Chapters 2 and 3, various flip-flop and logic circuits used standard gates for which the design details are now presented. All the designs were laid out using the Micro Magic MAX tool [46] and extracted to get the spice net list with 70 nm Predictive Technology Models (earlier - Berkley Predictive Technology Models) [23]. First, Figure B.1 shows the schematic of the Transmission Gate Flip-Flop that was designed to work at 1 V and 1 GHz. The widths in micrometers for each of the devices are shown and the lengths of all the devices were 0.07 micrometer (70 nanometer).

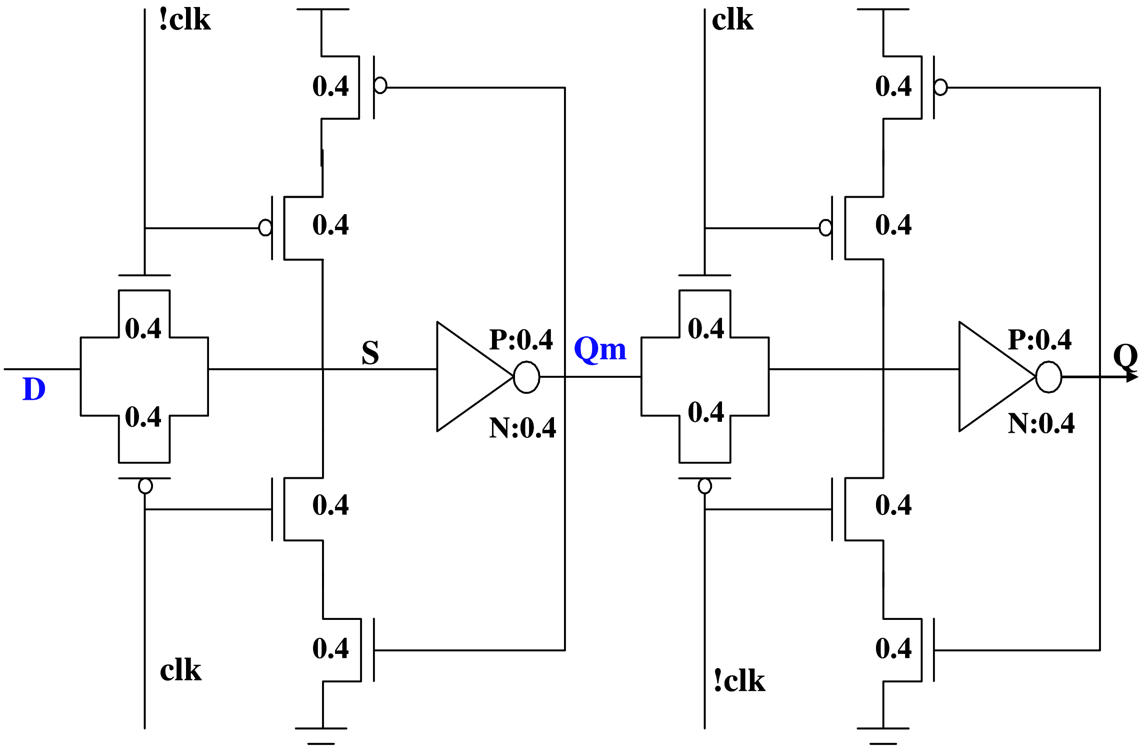


Figure B.1: Transmission Gate Flip-Flop (TGFF)

Next, the detailed schematics with device sizes of a C^{2}MOS and the Semi-Dynamic (SDFF) flip-flop are shown in Figure B.2 and Figure B.3. In the SDFF, one of the NMOS devices in the two inverter feedbacks loops has a device length of 0.13 micrometer (denoted by N:0.3/0.13) L to make that inverter weaker than the other for the proper operation of the flip-flop.

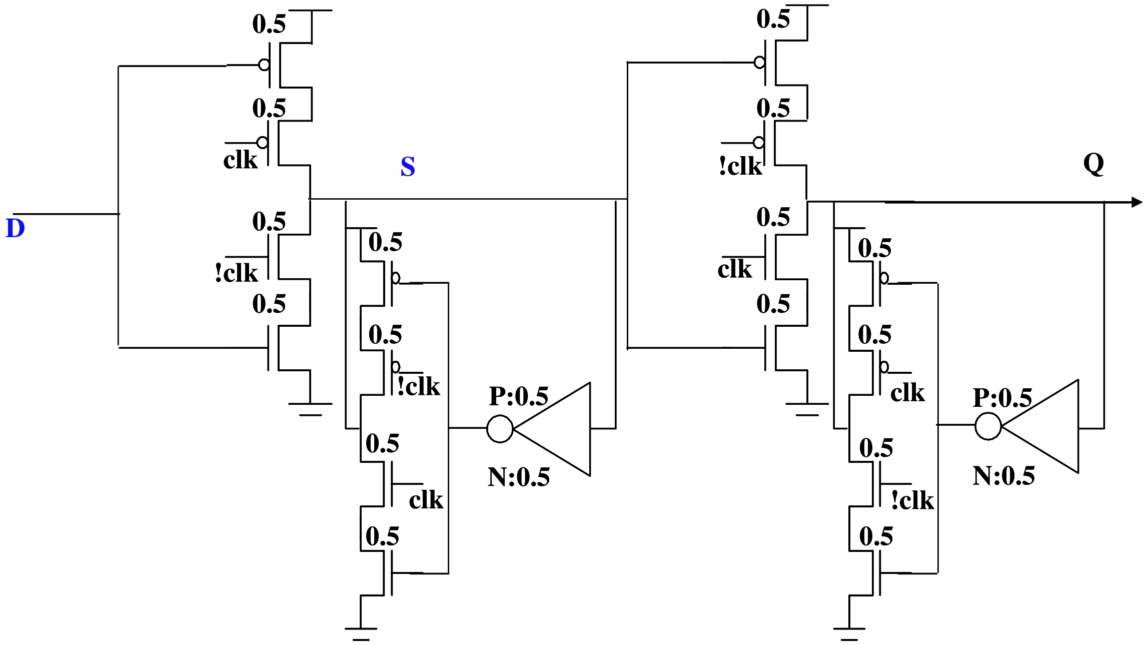


Figure B.2: C^{2}MOS flip-flop

All the logic circuits considered and analyzed in this work were built using the four basic types of gates (Inverter, NAND, NOR and XOR), and Figure B.4 shows the sizing of the devices in these gates that were used uniformly for all designs.

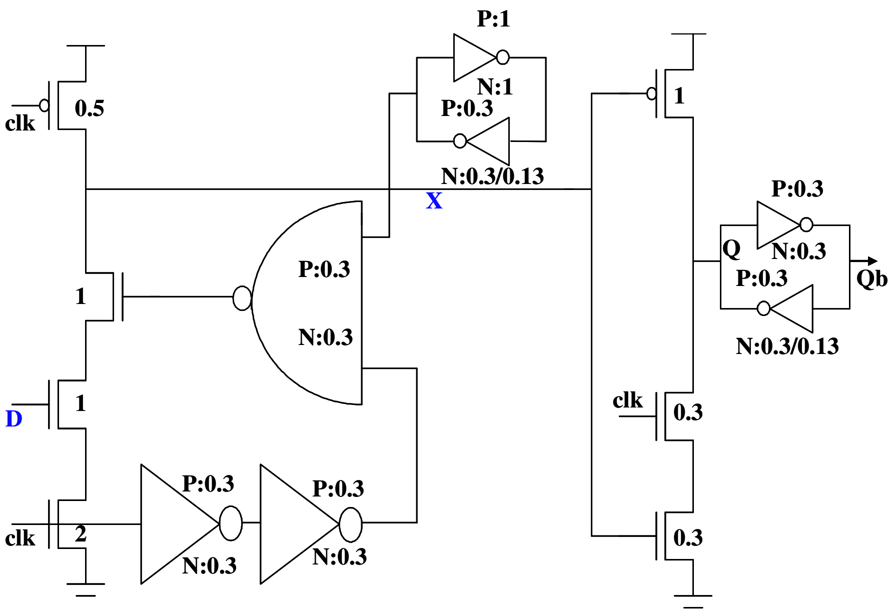


Figure B.3: Semi-dynamic flip-flop

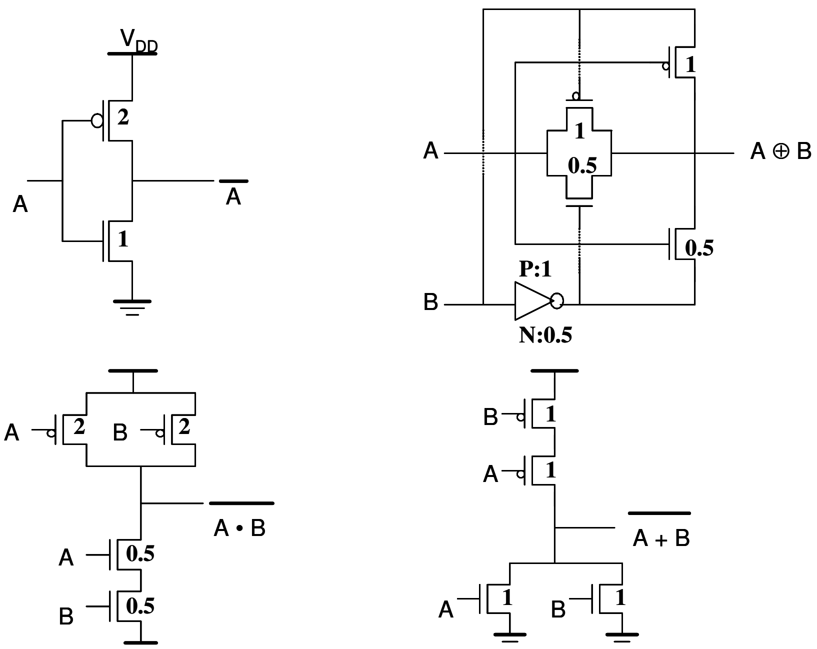


Figure B.4: Sizing for basic gates

VITA

Rajaraman Ramanarayanan

Rajaraman Ramanarayanan obtained the Bachelor of Engineering degree with distinction in Electrical and Electronics Engineering from Annamalai University, Annamalai Nagar, India in 2001. He obtained his Master of Science degree in Electrical Engineering from The Pennsylvania State University in 2003. He joined the Ph.D. program in the Department of Electrical Engineering at Penn State in August 2003. During this period he has been employed as both a research assistant and a teaching assistant at the Departments of Electrical and Computer Science Engineering. He was also a summer intern at Intel Corporation, Hillsboro, OR for three months in 2004. Currently, he is employed as a research scientist at the Circuit Research Laboratory in Intel Corporation. Rajaraman is also a student member of the IEEE and has reviewed papers for many conferences and journals including: TVLSI, DATE, ISVLSI, GLVLSI, SIPS, ISCAS and ISVLSI. He has also published papers in various conferences and journals.