Design Techniques for Power-Aware Combinational Logic SER Mitigation

By

Nihaar N. Mahatme

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

December, 2014

Nashville, Tennessee

Approved:

Dr. Bharat Bhuva

Dr. Lloyd Massengill

Dr. Anthony Oates

Dr. Robert Reed

Dr. Ronald Schrimpf

ACKNOWLEDGMENT

Isaac Newton is quoted to have said that “If I have seen further than others it is by standing on the shoulder of giants”. I owe many thanks to the entire scientific community working tirelessly in the field of microelectronics for the success of this work. Equally I would like to acknowledge the personal strength and solitude I have derived from the support of my grandparents and parents.

I have been privileged to have been under the tutelage of many wise teachers. Primus inter pares among them is my graduate advisor Dr. Bharat Bhuva. I cannot thank him enough for all the encouragement and calming influence even in cramped labs at unearthly hours throughout the past five years. He has always challenged me to strive for more than I thought I could. Professors Massengill and Schrimpf have served as great critics providing fresh perspectives and ideas and I owe a lot to them. I would also like to thank Dr. Reed for his useful inputs. The unwavering devotion towards research of Dr. Anthony Oates, my mentor at TSMC, has been a great source of inspiration. I would also like to thank professors Robinson, Fleetwood and Witulski with whom I have collaborated over various projects in the past. The patience that Dr. Robinson has shown in answering all my questions is much appreciated. Lastly, I would like to thank all my teachers from school especially Ms. Athaide, Chande, Pereira and Fernandes who laid the foundation for all my future achievements. Professors Kennedy and Bopardikar for teaching me that problem-solving can be fun and my mentors Alessio Griffoni, Dimitri Linten and Glenn Abeln for the fun summer internship experiences. I would like to thank Daniel Loveless, Zac, Nelson and Srikanth without whom critical parts of this work would have been incomplete.

Finally, thanks to Jon, Mike, Daniel, Indranil, Hamzah, Thiago, Cher, Tania, Jug, Enxia, Pierre, Vijay, Zac, Balaji, Sandeepan, Sayan, Shubhajit, Akash, Atul, Karan, Sourabh, Tushar, Nikhil, Karthik, Ashish and Arpun for help me stay sane all these years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The history of modern semiconductor devices and circuits suggests that technologists have been able to maintain scaling at the rate predicted by Moore’s Law [Moor-65]. With improved performance, speed and lower area, technology scaling has also exacerbated reliability issues such as soft errors. Soft errors are transient errors that occur in microelectronic circuits due to ionizing radiation particle strikes on reverse biased semiconductor junctions. These radiation induced errors at the terrestrial-level are caused due to radiation particle strikes by (1) alpha particles emitted as decay products of packing material (2) cosmic rays that produce energetic protons and neutrons, and (3) thermal neutrons [Dodd-03], [Srou-88] and more recently muons and electrons [Ma-79] [Nara-08] [Siew-10] [King-10]. In the space environment radiation induced errors are a much bigger threat and are mainly caused by cosmic heavy-ions, protons etc. The effects of radiation exposure on circuits and measures to protect against them have been studied extensively for the past 40 years, especially for parts operating in space. Radiation particle strikes can affect memory as well as combinational logic. Typically when these particles strike semiconductor junctions of transistors that are part of feedback structures such as SRAM memory cells or flip-flops, it can lead to an inversion of the cell content. Such a failure is formally called a bit-flip or single-event upset (SEU). When such particles strike sensitive junctions part of combinational logic gates they produce transient voltage spikes or glitches called single-event transients (SETs) that could be latched by receiving flip-flops. As the circuits are clocked faster, there are more number of clocking edges which increases the likelihood of latching these transients. In older

technology generations the probability of errors in flip-flops due to SETs being latched was much lower compared to direct strikes on flip-flops or SRAMs leading to SEUs. This was mainly because the operating frequencies were much lower for older technology generations. The Intel Pentium II for example was fabricated using 0.35 μm technology and operated between 200-330 MHz. With technology scaling however, operating frequencies have increased tremendously and the contribution of soft errors due to latched SETs from combinational logic could account for a significant proportion of the chip-level soft error rate [Sief-12][Maha-11][Shiv02] [Bu97]. Therefore there is a need to systematically characterize the problem of combinational logic single-event effects (SEE) and understand the various factors that affect the combinational logic single-event error rate.

Just as scaling has led to soft errors emerging as a reliability-limiting failure mode for modern digital ICs, the problem of increasing power consumption has arguably been a bigger bane of scaling. While Moore's Law loftily states the blessing of technology scaling to be smaller and faster transistor it fails to highlight that the power density increases exponentially with every technology generation. The power density problem was partially solved in the 1970's and 1980's by moving from bipolar and GaAs technologies to full-scale silicon CMOS technologies. Following this however, technology miniaturization that enabled high-speed, multicore and parallel computing has steadily increased the power density and the power consumption problem. Today minimizing the power consumption is as much critical for power hungry server farms as it for portable devices, all pervasive sensor networks and future eco-bio-sensors. Low-power consumption is now regularly part of design philosophies for various digital products with diverse applications from computing to communication to healthcare.

Thus designers in today’s world are left grappling with both a “power wall” as well as a “reliability wall”. Unfortunately, when it comes to improving reliability through soft error mitigation, most approaches are invariably straddled with overheads in terms of area or speed and more importantly power. Thus, the cost of protecting combinational logic through the use of power hungry mitigation approaches can disrupt the power budget significantly. Therefore there is a strong need to develop techniques that can provide both power minimization as well as combinational logic soft error mitigation. This dissertation, advances hitherto untapped opportunities to jointly reduce power consumption and deliver soft error resilient designs. Circuit as well as architectural approaches are employed to achieve this objective and the advantages of cross-layer optimization for power and soft error reliability are emphasized.

Key Research Contributions

1. **Identification of key factors that affect technology scaling trends of combinational logic soft errors**. Sensitive area, single-event transient pulse-width, drive currents and operating frequency are identified as key factors that affect the combinational logic soft error reliability. Understanding these factors is critical for designers to estimate the relative contribution of combinational logic, flip-flop and memory soft error rate at the chip-level, especially for future technology nodes. Using this information the most efficient hardening approaches can be adopted. The experimental scaling trends for soft errors has

eluded researchers for last several years and limited estimates were derived through models and simulations [Shiv-02], [Buch-97]. This work comprehensively analyses the impact of scaling at 40 nm, 28 nm and 20 nm bulk technology nodes.

2. **Development of power-aware techniques with minimal performance overheads to mitigate combinational logic soft errors**. Modern integrated-circuit design emphasizes low-power design approaches. This dissertation presents, for the first time, formal approaches to identify key factors that can reduce both, the power consumption and combinational logic error rate. Different approaches at the gate-level, circuit-level and architectural-level and the associated trade-offs and penalties are presented. A cross-layer approach to co-optimizing for combinational logic soft-error reduction and power minimization is emphasized. The pitfalls of optimizing both variables independently is also highlighted.

These contributions extend the ability of designers to estimate and improve the reliability of circuits in the presence of radiation induced soft-errors, even in the case of large and complex circuits. Using some of the results and techniques presented in this work a designer can compare the relative contribution of flip-flop and logic errors for large circuits for future generations as well. Based on this comparison designers can adopt the most efficient hardening approaches. In this exercise, the results presented in this work present designers options to improve the soft-error reliability of combinational logic circuits while also minimizing power consumption. These results also emphasize that optimization across a variety of levels is necessary to achieve maximum reduction of combinational logic soft errors.

Thesis Organization

The subsequent discussion in this thesis is organized as follows.

1. Chapter 1 provides a background on radiation effects in digital integrated circuits and combinational logic circuits in particular. The different factors that influence the combinational logic soft-error sensitivity and the different circuit-level masking factors that reduce their impact are discussed. A variety of approaches to mitigate combinational logic soft errors and their associated power penalties are also discussed **[Master's Thesis Work** (included in detail as Appendix I)**]**.
2. Chapter 2 includes a discussion on commonly used low-power design techniques at various levels of design abstraction. The relative merits and demerits of these strategies are discussed and their potential impact on logic SER is also discussed. Some of the key factors that could be identified for co-optimization of logic SER reduction and power minimization are identified.
3. **[Contribution 1]** Chapter 3 presents experimental results that characterize logic soft errors as a function of frequency for three different technology generations (40 nm, 28 nm and 20 nm CMOS bulk technology nodes) and presents technology scaling trends.

4. **[Contribution 2]** Chapters 4, 5 and 6 present circuit level techniques to reduce power consumption and mitigate soft errors. The common thread across these chapters is the idea that reducing the effective number of switching nodes can reduce the power consumption as well as reduce the combinational logic soft error rate.
5. **[Contribution 2]** In chapter 7, an architectural technique like pipelining for low power is employed to deal with the problem of combinational logic errors efficiently. The idea is to pipeline circuits to maintain constant throughput and reduce operating voltage. The reduced operating voltage results in lower power and permits the inclusion of slower and harder logic to mitigate combinational logic soft errors.
6. In summary, Chapter 8 highlights the key contribution of this work and different conditions under which the above techniques have limited application are also identified. Future directions in the rich unexplored area of power-aware design for soft error reliability are suggested.
7. Chapters 9, 10 and 11 are appendices which include important results about 1) gate level logic soft error mitigation; 2) impact of supply voltage variation on the logic soft error rate and 3) fast estimation of combinational logic SER for large circuits.

Chapter I. Single Event Effects in Combinational Logic Circuits and Mitigation Approaches

Microelectronic circuits operating in harsh environments such as space, are exposed to cosmic radiation that can cause failures in integrated circuit (IC) operation. Early evidence of these effects emerged through in-flight observation of single-event upsets (SEUs) in memories [Bind-75], [Pick-78]. A few years later, single-events errors or soft errors at the terrestrial level were reported in groundbreaking work where alpha particles in packaging material were reported to be the cause of upsets in dynamic random access memories (DRAMs) [May-79]. As technologies have scaled, the problem of soft errors has been exacerbated for digital circuits used for high-reliability applications in space environment as well as terrestrial environment. More recently, experimental results have shown that SRAMs have now become sensitive enough to be upset by particles like neutrons, muons and electrons [Hazu-00], [Siew-10], [King-13].

On the other hand, the generation of soft errors in flip-flops due to the generation and latching of single event transients (SETs) in combinational logic is also becoming a significant reliability challenge in modern CMOS ICs. First observation of single-event transients appeared in the 1980's with the development of radiation-hardened space-grade microprocessors [Harb-86] [Koga-85]. However the importance attached to soft errors created due to latched transients received limited attention for several years. The main reason for this is that the operating frequency of semiconductor ICs in the 80's and 90's were in the few hundreds of MHz at best. As a result the probability of soft errors due to SETs that are latched compared to direct upsets in latches and memories was very low. Researchers have suggested that the number of soft errors caused due to latched transients from logic would exceed soft errors caused due to SEUs in flip-flops and unprotected memory elements at advanced technology nodes [Bu-97] [Sh-02]. Today, there is considerable interest in ascertaining whether soft errors due to SETs from combinational logic could exceed the raw flip-flop and memory error rate, and in understanding the future technology trends in this regard. Understanding these trends would influence hardening strategies for future generations of high-speed digital circuits employed in different environments. Some of the key mechanisms and findings related to how single-event particle strikes cause errors in memories as well as combinational logic circuits are presented in this chapter. The key topics discussed in this chapter are as follows:

1. Soft errors in memories and flip-flops.
2. Soft errors in combinational logic circuits
    a. Temporal Masking
    b. Electrical Masking
    c. Logical Masking
3. Previous results related to logic soft errors
4. Combinational logic soft error mitigation and overheads

Single event effects occur when a radiation particle strikes semiconductor junctions. When an ionizing particle passes through a reverse-biased junction it creates electron-hole pairs which results in a transient current at that node when some of the generated carriers are collected. This is illustrated in Figure 1-1. Consider an NMOS transistor with drain biased high. Following the strike, electron hole pairs are generated along the particle track. As the junction is reverse biased, negative electrons are swept towards the terminal. Holes on the other hand slowly diffuse towards the other terminal. This leads to a fast current transient at the output node. The collected charge manifests itself as voltage perturbation at the struck node [Mass-93], [Dasg-07]. Previous works have explained the transient charge collection and recovery of reverse biased junctions [Mass-93], [Dodd-03]. However as far digital circuits are concerned, the presence of a restoring device changes the voltage response at the struck node. The voltage perturbation and current pulse profile in the case where a stand-alone NMOS transistor is struck is different from the case when the NMOS is part of an inverter for example. This is shown in Figure 1-2. When the reverse biased drain region of a stand-alone NMOS is struck the current response is characterized by a sharp peak due to drift charge collection followed by a slow tail from diffusion charge collection. The initial fast rise time of the current is still dominated by drift-collection, the slowly falling component is dominated by diffusion currents and is characterized especially by the presence of a plateau for higher-LET particle strikes [Dasg-07], [Nara-08]. The electric field plays an important role in the drift component and a higher electric field at the reverse biased junction would mean greater collected charge. The restoring transistor drive helps restore the struck node to its initial value and is largely responsible in dissipating the charge deposited by the ion leading to voltage recovery at the struck node. Thus the strength of the restoring transistor drive influences the single event transient pulse-width [Dasg-07].

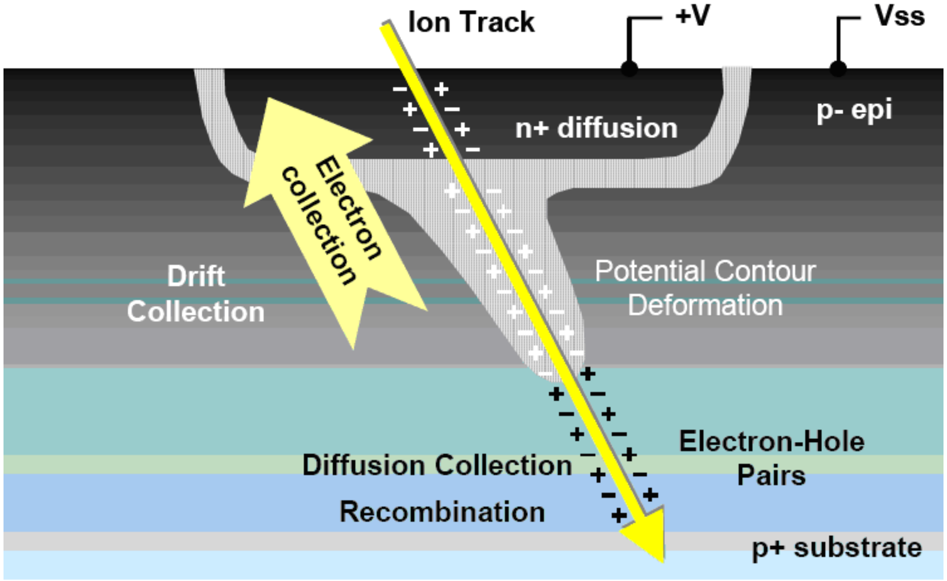


Figure 1-1 Illustration of an ion strike on a reverse-biased n+/p junction [Ba05]

Other effects like bipolar amplification may also further enhance the transient current, depending on the device structure and the exact position of the particle strike. For example, in a PMOS transistor, the parasitic source-body-drain (p-n-p) bipolar structure can significantly amplify the charge deposited in the well (or body) region [Mass-90], [Ferl-04]. The presence of additional transistors can also "share" some of the charge deposited by the ion-strike leading to the effective transient pulse-width being "quenched" or reduced [Ahlb-12] [Amus-06]. For a full description of all these different mechanisms the reader is referred to [Amus-06], Ahlb-12].

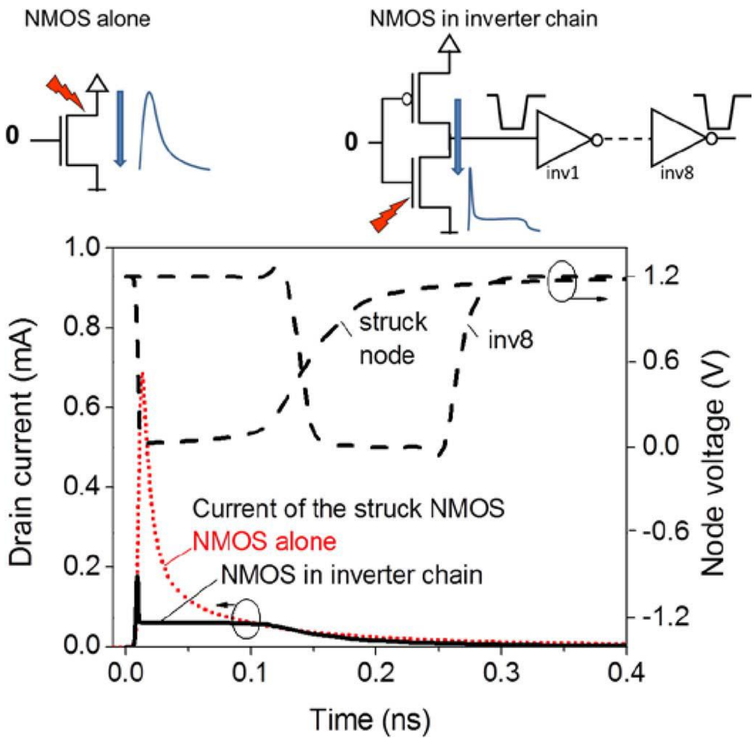


Figure 1-2 Simulation of off-state NMOS transistor, either as a stand-alone transistor, or embedded in an inverter chain [Ferl-06].

1.1 Single Event Effects in Memory Circuits

As far as the effects at the circuit level are concerned, due to the collection of charge the voltage at the node is perturbed. In the case of memory circuits, especially SRAMs, transistors are connected in a way that they form a feedback loop. Figure 3 depicts a 6-T SRAM structure that is used to store a logic value. The transistors that are part of the feedback structure are M1, M2, M3, M4. Consider a case where the node NQ is at logic 1 = VH. In other words the transistor M2 is ON and M1 is OFF. When a particle strikes the reverse biased junctions of M1, the nodal voltage at that reverse junction is perturbed in response to the charge collected. For example Figure 1-3 illustrates a strike on the reverse biased NMOS transistor. This results in a perturbation in the voltage at this node.

Due to this perturbation, the voltage at the struck node which is initially high begins to go low. This is illustrated in Figure 1-4. Since the struck NMOS transistor is OFF the restoring PMOS transistor is ON. The ON transistor attempts to restore the node to the original value. If the charge collected is enough to exceed the restoring strength of the PMOS transistor and the node stays low for long enough to exceed the feedback delay of the loop, then the incorrect value gets latched. If the restoring drive is however is able to recover the struck node and restore the nodal value to its original value then a voltage perturbation merely manifests itself as a transient. These two cases are shown in Figure 1-4. The minimum amount of charge required to flip the nodal vale of a feedback structure is called the critical charge (Q_{crit}). The mechanism explained above is a characteristic of feedback structures where the logic value stored by the memory cell gets inverted in response to a radiation particle strike. The critical charge thus depends on the output capacitance of the struck node, supply voltage, restoring drive and the feedback delay. Higher the capacitance, voltage, restoring lower and feedback delay, higher is the critical charge and lower is the probability of an upset. As technology scales, the voltage, nodal capacitances and feedback delay have been getting progressively smaller, leading to lower critical charge and increased sensitivity to soft errors. The general mechanism that induces errors in SRAMs is also true for flip-flops, latches and other storage elements that employ feedback structures.

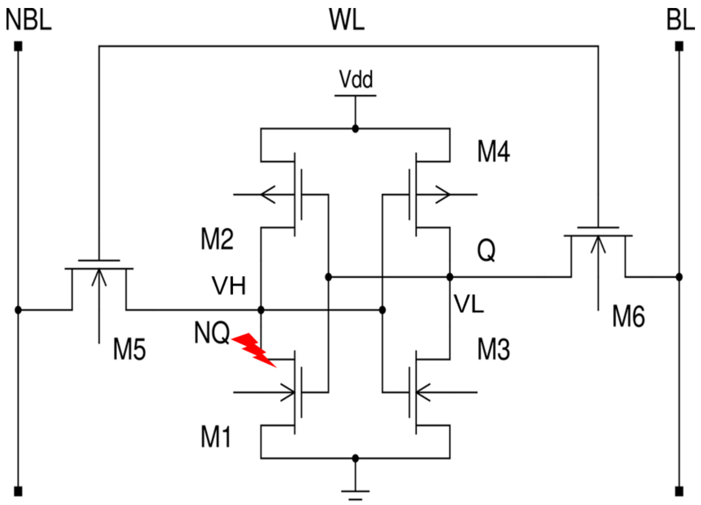


Figure 1-3 Illustration of an ion strike on a reverse-biased leading to an incorrect value being latched.

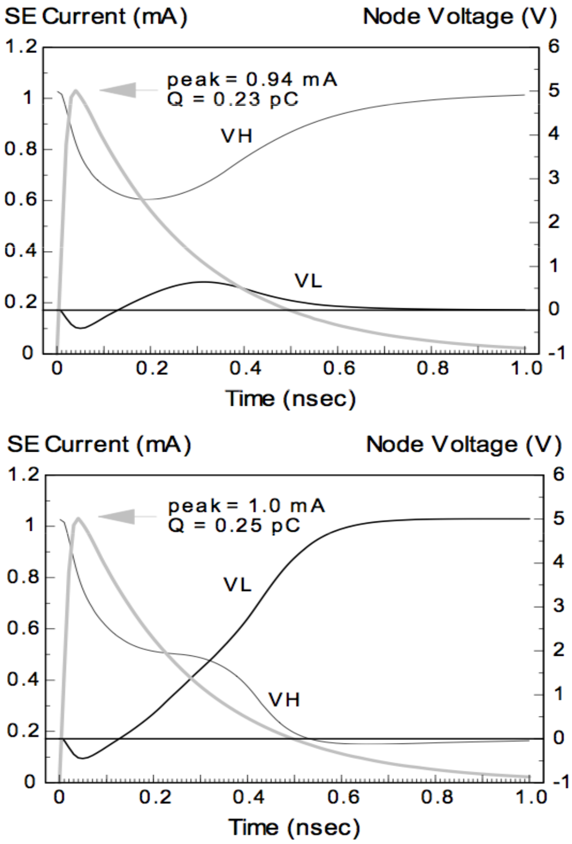


Figure 1-4 SE current and voltage perturbations on the storage nodes VH and VL. First case corresponds to charge collection less than Q_{crit}, leading to a transient at the struck node. Second case corresponds to charge collected > Q_{crit} leading to an SEU.

1.2 Single Event Effects in Combinational Logic Circuits

Feedback loops are the characteristics of memory structures. However combinatorial logic gates do not use feedback structures. When radiation particles strike semiconductor junctions and if enough charge is collected, single event transients could be generated as explained earlier [Dodd-03]. These transients must then propagate through the combinational logic and be latched by the receiving flip-flop to be register as a soft error. In order to be latched by the receiving flip-flop the transients must be 1) wide enough and have sufficient amplitude to propagate unattenuated through the logic chain 2) must not be logically screened or prevented by other gates from propagating through the logic chain and 3) must arrive during the "window of vulnerability" or setup-and-hold time window of the flip-flop. In this work such errors are termed as ***combinational logic soft errors or plainly logic errors***.

Each of the above conditions must be satisfied by SETs for them to be successfully latched. Hence factors that prevent or mask SETs from causing errors are called masking factors [Lide-94]. These factors are:

1. Temporal masking
2. Electrical Masking
3. Logical masking

1.2.1 Temporal Masking

SETs generated in the logic circuit can be masked or prevented from being latched by the storage cell, if the pulses do not reach the output node during the time that the storage element is ready to capture its input value [Kaul-93]. This window is referred to as the latching window or window of vulnerability or the setup-and-hold time window. An example of latch-window masking can be seen in Figure 1-5. The minimum amount of time that data must be stable before a clock transition is called setup time. The hold time is the minimum amount of time that data should be constant after a clock transition to ensure reliable function. The latching window or window of vulnerability is the time period between the setup time and the hold time. Violating either requirement creates meta-stability in latches. As Figure 1-5 shows, SET pulses that fully span the latching window will be latched. SET pulses that partially overlap the latching window may or may not be masked because of the meta-stability of the latch. SET pulses that are fully outside of the latching window will be masked. Thus the probability of latching must account for the SET pulse-width. The "wider" or larger the pulse-width, the greater is the probability of being latched. Similarly, the larger the feedback delay of the feedback element, the wider is the latching window, which makes it more difficult for the transient to be latched. The greater the charge deposition by ions, the larger are the transient pulse-widths [Dasg-07]. In space environments where heavy-ions are part of the radiation flux, these ions deposit much more charge than protons, alpha particles and neutrons, which are generally the primary cause of soft errors in the terrestrial environment.

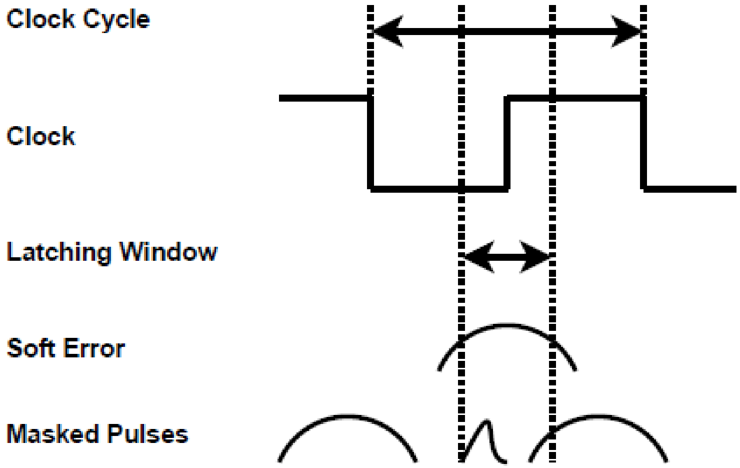


Figure 1-5 Example circuit showing electrical masking as a result of circuit delays caused by the switching delay of the transistors (Ra09).

The probability of a transient resulting in an error is given by the following piecewise Equation [Sh02];

$$TM = \begin{Bmatrix} 0 & if \ t_{SET} \leq w \\ \dfrac{t_{SET} - w}{T_{clk}} & if \ w < t_{SET} \leq T_{clk} \\ 1 & if \ t_{SET} > T_{clk} \end{Bmatrix} \qquad \textbf{1-1}$$

It is quite clear from this discussion that higher the frequency higher is the latching probability of transients. This is so because at higher frequencies there are more number of latching intervals which increases the likelihood of combinational logic errors. Thus as circuits operate faster, the effects of temporal masking diminish. In older technologies, the operating frequency was very low (Pentium II, for example was fabricated on 0.35 μm process and operated in the 233-300 MHz) as a result of which transients had a very low probability of being latched. Thus logic soft errors could be neglected in comparison to flip-flop errors, at that time.

1.2.2 Logical masking

SETs in combinational logic produce an observable error at an output only if there exists an available path for the SET to propagate. If no path exists, then the fault is considered to be logically masked. For example, a two-input OR gate logically masks a strike on an input node if the other input has a high logic value. Logical masking is illustrated in a combination of cells in Figure 1-6. Logical masking is a property of the circuit topology and the input conditions. Both these factors determine whether transients on any of the nodes propagate to the output. In general, gates that are closer to the output have a higher chance of propagating to the latch, because there are fewer gates that could mask the transients. In terms of logical masking, the effects of technology scaling are minimal. However, the logic depth to a certain degree does influence the latching probability of transients. Modern pipelined systems use no more than 8-12 stages of logic between pipeline stages. The trend over the years has been towards less logic between pipeline stages [Hris-02]. With deeper pipelines (less combinational logic between flip-flop stages) the impact of logical masking is thus expected to diminish as well [Sh-02].

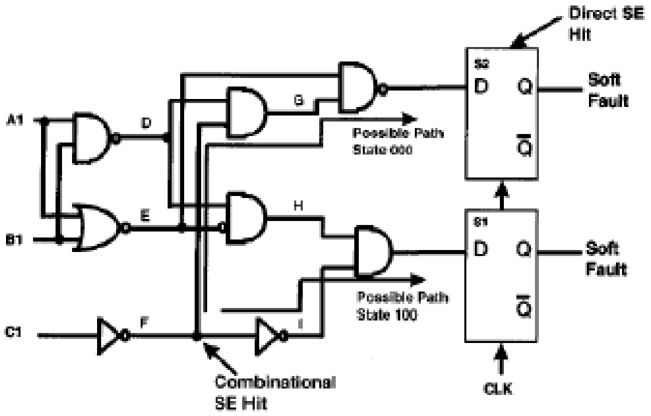


Figure 1-6 Example circuit showing possible paths for sequential soft fault creation. State 000 path is only possible with an input vector of (A1; B1; C1) = (0; 0; 0). State 100 path is only possible for an input vector of (1, 0, 0). Both combinational node hits and direct latch hits can contribute to SE soft faults. (Mass-00).

1.2.3 Electrical masking

SETs that are generated at circuit nodes can be attenuated prior to reaching an output flip-flop. This occurrence is referred to as electrical masking. Electrical masking occurs because the generated single event transient must propagate through a network of logic gates (equivalently speaking, a network of capacitors and resistors) to reach the latching element. This network of RC elements tends to diminish the amplitude and reduce the pulse-widths of SETs. The probability of a signal being electrically masked depends on the characteristics of the generated pulse, the electrical noise margin of the output node and the capacitive loading at the struck node. The generation of the SET depends on the drain area struck. The larger the area, the greater is the probability of an ion-strike leading to a transient. The electric field and diffusion mechanisms also influence the SET pulse-width. The higher the field and slower the charge collection, the higher will be the SET pulse-width. Conversely, a higher restoring drive and higher capacitive loading at the struck node reduces the SET pulse-width. As the transient propagates to the output, the network of metal lines and transistors act as a network of R-C elements that may attenuate the pulse. If the SET pulse-width diminishes as it propagates, the probability of its meeting the latching window requirement decreases. This can be seen in Figure 1-7 where the shape of the pulse is modified as it propagates through the logic chain. Results from [Mass-08] suggest that SETs propagate unattenuated through combinational logic if the SET pulse-width exceeds the delay of the logic gates through which they propagate. With scaling, gate delays become progressively smaller, thus reducing the effects of masking.

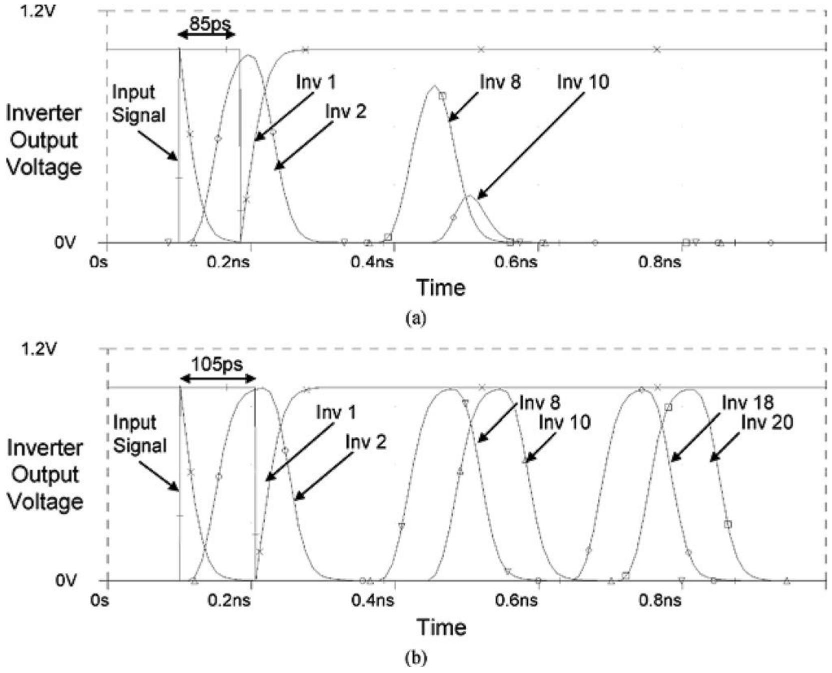


Figure 1-7 Example of transient pulse-widths that are attenuated as they propagate through logic gates (a) and transients that do not suffer attenuation as they propagate through the circuit (b) [Mass-08].

1.3 Comparing Combinational Logic SER and Latch SER

Historically, soft errors in memory arrays have been extensively studied especially because DRAM, SRAM arrays occupy huge area on chip. Most incorrect operations are likely to arise because of reading incorrect values from memory. L1, L2 and L3 cache together make up about 60-70 % of total area on processors. The rest of the area is devoted to the core, graphics processing and miscellaneous I/O, control and processing circuitry. These blocks mainly consist of logic and registers. Incorrect operation resulting from soft errors in these blocks can lead to erroneous operation as well. One of the earliest papers to characterize the problem of SETs and their possible trends with scaling was by [Wall-62]. This study predicted that this could be the dominant type of error in combinational and peripheral sections of SEU-hardened circuits. In other related work, the authors studied propagation and latching of these SETs more detail. However, the SETs were found to be so small and insignificant in number that the authors concluded that the SET problem was not yet as severe as memories or latches [Dieh-85] [Frie-85] [May-84]. However, in the 1990's and 2000's operating speeds of digital circuits were steadily on the rise and billion transistor gate count was nearing. With increasing frequency of operation, logic soft errors were predicted to equal or even exceed flip-flop errors and become the dominant mechanism for errors in data path and control circuits [Bu-97]. Other more detailed SPICE simulation studies of processors have suggested the same [Sh-02]. As Figure 1-8 and Figure 1-9 indicate, some previous results had predicted a substantial increase in logic soft errors.

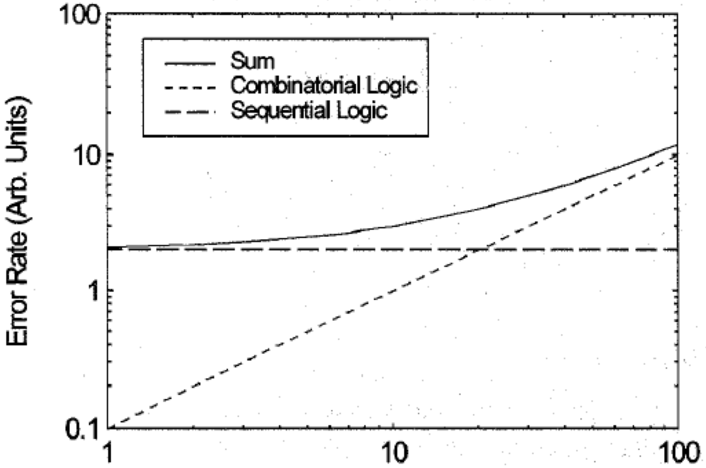


Figure 1-8 Predictions that indicate logic soft errors could be problem with increasing frequency ([Bu97])

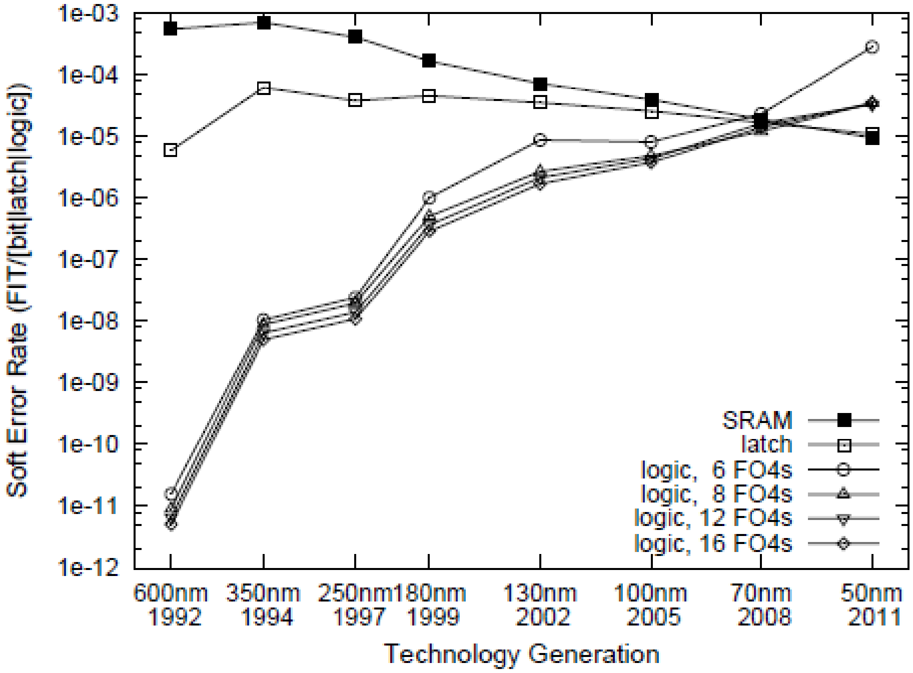


Figure 1-9 Predictions that indicate logic soft errors could be problem with technology scaling ([Sh02])

In the recent past, however, some experimental results have shed light on the logic SER problem. Depending on the design and technology SETs from simple structures like inverter chains have been shown to be the dominant source of errors. Authors in [Seif-12] state that under static testing for a 6-inverter skewed NMOS/PMOS chain the total number of errors recorded at 1.4 GHz is 430 and that with clock turned off is 110. The primary reason for this was the combinational logic soft error contribution of datapath inverters. On the other hand, the contribution of logic soft errors from 10 inverters is equal to that of latches at 3 GHz [Seif-12]. Results from 40 nm bulk technologies suggests that alpha particle combinational logic soft errors are steadily increasing and could exceed flip-flop SER for future technologies [Maha-11]. The key message from these papers is that the trends with technology scaling and increasing clock rates is not yet clear, let alone the comparison between combinational logic SER and latch or memory SER. Another reason that makes this a complex problem to analyze is the fact that most test structures to study the effects of logic SER are simple in nature (inverter chains, simple logic circuits). It is extremely difficult to extrapolate to system wide trends or SER numbers using simple experimental results. Secondly, there is no defined metric to compare combinational logic SER unlike memory or latch SER. For example while testing memories or latches, a large number of similar structures can be exposed to radiation and the number of errors can be counted. The cross-section or error rate can then be easily calculated. On the contrary, there is no consensus on which combinational logic circuit must be tested to compare the combinational logic soft error rate with the latch error rate. The circuit itself can be constructed in different ways, synthesized in different ways with different types of gates and varying drive strengths etc.

In this work we attempt to simplify this problem and report the combinational logic soft error trends for the same circuits (inverter chains and comparators) across three different technology nodes. The factors that change with technology scaling are identified and separated from those that don’t with technology scaling. The understanding of these factors can be incorporated in models to estimate chip-level trends under a variety of different operating conditions and circuit design styles. Chapter III discusses the different technology scaling trends for combinational logic soft errors.

1.4 Mitigation of Combinational Logic Soft Errors

Combinational logic soft error mitigation is a challenging task because of the difficulty in estimating the most sensitive gates/nodes in the circuit, individual pulse-widths of gates/nodes, the likelihood of transient propagation and latching etc. Historically, many papers, especially those emerging from university, government and industrial groups involved in radiation-hardened ICs for space repeatedly suggested that combinational logic upsets were not the biggest threat as far single-event effects were concerned [Guen-81], [Dieh-84], [Hass-89]. Here is an excerpt from [Guen-81] that perhaps is the first instance of non-SRAM and DRAM memory related upsets being recorded on three different processors designed by AMD, Intel and Motorola. "The upset rate is expected to be dependent upon the details of the programming of the microcomputer. It is nonetheless expected at some degree of logic integration that significant numbers of single even upsets will be observed. The observation of upsets in microprocessors is important because of the operational difficulty such upsets will cause. Because upsets in memories occur in a predictable fashion on repetitive structures, the upsets can be detected and corrected with a relatively low overhead in time, cost and complexity by the use of error correction circuitry. However such techniques are not so readily available in logic. Also the operation of a logic device, particularly of the complexity of a microprocessor, in not as predictable as a memory, so that occasional checks for errors are not likely to remove randomly occurring errors before they become inextricably intertwined in the calculation or control of the calculation."

As a result, the techniques employed to mitigate or eliminate combinational logic upsets in the circuits used in these times were limited to one or a combination of the following 1) Triple Modular Redundancy (TMR): In this approach, three identical copies of a circuit are designed and the output is computed using a majority voter that compares the outputs of the three circuits. If single event transients or SEUs produce errors in one of the circuits but the other two are unaffected, the majority voter ensures that the final output is correct [Lyon-62]. Such an approach can successfully correct both SEUs and combinational logic errors [Schm-90]. However the area and power cost is 200% and can be prohibitive for power-starved space ICs. 2) Pulse-suppression using R-C elements : The SET pulse-width depends on the loading capacitance as well as the resistance at the output of a gate. Large values of resistance and capacitance act as RC filters to reduce the SET pulse-width. This technique drew from efforts to mitigate SEUs in latches and SRAMs by introducing RC delay elements in the feedback structure of these circuits [Sava-86]. In the 1980s and 1990s polysilicide resistances and trench capacitors and dummy gates in parallel were commonly used as R and C elements respectively. Using resistances for pull-up was a feature of resistive load logic families. The RAD6000 processor used for the Mars Orbiter employed this feature. 3) Increasing the size of gates: This was another popular method to reduce the SEE sensitivity of combinational logic gates. Increasing the size was a straightforward solution to increase capacitance as also the restoring drive and thus reduce combinational logic SER. The size of almost all the gates were uniformly upsized in the radiation hardened 16/32 bit National Semiconductor processor as explained in [Hass-89]. The third technique Many of these techniques have now been replaced by more sophisticated approaches as the performance and power penalty due to triplicating circuits and deliberately slowing down operation cannot be tolerated. The more recent approaches for combinational logic soft error mitigation can be grouped into four broad categories 1) Reduction in pulse-width at source 2) Selective hardening of critical nodes in the circuit 3) Temporal filtering at latch-level and 4) Logic protection using arithmetic error detection and correction.

1.4.1 Pulse-width reduction at source

The most common approaches to reducing the soft error rate of combinational logic circuits is to reduce the pulse-width for all the transistors in the circuit. This can be done in several ways. The most direct way to achieve pule-width reduction is to limit charge collection processes. Adopting technologies like silicon-on-insulator (SOI) have shown the benefit in reducing the SET pulse-width. The use of certain implants, especially highly-doped layers that limit charge collection have also been used to reduce SET pulse-widths. Epitaxial active layers, triple-well fabrication and other process features that confine or limit charge collection, as shown in Figure 1-10, have been shown to be effective in mitigating SETs.

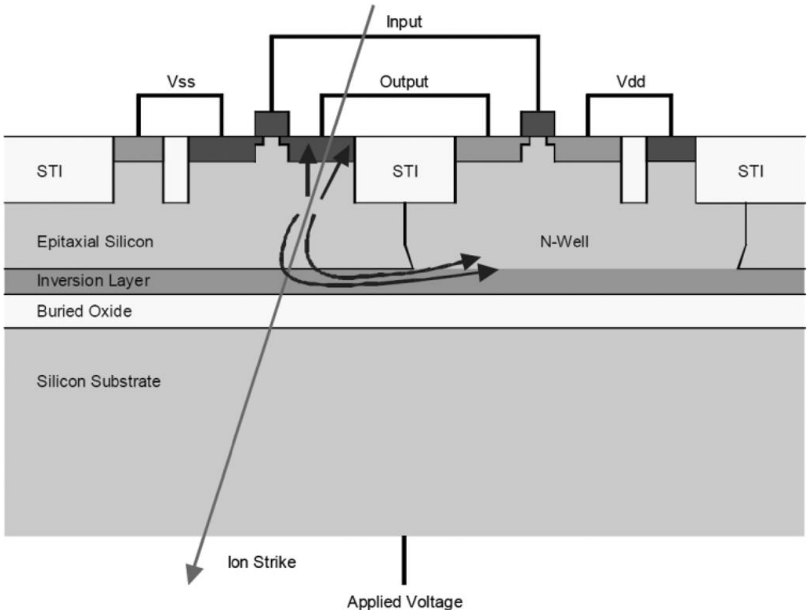


Figure 1-10 Charge limiting during a single event strike using process modification [Mavi-02].

Another simple and often employed approach is to increase the sizes of transistors that reduce the SET pulse-widths. Large devices have large ON currents and large capacitances that help dissipate the charge and thus reduce the pulse-widths. This however has the inevitable effect of increasing the power consumption.

1.4.2 Selective hardening of critical nodes in the circuit

Large combinational logic circuits generally consist of several nodes. SETs generated due to single event strikes on sensitive regions at the circuit nodes could propagate to the output and lead to a soft errors. However, in most circuits SETs from certain nodes have a greater likelihood of propagating to the outputs compared to other nodes. This is because electrical, logical and temporal masking factors tend to mask SETs from certain nodes more than from certain For example, nodes closer the output are less likely to be logically masked than those further away from the output, due to the presence of fewer gates in the logic path between those nodes and the output. Thus, selective hardening approaches rely on the fact that not all nodes in the circuit need to be hardened. Only those nodes or gates from which SETs have a very high likelihood of propagating to the output and being latched during the window of vulnerability need to selectively hardened. Several techniques exist in literature to identify the most sensitive nodes in a logic circuit. Some of them are discussed in [Zhou-06], [Maha-11], [Lim-12], [Srin-05], [Karn-02], [Poli-08], [Pagl-12], [Poli-08] and therein. The idea behind all these approaches is that it is wasteful in terms of area, power and speed to harden all the nodes in a combinational logic circuit. Rather, maximum fault coverage can be obtained by hardening only a few of those. Thus soft error reliability can be traded against area, power and speed concerns.

1.4.3 SET filtering using Delay elements

The most commonly used technique to mitigate combinational logic soft errors is by filtering transients through the use of tuneable or fixed-delay filters. This technique relies on the fact that signals that are less than a certain critical pulse-width can be filtered using the topology described in [Mavi-02]. Any SET that has a pulse width shorter than the tuneable delay is effectively filtered or masked from further propagation. [Mavi-02]. Similar techniques were employed by [Bala-08], [Nico-10] to filter SETs efficiently. Some of these SET filters or guard gates have also been incorporated in latch designs [Nase-06]. Again, the key factor with all these mitigation approaches is that the introduction of SET filter elements has performance as well as power overheads. With technology scaling such a solution has two important drawbacks. Firstly, with scaling the delays of individual gates is decreases which means more such gates are required to filter transients. Secondly, the trend in logic design is to use short logic paths per pipeline stage. This means that more such filters would be necessary at each stage to filter transients this would increase the dynamic power consumption significantly and also lead to performance penalties.

1.4.4 Logic Protection using Arithmetic Error Detection and Correction

Apart from circuit-level approaches, architectural and system level approaches can also be adopted to mitigate combinational logic soft errors. Error detection and protection and correction of arithmetic logic circuits has been known for a very long time [Wats-66]. These techniques, however are mainly applicable for arithmetic circuits and circuits that can be protected using parity-based protection schemes. Arbitrary circuits such as those used in datapaths are less suitable for parity based logic protection. Some of the earliest approaches for space-borne electronics were adopted in [Gais-97] where parity-based protection was used to mitigate the effects of combinational logic soft errors in the Arithmetic Logic Unit (ALU) and certain other structures in the first few stages of the pipeline. More recently arithmetic protection based on residue codes and parity checking is beginning to seem attractive because the large amounts of arithmetic processing circuits especially in Graphics Processing Units (GPUs) as well as the technology agnostic nature of these approaches [Sika-13], [Naza-11].

1.5 Summary

In this chapter, the key factors that influence combinational logic soft errors are discussed. Experimental approaches to characterize logic SER and results are presented. Comparison between the latch and logic SER suggests that this is an important problem for future technologies and has several unanswered questions. Lastly, some popular soft error mitigation approaches are discussed and their associated penalties are discussed. In the next few chapters, technology scaling trends of combinational logic SER are presented. Following this novel power-aware approaches to mitigate combinational logic SER are presented at various levels of abstraction.

Chapter II. Power Minimization Techniques

Low-power design has emerged as the all-important theme behind electronic design today. Low-power as a design philosophy is now part of designs beginning from the transistor and circuit level right up to the system, architecture and software level. This all-pervasiveness stems from the fact that minimizing power consumption increases the operational time of portable devices like laptops, cell phones and tablets. With an increasingly connected world, biosensors for health monitoring and other wearable battery operated systems will increase in popularity. All these devices must operate under ultra-low-power consumption constraints. Similarly, systems that must maintain very high-reliability and data integrity, like data centers and servers, also draw huge amount of power due to their high-speed computation intensive design. Here energy efficiency in different forms beginning from circuit-level power efficiency to thermal cooling is necessary to keep operational costs low as well as increase the reliability. Yet, another class of systems that must maintain high-reliability and integrity over long periods without regular repair and upkeep are those that are deployed in space. Often such space-systems rely on different forms of energy generation such as on-board thermal reactors and solar panels to provide power. As such sources of power are limited, every sub-system must adhere to strict power budgets. Thus systems across the semiconductor application space must now confirm to low-power design approaches as a requirement rather than as an afterthought.

In the field of very large scale integrated circuit (VLSI) design, optimizing designs for the lowest area, best performance (fast operation) and lower cost was of paramount importance. Power considerations were often secondary. However, today, due to the remarkable growth in the field of personal computing devices and wireless communication systems, the emphasis is clearly on high speed computation and complex functionality with low power consumption. The motivations for reducing power consumption differ from application to application. In the class of micro-powered battery operated portable applications such as cell phones, the goal is to keep the battery lifetime and weight reasonable and packaging cost low. For high performance portable computers such as laptops the goal is to reduce the power dissipation of the electronics portion of the system to a point which is about half of the total power dissipation. Finally for the high performance non battery operated system such as workstations the overall goal of power minimization is to reduce the system cost while ensuring long term device reliability.

Fortunately, as each application requires different levels of power regulation and power management, different approaches are available to designers to reduce power consumption to acceptable levels. Each of these techniques is targeted towards reducing power consumption based on the source of power consumption. In this chapter, the different sources of power consumption for transistors, circuits as well as architectures and systems are discussed. Later chapters in this thesis however focus on techniques that emphasize minimizing circuit-level power consumption. Wherever relevant, the impact of certain variables or factors that affect power, on the SER is also discussed. This way the reader can grasp the effects on both the variables concurrently. Specifically, this thesis establishes the relationship between power and SER in several ways.

2.1 Sources of Power Dissipation

There are three major sources of power dissipation in digital CMOS circuits, which are summarized in the following Equation:

$$P_{total} = P_{dynamic} + P_{short-circuit} + P_{leakage} \qquad \textbf{2-1}$$

The first term of represents the switching component of power. This can be expressed as

$$P_{dynamic} = \alpha NCV^{2} f \qquad \textbf{2-2}$$

where C, is the loading capacitance, f is the clock frequency, and α is the switching probability of N nodes that switch at a rate of f, where f is usually the clock frequency. In most cases, the voltage swing is the same as the supply voltage V; however, in some logic circuits, such as in single-gate pass-transistor implementations, the voltage swing on some internal nodes may be slightly less [Chan-92].

The second term is due to the short circuit current, which arises when both the NMOS and PMOS transistors are simultaneously active, conducting current directly from supply to ground [Chan-94]. The third term is the leakage power consumption which results due to substrate injection, sub-threshold leakage and gate leakage. These are primarily non-ideal currents that result due to reverse biased diode leakage, source-drain leakage and tunneling across the gate dielectric respectively. In circuits that operate at full speed, the dominant term is usually the switching component, and low-power design thus becomes the task of individually minimizing the number of switching nodes (N), their switching probabilities (α), the output capacitance as well as the voltage and frequency. Voltage and frequency are closely related together through the delay. A higher voltage corresponds to a lower delay which allows a higher frequency. Thus operating at a higher frequency which is needed for faster computations, mandates higher voltages. Thus the designer is in a trap that does not allow reduction in voltage without a forcible reduction in frequency. As a result, lowering the voltage is adopted when the system is not required to run at full speed or can be “idled”. Such approaches to change the voltage and frequency adaptively are performed at the system level and are collectively called Dynamic Voltage and Frequency Scaling (DVFS). On the other hand, factors like the number of switching nodes, switching probabilities of components and gates are design dependent and can be reduced at the gate level, circuit level or higher level of abstraction based on intelligent placement and interconnection of sub-circuits and sub-systems. The capacitance is dependent on individual choice of gates used to synthesize the circuit but more fundamentally the capacitance of the transistor junctions and gate itself. Technology scaling generally results in lowering of the capacitance through the use of smaller transistors. Thus the switching component of power can be minimized at various levels.

The power-delay is another useful metric to keep in mind. This product can be interpreted as the amount of energy expended in each switching event (or transition) and is thus particularly useful in comparing the power dissipation of various circuit styles. If it is assumed that only the switching component of the power dissipation is important, then it is given by Equation 3.

$$Energy/transition = \frac{P_{total}}{f} = \alpha NCV^{2} \qquad \textbf{2-3}$$

Energy is generally defined per task. For example a high-performance fast processor may compute the result with high power consumption in a short time, resulting in significant energy consumption. Conversely, a slower processor may take longer consume less power but also consume less energy. Thus, either power consumption or energy consumption can be a metric of interest when designing systems. Often power is a more important metric for high-speed high-performance applications like desktop processors or servers. Energy on the other hand maybe more critical for portable systems like phones, tablets where battery longevity is relatively more important. In this work we emphasize power minimization techniques for high-performance and high-speed systems, although the same principles can be extended to energy starved systems equally well.

2.1.1 Component-Level Power Dissipation Trends and Contributions

Different components or sub-systems parts of the integrated circuit collectively contribute to the total power dissipation of the chip. The percentage contribution of each of these components varies according to the mode of operation of the IC, the application that is being run, the communication speeds, the performance and reliability constraints etc. In general however, technology scaling has led to an increase in both dynamic power consumption as well as static power consumption. The following trends indicate the power dissipation in different on-chip components in different regions of operation. As shown in Figure 2-1, a high frequency facilitated by full-rail supply voltage is necessary for high-speed high performance operation. Thus a higher frequency not only means higher performance but also higher power consumption. As illustrated in previous chapters, the combinational logic SER is also directly proportional to the frequency of operation. Hence, operating at higher frequencies leads to the dual problems of high power consumption and higher SER. On the other hand, whenever the full clock speed is not quite needed by the system, the supply voltage can be lowered to operate at lower frequency, resulting in power savings. Tremendous saving in power can be achieved by lowering the supply voltage and thus the frequency. In such operating regimes where the chip operates at a modest frequency, other sources of power consumption become comparable to the logic switching power dissipation. As shown in Figure 2-1, when the supply is lowered to near threshold voltage regions (NTV) or half-rail, the leakage power dissipation from logic and memory forms a substantial proportion of the total power dissipation. At even lower voltages, the total SRAM memory contributes tremendously to the total power dissipation of the IC. Several circuits especially for biomedical applications operate at very slow frequencies : in the kHz to the MHz range where the primary source of power dissipation are the memory blocks. Thus different regions of operations that are needed for different applications result in different on-chip components dominating the power consumption. It is therefore important for designers to understand the impact of operating regimes, design specifications and application and user demands on the power consumption trends of different components. Targeted power minimization techniques can then be employed to result in savings in the power consumption.

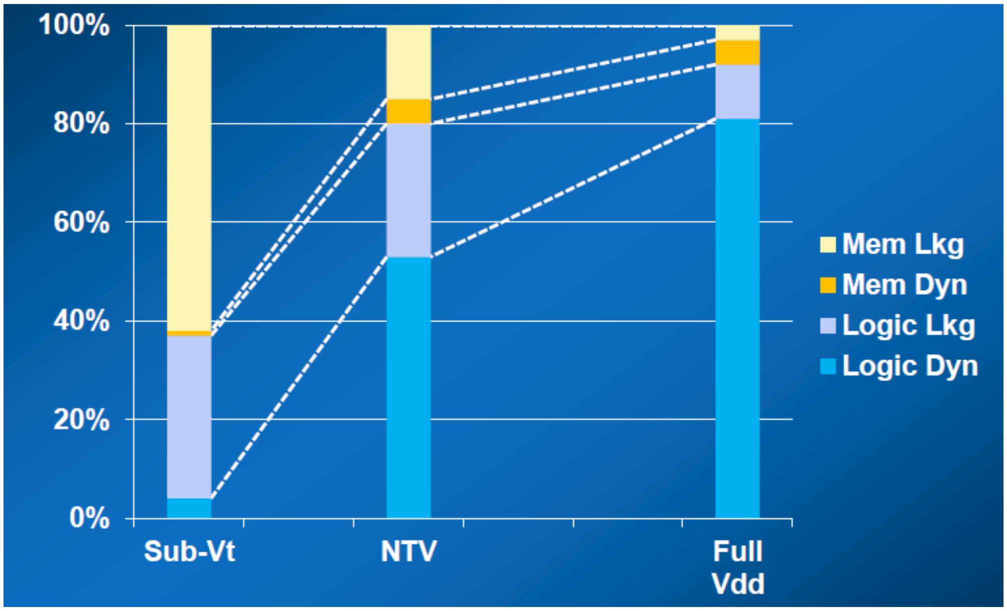


Figure 2-1 Contribution of different components of power in different modes of operation. As the voltage is increased from sub-Vt to Near Threshold voltage (NTV) to Full Vdd the logic switching component dominates the power consumption [Bork-12].

2.2 Power Minimization Techniques

Several power-minimization techniques are commonly used to reduce power consumption of IC operating under various conditions. These techniques apply at the transistor-level, circuit-level, architectural-level as well as the system and software level. The bulk of work in this dissertation focuses on circuit level and architectural approaches to mitigate the effects of soft errors and reduce power consumption. These involve partitioning and redesigning the circuit effectively to reduce the number of switching nodes and thus achieve lower power. Some of the commonly used techniques to reduce power consumption and their impact at the system-level are discussed. In general, as a rule of thumb it is reasonable to say that system-level and software-guided approaches tend to maximize the overall power reduction when limited information is available about the applications and usage patterns of the system. For example, in the case of cell phones and portable mobile devices, when the phone and most of its applications are not under use, system-level approaches to "shut down" and "power-down" the device can save a lot of power. Indeed these are decisions that are based on the usage pattern and if several applications are in continuous use then such techniques would have limited impact. On the other hand, when certain specific portions of the IC dominate the power consumption, taking recourse to circuit and device level approaches is much more beneficial. For example if the logic switching dominates the total power consumption of the processor being used in the cell phone, power minimization techniques like redesigning the circuit to save power through Boolean manipulation, selective use of V_t implants in certain gates in non-critical paths in the circuit to lower dynamic and leakage power etc. can be very useful. Again, if the usage profile of the cell phone is such that several applications lead to logic switching power dominating then circuit level techniques would be especially useful. Network processors, server-class processors, ASICs used in digital set-top boxes, gaming consoles that hardly have any downtime can benefit hugely from circuit-level and device-level power minimization strategies. In the following sections different approaches to minimizing power consumption at various levels of abstraction are discussed along-with their potential SER impact, wherever possible.

2.2.1 Circuit Level Power Minimization Techniques

In this section some circuit level minimization techniques have been discussed. The key optimizations are at the transistor level through sizing as well as gate reordering and logical implementation.

2.2.1.1 Complex Gate Design

Circuit design involves the choice of lot of gates of different drive strengths, area, delay and power. Consider a function f = (a+b)·c. Such a function can be implemented in two different ways as shown in Figure 2-2. Both the implementations shown below can have significant impact on the power consumption. For example, (assuming C = 1) in the first implementation (a), the output of the OR gate switches from 1-0 only when the inputs transition from A or B = 1 to A and B = 0. In the second implementation (b), the outputs of the AND gates switch whenever transitions occur on A and B if C = 1 and not otherwise.

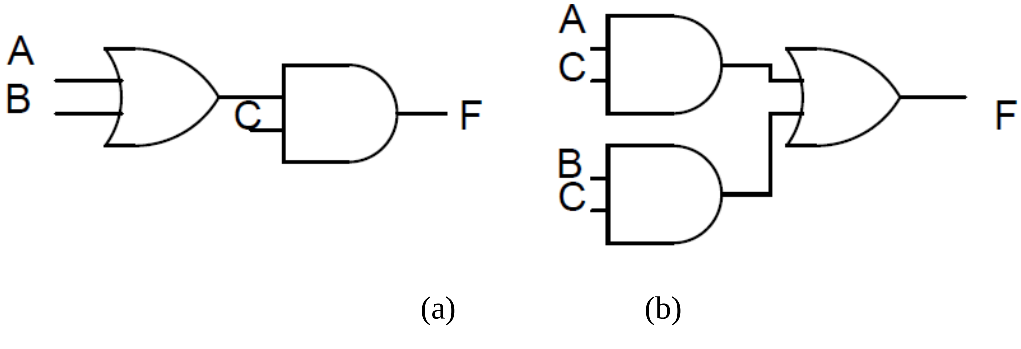


Figure 2-2 Different ways of synthesizing the same Boolean function. The synthesized version in (a) results in lower power consumption compared to the one on the left (b). This is mainly because the logic path delays are balanced and glitching power is minimized.

Hence the transition probabilities as well as the number of switching nodes can differ. Thus differences in Boolean synthesis can affect the power consumption inspite of the implemented function being the same. It is also very well known that signals that arrive late should be placed closer to the output to minimize gate delay. In [Pras-94] and [Tan-94] methods to optimize the power and/or delay of logic-gates based on transistor reordering are given. Modest power and delay improvements can be obtained by judiciously ordering the gates within a design.

Effects on SER : The choice of gates influences the sensitive area and the SET pulse-widths at different nodes in circuit as well as the logical masking factor. Thus different synthesis styles can have an important impact on the design of the circuit. Some of these aspects are dealt with in [Limb-13]. The author concludes that implementations that optimize the speed of designs through synthesis generally result in lower SER.

2.2.1.2 Transistor Sizing

Transistor sizes are a tool available to designers to control the delay of individual gates. However sizing the transistor not only affects the circuit delay but also area and power dissipation. If the transistor sizes are increased, gate delay decreases but both area and power dissipation increase. Additionally, delay of large fan-in gates increases because of increased load capacitance. For a given delay constraint, choosing transistor sizes such that area and power are minimized is a computationally complex problem. Typically, the slack at each gate in the circuit is calculated, (slack = how much gate can be slowed without affecting the critical delay of the circuit). Sub-circuits with slacks greater than zero are recorded and their sizes are reduced until their slack becomes zero or the transistors are all minimum size. Variants of the above approach are presented in [Tan-94] and [Baha-94].

Effects on SER: Generally, increasing transistor sizes decreases the soft-error sensitivity of gates. This is because the increased sizes result in increased nodal capacitances and increased restoring current drive leading to smaller transients. However increasing the transistor area also means a higher probability of striking the sensitive regions of the transistor leading to more transients. Thus what matters is the ratio of I_{ON}/W, where I_{ON} is the restoring current drive and W is the width of the struck device. If increasing transistor sizes increases this ratio, there is an obvious benefit. Several techniques in the past have attempted to selectively increase transistor sizes to reduce combinational logic SER due to transients [Zhou-06] [Maha-13] [Nieu-06]. However as explained earlier, although the delay may decrease in certain cases, the area and power penalties are unavoidable.

2.2.1.3 Logic Level

The logic level optimizations that can reduce switching activity power of combinational and sequential circuits is surveyed in this section. Logic optimization can be applied to combinational and sequential circuits.

2.2.1.3.1 Combinational

Logic optimization must be decomposed into two separate stages: technology-independent optimization (traditionally called “logic optimization or Boolean minimization”) and technology-dependent optimization (in the physical design domain called synthesis where functions are implemented using combinations of gates). In the first stage, logic equations are manipulated to reduce area, delay or power dissipation. In the second stage, the functions are mapped to particular technology libraries with the aid of technology mapping algorithms that optimize for area, delay or power or a combination of the three. [Deva-94] comprehensively analyzes logic and technology optimization issues. Most of the low-power techniques reviewed below have been drawn heavily from [Deva-94].

A. Don't-care Optimization

Any gate in a combinational circuit has an associated *controllability* and *observability* don't-care set. The controllability don't-care set corresponds to the input combinations that never occur at the gate inputs. In other words, for an AND gate when the output is 1, the controllability don't care set includes A, B = {00, 01, 01}. The observability don't-care set corresponds to *collections of input combinations* that produce the same values at the circuit outputs. In other words, all possible input combinations that produce an output 1 in function are in the observability don't care set of that function. These are extensively explained in [Savo-91]. Gate-level power dissipation is dependent on the probability of the gate evaluating to a 1 or a 0. This probability can be changed by utilizing the don't-care sets. A method of don't-care optimization to reduce switching activity and therefore power dissipation was presented in [Shen-92]. Observability don't care sets on the other hand help in fault testing algorithms to identify stuck-at-faults because the circuit output can be made immune to certain inputs using observability don't care sets. This method is utilized in in [Iman-94] where the effect of don't-care optimization of a particular gate is considered to reduce power consumption.

B. Path Balancing

Spurious transitions or glitches contribute to 10% and 40% of the switching activity power in typical combinational logic circuits [Ghos-92]. To reduce spurious switching activity, the delays of paths that converge at each gate in the circuit should be roughly equal. This can be done by adding delay buffers. This removes the spurious transitions if done carefully but adds to the total capacitance. Thus the switching power overheads from buffer insertion must be less than the total power consumption of the original circuit. Glitch reduction with minimal delay overheads by path balancing is described in [Lemo-94].

C. Factorization

Factorization is a means of reducing the transistor count by minimizing the logical expressions. For example, a common factor in a sub-expression can be combined or collapsed or shared across multiple functions. For example if an ADD function is needed in different blocks of the circuit in one form or another it does not make much sense to implement a separate adder for each instance that the ADD function is needed. The same adder can merely be shared across designs or functions. Formally, kernels or sub-function reduction are commonly used to perform multilevel logic optimization for area [Roy-92].

D. Technology Mapping

Once optimized logic equations have been obtained, the task that remains is to map the equations onto a target library that contains optimized logic-gates in the chosen technology. A typical library will contain hundreds of logic gates with different transistor sizes and functions. Modern technology mapping methods use a graph covering formulation, originally presented in [Keut-87], to target area and delay cost functions. The graph covering formulation of [Keut-87] has been extended to the power cost function. Under the zero-delay model, the optimal mapping of a tree can be determined in polynomial time, by extending the algorithm of [Keut-87]. Various approaches to technology mapping that assume different delay models and target minimal power dissipation have been described [Tiwa-93] [Tsui-93].

2.2.1.3.2 Sequential

Sequential circuit optimization is briefly surveyed. Sequential logic optimization apply to two levels of abstraction; 1) at the State Transition Graph level and 2) at the logic-gate and flip-flop level.

A. Encoding

State encoding for minimal area is a well-researched problem [Asha-91]. These techniques can be extended to target a cost function, such as power. Intuitively, if a state has a large number of transitions to different states, then the different states should be given uni-distant codes, so as to minimize switching activity at the flip-flop outputs. However, the complexity of the combinational logic resulting from a state assignment should not be ignored. Methods to encode State Transition Graphs to produce two-level and multilevel implementations with minimal power are described in [Roy-92] and [Tsui-94]. A method to re-encode logic-level sequential circuits to minimize power dissipation is presented in [Hach-94].

Encoding or the use of tokens to reduce switching activity in datapath logic has also been explored. Others have proposed switching activity minimization on busses [Stan-94]. Here, an extra line is added to the bus which signifies if the value being transferred is the true value or needs to be bitwise complemented upon receipt. Depending on the value transferred in the previous cycle, a decision is made to either transfer the true current value or the complemented current value, so as to minimize the number of transitions on the bus lines. For example, if the previous value transferred was 0000, and the current value is 1011, then the value 0100 is transferred instead, and the line is asserted to signify that the value 0100 has to be complemented at the other end.

Other methods of bus coding are also proposed in [Stan-94]. Certain methods to replace the traditional two's complement arithmetic are also being investigated. A method of one hot residue coding to minimize switching activity of arithmetic logic is presented in [Chre-95].

B. Retiming

Retiming [Leis-83] is a well-known optimization method and is similar in concept to the path balancing problem discussed earlier. It repositions the flip-flops in a synchronous sequential circuit to minimize the required clock period. Polynomial-time algorithms $O(n^x)$ to minimize the delay and/or the power consumption have been developed. It has been observed that the switching activity at flip-flop outputs in a synchronous sequential circuit can be significantly less than the activity at the flip-flop inputs. This is because there may be many spurious transitions at the inputs to the flip-flops which are filtered out by the clock. A retiming method that exploits the above observation and targets the power dissipation of a sequential circuit is described in [Mont-93]. The idea is to clock the circuit in a way such that the spurious output transitions are minimized by careful delay manipulation of the clock line and the inputs.

C. Clock-Gating

Modern VLSI circuits consist of complex sub-systems like data-paths, memories, communication modules, digital signal processing units, often integrated on a single die. All these modules must communicate with each other and this happens frequently in synchronous fashion using clocks to provide timing synchronization. The internal clocks of the sub-systems can be independent but must be synchronized with the system clock to transfer data reliably across modules. Typically one or more of these sub-systems may be inactive or less active during certain time periods. For example, the communication module could be inactive during periods when there is no communication required. During such periods, the switching activity can be tremendously reduced by "turning off" or gating the clock to the register files, flip-flops and latches within the module. This is termed as coarse clock gating. A more refined approach can also be adopted. For example, register files are typically not accessed in each clock cycle. Similarly, in an arbitrary sequential circuit, the values of particular flip-flops need not be updated in every clock cycle. In such cases the clock to such register files and flip-flops can be gated on a cycle-by-cycle basis. This approach is called fine-grained gating. If relatively simple logic and usage pattern-oriented conditions that determine the inaction of particular registers can be determined, then power reduction can be obtained by gating the clocks of these registers [Chan-94]. When these conditions are satisfied, the switching activity within the registers is reduced to negligible levels. Clock-gating can be included in a general class of techniques that reduce the switching activity of gates or nodes (α in Equation 2).

2.2.2 Architectural Level Power Minimization Techniques

Finally, in this section architectural power-reduction approaches are evaluated. These techniques are technology dependent and the results from such techniques can vary depending upon the quality of algorithms used to synthesize circuits. Often the task to evaluate the power at the architectural level is a highly complex task and well defined benchmarks do not yet exist for power estimation.

A. Architecture Level Power Analysis

Estimating switching and leakage power consumption of a design is a first step towards incorporating power optimization techniques in a synthesis system. Without adequate and accurate analyses it is impossible to evaluate the various designs in the solution space explored during synthesis. Power analysis tools can be of great use to designers, by helping them explore the design space manually. A simple approach to develop power analysis tools is to translate high level descriptions into physical architectural descriptions for gates, circuits, and physical level; at such levels low-level power analysis can be carried out by physically switching inputs and characterizing nodal switching activity. This also involves Monte-Carlo campaigns to gain adequate accuracy. (For a survey of available tools at the gate level see [Najm-94].) This method is obviously infeasible if a large number of design alternatives have to be evaluated, which is the case in synthesis. Reasonable power models, however, can be built if the final lower level circuit style, module and gate library, etc., are fixed, or at the least, restricted in some way. The lower level analysis tools can then be used to create power models for the underlying architecture primitives, such as datapath execution units, control units, memory elements, and interconnect. However beyond simple modules, the solution space explodes and random or even stratified sampling of the input space may be inadequate. Instead what is needed are power estimates for each module. The power models are obtained by characterizing the estimated capacitance that would switch when the given module is activated. This approach is used in [Powe-90]. In [Land-93], known signal statistics are used to obtain models that are more accurate than those obtained from using random input streams. Activity factors for the modules can be obtained from functional simulation over typical input streams, or from statistical/analytical models that are built where possible. In [Sato-94], an alternative approach is adopted where average power costs are assigned to individual modules, in isolation from other modules. During simulation, the power costs of the modules involved in the given computation are added up. This method ignores the correlations between the activities of different modules. This speeds up simulation tremendously and is surprisingly accurate in measured usage conditions. Other specialized approaches for architecture-level power estimation have been developed. These tend to be less accurate than the above methods, but may be acceptable since they are intended to provide only rough predictions. A model for estimating the power consumption of CMOS chips using gate counts, memory size, logic styles, and layout styles is described in [Sven-94]. A power model to evaluate the power cost of cache options, and multiple function units is developed in [Bund-94]. Again, the different synthesis and power estimation techniques in use today utilize knowledge of input correlations, usage conditions and applications to tailor the power estimation procedure.

B. Power Optimizations in Behavioral Synthesis

Behavioral synthesis involves mapping a high-level specification of a problem into a register-transfer level design. Data flow graphs, tokens or control flow graphs are used to describe the high-level specification. Alliteratively, an algorithmic routine or set of instructions may be used. Converting such high-level descriptions or tasks to low-level gates and transistors can involve optimization across multiple hierarchies. Some recent work addresses these issues for optimizations for low-power that are possible at this level. The input high-level specification can be modified through specific transformations that potentially lead to power reduction.

The most important transformations for fixed throughput systems are those which reduce the number of control steps. Slower clocks or can then can then be used for the same throughput, enabling the use of lower supply voltages. The quadratic decrease in power consumption can compensate for the additional capacitance introduced due to transformations that increase concurrency. Transformations that reduce the amount of resources needed to implement a given graph can be extended to reduce the amount of capacitance that switches. A number of these transformations are used in an automated system as described in [Chan-95]. The transformations are guided by a power estimation method that is based on the parameters of the given data/control flow specification, such as the number of operations of each kind, number of edges, etc. [Mehr-94]. Specific transformations for digital signal processing (DSP) circuits are studied in [Chat-94].

After the initial specification (data/control flow graph) has been transformed, the individual operations have to be assigned control steps (scheduling) and execution units or modules (allocation and assignment). If a number of modules, with a range of power/delay costs, is available for implementing the given operation types, an appropriate choice of modules can lead to lower power costs for the same performance [Good-94]. In summary, the mapping tool has a library of functions available to it. Tasks can be decomposed into circuits using one or more of these functions from the library. The order in which these functional modules are ordered, placed, combined and even executed affects the overall switching capacitance. The allocation and assignment processes map operations in the control/data flow graph to functional units, variables to registers, and define the interconnection between them in terms of multiplexers and buses. The decisions made during these processes, including the extent of hardware sharing and the sequence of operations (variables) mapped to each functional unit (register), affect the total switched capacitance in the data path. Correlations also occur among these modules and can affect the power consumption estimates [Ragh-94], [Ragh-95]. The power consumed in memories can be a major part of the system power consumption. This problem is addressed in [Catt-94] in the context of multi-dimensional signal processing subsystems. It is noted that the memories impact power in two ways. Accessing memory must be done synchronously with the data path clock. When reading and writing data, large columns or words of data may switch, consuming active power. Additionally, off-chip access leads to greater power consumption due to larger capacitances and larger memory words. Control flow transformations, such as loop reordering are presented to try to minimize the memory component of the overall system power consumption. Several specific design examples illustrate some of the architectural and algorithmic tradeoffs and optimizations that can be used for low power designs.

2.2.3 System and Software Level Power Minimization Techniques

Embedded systems form a large part of the modern electronic ecosystem. These consist of hardware and a software component along with a lot of supporting firmware. Software is written in application specific languages to be compiled and burned onto dedicated microprocessor/application specific processor (ASP) or microcontrollers, while the hardware component consists of application specific circuits, processors and microcontrollers. Hardware-based power estimation and optimization approaches are not completely applicable here, since a major part of the functionality is in the form of instructions as opposed to gates.

This motivates the need to consider the power consumption in microprocessors from the point of view of software. This is a branch of power optimization possibilities that have been ignored until recently mainly because accurate power analysis tools existed only at the circuit or gate level. Large programs are difficult to analyze in terms of power they consume because of the number of variables involved, like clock speed, hardware implementation from software descriptions, the software code and its optimization itself etc. Instead it is wiser to analyze sequences of task or program executions. This is similar to the divide and conquer approach seen in earlier sections for power estimation [Mont-95].

Another approach is to merely monitor the current being drawn by the CPU during the execution of a program which can be physically measured. An inexpensive and practical technique in this regard has been developed [Tiwa-94] for analyzing the power cost of programs for a given CPU. It has been successfully incorporated to develop instruction-level power models for commercial CPUs. The rudimentary measurement technique can also be adapted to use architecture-level power simulators, described in the earlier sections. The goal of these studies is to describe, develop and evaluate programs in terms of the associating power costs with executing them. It is true that power consumption is a physical real-world quantity, but actions or instructions can be translated into power consumption equivalents. For example doing things in parallel at a slower speed may lower the power consumption. Such decisions can be encoded into programs to choose between multi-core (parallel and low power) execution versus single-core (fast, high power) execution. Given the ability to characterize programs in terms of their power/energy costs, it is possible to search the design space in software power optimization. The choice of the algorithm used can also impact the power cost because of the runtime complexity of a program. This issue is explored in [Ong-94]. Automated tools for synthesizing the optimum algorithm for *power*, however, are not available, and this is a very difficult problem. If power costs of individual instructions are available, an appropriate choice of instructions in the generated code can lead to a reduction in the power cost. This aspect has been studied in the context of specific CPUs [Ong-94].

A general statement seems to be true for program execution "A faster code almost always implies lower energy code". Such observations can be used to tailor power-optimization driven algorithms and code development. In addition, scheduling techniques to reduce the estimated switching in the control path of the CPU have also been proposed [Su-94]. Experiments suggest that this may not be an important issue for large general purpose CPUs but may be applicable to server-class processors where jobs and tasks must be scheduled appropriately to maximize performance, improve efficiency and lower power consumption [Tiwa-94]. Scheduling of instructions can also affect DSP processor [Lee-95].

2.3 Summary

Power optimization techniques spanning the design and development space from device to circuits to architectures and software has been presented. One or more of these techniques can be applicable to different domains of electronic design today. Lowering power dissipation at all abstraction levels is a focus of intense academic and industrial research. Briefly, for each technique the associated impact on the SER has been mentioned. Some of the impacts cannot be quantified in this work and is indeed beyond the scope of this work, but qualitative understanding of the effects of power minimization can guide SER-mitigation and SER-aware approaches. However the discussion from this chapter as well as the previous chapter that highlighted SER mitigation approaches is that in most cases, approaches to reduce the power can have SER overheads and conversely, approaches to reduce the SER can have power overheads. Worse still, the two are rarely ever co-optimized or studied in conjunction. This means that applying SER mitigation and power minimization independently can eliminate the benefit obtained from these approaches due to power overhead from SER mitigation and SER overhead from power minimization! Therefore there is much scope for research that seeks to co-optimize the variables of soft error reliability and power minimization and has certainly not been explored in depth so far. The next chapters focus on techniques that utilize the philosophy of power minimization techniques to reduce the combinational logic soft error rate of circuits. The approaches presented in this dissertation are limited to the circuit level and architectural level, but soft error mitigation and power minimization can be achieved jointly at various levels of abstraction.

Chapter III. Combinational Logic Technology Scaling Trends

Digital circuits are mainly comprised of large memory banks to store data as well as datapath and computational blocks that consist of combinational logic and flip-flops. Single-event effects (SEE) due to radiation particle strikes in memories, flip-flops as well as logic circuits can lead to soft errors. Soft errors in memories, latches and flip-flops are caused due to ion-strikes on the sensitive nodes part of these structures. However, ion-strikes in combinational logic produce single-event transients (SETs) that must be latched by the receiving flip-flops. SETs can be masked or prevented from being latched by the flip-flops if they are 1) electrically attenuated by gates 2) logically masked from propagating through the logic and 3) do not arrive during the latching window (setup-and-hold time) of the flip. The third condition implies that if the number of latching intervals increases due to an increase in the operating frequency, the likelihood of latching SETs would be higher. Due to these factors, in older technology generations, where the frequency of operation was much lower, the combinational logic soft error (SEU caused due to latched SET originating from combinational logic) problem was not as big a threat as soft errors in memories and flip-flops. However, as technology has scaled, operating frequencies of microprocessors have steadily increased to the 5+ GHz range. Expecting such a trend in frequency of operation with technology scaling, [Shiv-02] predicted mainly through simulations and engineering insight that "combinational logic soft error rate driven by increasing clock frequencies and device scaling would exceed the flip-flop and unprotected memory error rate in the terrestrial environment". However, there have been very few experimental results to establish the relative contribution of combinational logic soft errors, flip-flop soft errors and memory soft errors, especially as a function of technology scaling and frequency of operation [Sief-12] [Maha-11] [Gill-09]. This mainly due to the fact that unlike memories and flip-flops, combinational logic circuits contain a large number of gates, input combinations, circuit topologies etc. which makes it difficult to experimentally estimate the combinational logic soft error rate of circuits. In addition to this, technology scaling also affects all aspects of logic SER (sensitive area, critical charge, frequency, masking factors, etc.) increasing the complexity of such modeling efforts. All these issues made real progress in experimentally characterizing logic SER very difficult. Due to the lack of experimental evidence, the contribution of logic soft error rate (SER) was considered negligible compared to latch SER. As a result, in the terrestrial environment, soft error mitigation efforts have focused on protecting memories and latches rather than combinational logic gates [Maiz-03] [Mukh-05] [Hazu-00]. To estimate the impact of logic SER on overall chip-level SER, designers also need to identify the frequency at which combinational logic SER exceeds latch SER. If the operating frequency is well beyond this frequency threshold then combinational logic errors will dominate overall SER. On the other hand if the operating frequency is well below this threshold, combinational logic errors may not be a major issue for the chip-level SER.

To address these issues, this work presents the alpha particle, proton and heavy-ion combinational logic SER and latch SER of identical circuits fabricated in 40-nm, 28-nm and 20-nm bulk CMOS technology nodes. Logic SER is estimated as a function of frequency and compared to the latch SER to determine the frequency at which logic SER will equal to the latch SER. By testing circuits that are representative of real-world data path circuits, results can be extended to evaluate relevant chip-level SER for modern technology nodes, especially at high frequencies. Additionally, factors such as sensitive area, single-event transient (SET) pulse-widths and masking factors can be estimated from the slope of the logic SER as a function of frequency, to predict logic SER trends for future technology nodes.

Some of the key results from this work suggest that:

1. The raw combinational logic SER for a single gate decreases as a function of technology scaling.
2. This is also accompanied by a corresponding decrease in the latch SER with scaling.

The combined effect of these two trends has several implications. Firstly, while the total SER (latch+logic SER) decreases with scaling, the proportion of the logic SER to the total SER changes, with technology scaling. As a result the threshold frequency at which logic SER exceeds the latch SER changes, which must be factored into attempts to improve the soft error resiliency of circuits. The following sections introduce the experiments and results to characterize the latch and logic SER with scaling.

3.1 Test Circuit Description & Experiments

The purpose of the test circuits designed and tested in this study was to evaluate the soft error sensitivity of combinational logic circuits as well as latches as a function of technology node and frequency of operation. Masking factors that affect combinational logic circuits are also studied. Through these experiments an attempt is made to tease out the key parameters that change as a function of technology scaling so that designers can focus on these parameters while estimating the trends in combinational logic and latch SER for different circuits. The test circuits and the experimental set-up is explained in the following sections.

3.1.1 Circuit Description

In this work, the Combinational Circuit for Radiation Effects Self-Test (C-CREST) technique was used to measure the combinational logic soft error rate [Ahlb-08]. The block diagram for this technique is shown in Figure 3-1.

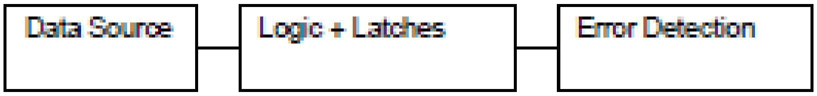


Figure 3-1 Basic structure used to evaluate flip-flop and combinational logic cross sections.

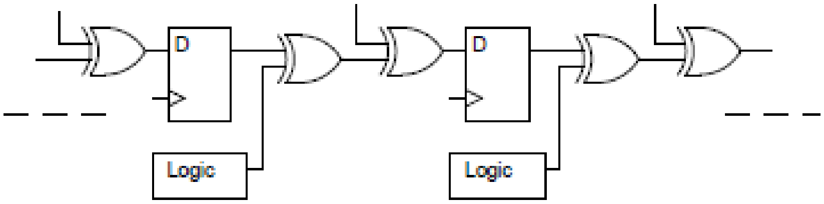


Figure 3-2 The Circuit Under Test consists of flip-flops and logic blocks.

The data source can be used to provide random input patterns as well as static patterns to the circuit. The Circuit-Under-Test (CUT) consists of a shift register design with logic circuits interleaved with flip-flops as shown in Figure 3-2. The logic circuit is separated from the critical path between flip-flops so that a circuit of arbitrary size or depth can be implemented. One flip-flop circuit along with the associated logic circuit comprises a single stage. The C-CREST design consists of 2,056 of such stages to improve the error statistics. The error detection circuit compares the correct data patterns with the output data pattern from the CUT. If there is no mismatch between the two patterns, no errors are detected and recorded. If errors occur, then a counter records the total number of errors observed. The errors recorded by the error detection circuit can be due to direct strikes on the combinational logic circuits as well as direct strikes on the flip-flops. To separate the effects of the two, a separate shift register chain was built with no logic included. This allows errors from both sources to be identified independently. The data is shifted out using an external slower clock compared to internal high-speed clock. All the error detection circuits and counters are protected against single event errors using Triple Modular Redundancy (TMR). The clock is provided by a high-speed Phase Locked Loop (PLL). The PLL was capable of being operated up to 1.2 GHz with low-noise characteristics and was also hardened against upsets.

Two variants of the C-CREST design were fabricated in each technology node : 40 nm, 28 nm and 20 nm. For both the variants, 2,056 stages were used. The flip-flop design used for both the variants was a conventional cross-coupled NAND gate D flip-flop. For the first C-CREST design, the logic circuit consisted of a block of 72 inverter gates (12 chains of six inverters each) OR'ed together The second C-CREST design consisted of a four-bit 'greater than or less than' comparator. The four-bit comparator compares two four-bit numbers, A and B. The output of the comparator is a logic '1' if B > A. The four-bit comparator was chosen because the logic depth for this circuit is close to that of modern circuits [Shiv-02(2)] [ARM-11] [Inte-10]. Additionally this circuit is used to illustrate the impact of logical masking (which is independent of technology) on the relative logic and latch SER. Table 3-1 provides details about the gate count and transistor count for individual circuits for each technology node.

Table 3-1 Number of gates, transistors, and transistor total area for different circuit types.

Circuit type	Inverter	Comparator
Total # of gates	94	46
Type of gates	83 NOT 11 NOR	26 NOT 12 2-input NAND 2 3-input NAND 6 2-input NOR
Total # of transistors	210	136

All the gates were sized to have their rise and fall time equal to that of inverter rise and fall times. The sizes and threshold voltage of the PMOS and NMOS transistors used in the inverters in each technology are listed in Table 3-2. Minimum design widths available in the technology were not implemented in each case due to variability issues, as well as to meet appropriate timing margins across the full-custom design. This was especially the case with the 20-nm circuits.

Table 3-2 Transistor widths and threshold voltages for inverters used in 40 nm, 28 nm and 20 nm technology.

Drawn L	W_P/W_N (nm)	V_{TP}/V_{TN} (mV)
40 nm	350/140	(-480/450)
28 nm	250/100	(-380/360)
20 nm	220/180	(-470/460)

3.1.2 Alpha Particle Test Details

The circuits were irradiated with 6 MeV alpha particles from a Polonium-210 source with an activity of 500 µCi, at room temperature. The alpha source was placed at a distance of less than 1 mm from the dies during testing. The size of the alpha source was 1.4 cm^2, and the maximum die size was 3 mm x 3 mm (40 nm). To account for both inter-die and experimental variability, measurements were repeated several times at each frequency and logic input value. Testing was conducted in accordance with JEDEC specifications [JEDE-06]. The nominal operating voltage was 0.9 V, 0.85 V and 0.9 V for the 40 nm, 28 nm and 20 nm circuits. The operating frequency of the circuits was varied up to 1.2 GHz using an on-chip low-noise Phase Locked Loop (PLL) circuit. The PLL was also implemented using Triple Modular redundancy for the 40 nm and 28 nm circuits and using a hardened PLL design for the 20 nm circuits.

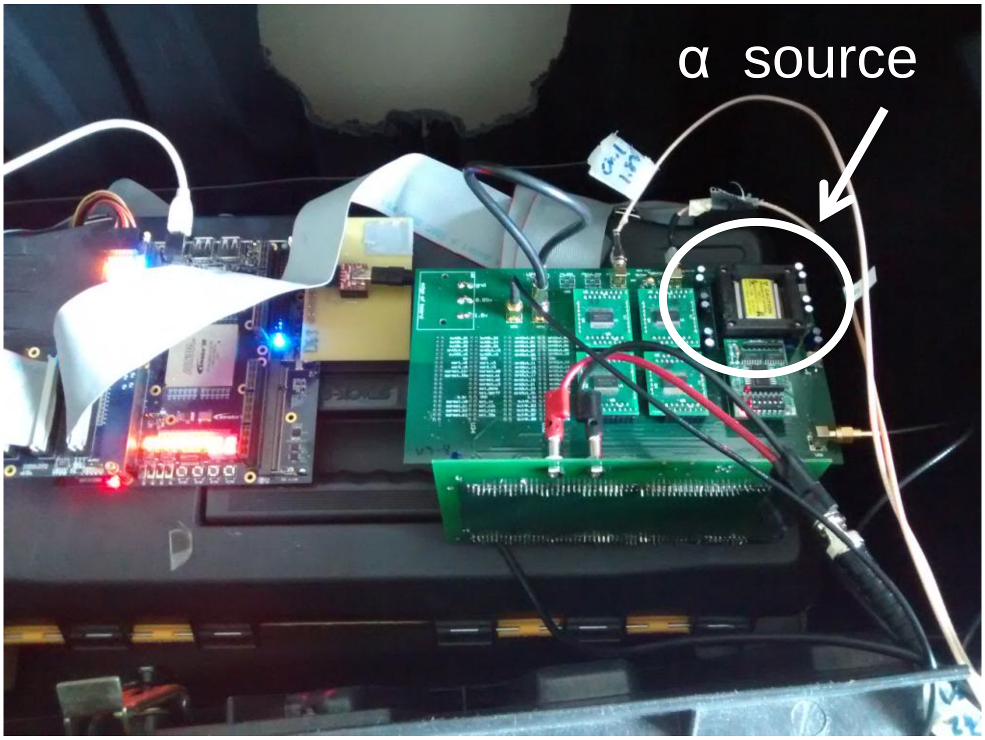


Figure 3-3 Alpha particle test set-up with alpha source encircled. The source was placed directly on top of the die. Testing was performed using a Field-Programmable Gate Array (FPGA) that communicates with the test IC.

3.1.3 Experimentally Measuring Logic Cross section

The logic and flip-flop cross-sections were measured separately. To determine the flip-flop cross-section alone, a separate shift register chain that contains no logic was built and operated at different frequencies. The total soft errors observed from the shift register chains with no logic was normalized by the fluence and the number of flip-flops, which yields the flip-flop cross section. Several trials were performed across four different dies at each operating voltage and frequency to minimize effects of experimental as well as die-to-die variations. Unless otherwise stated, the error bars in all the figures in this work represent the standard error of measurement at each data point. Each data point corresponds to at least 3 measurements each from 2 die. Thus, the experiment was repeated at least 6 times for each data point, the standard error is σ/√n, where *σ* is the standard deviation of the sample and *n* is the number of times the experiments was repeated (n=6). Similarly, the shift register chains containing logic were operated at different frequencies to record the cross-section due to logic (SETs that are latched by flip-flops) as well as direct hits on flip-flops. The flip-flop cross-section measured at each frequency was then subtracted from the total cross-section (logic +FF) measured from the chain containing logic. As observed in [Jaga-12], a small frequency dependence was observed among the flip-flop chains, however this was negligibly small compared to the frequency dependence of the logic cross-section itself. The frequency dependence of flip-flops arises from the fact that one of the latch stages is always transparent while the other stage holds data. If the slave stage is transparent, the master stage can latch transients from the slave latch portion of the flip-flop [Jaga-12]. Thus as the flip-flop cross-section frequency dependence was weak compared to the logic cross-section frequency dependence, the soft error contribution of the transparent latches has been assumed to be part of the combinational logic interfaced to the flip-flop, especially because the size of the logic is much larger compared to the latch itself. This assumption is also true for conventional circuit designs where substantial computational logic is present between flip-flop stages. On the other hand, the frequency dependence also reduces the latch cross-section due to the presence of logic between latch stages [Seif-04]. However the amount of logic directly between latch stages in this work is minimal and the total delay of the logic stages in the critical path between the latches is less than 2% of the clock period in all cases. Hence, the mechanism that leads to derating of the latch errors described in [Seif-04] does not directly apply to the results presented here.

$$Total\ \ Cross\ Section = \frac{Total Number of\ Errors}{Total Number of\ stages \times Fluence} \tag{3-1}$$

$$Logic\, Cross\ Section\, per\, Stage\ = (Total\, Cross Section) - (Flip\ Flop\ \ Cross\ Section) \tag{3-2}$$

3.2 Alpha-particle irradiation results

Figure 3-4 shows the alpha particle logic cross-section normalized for 10 inverters each for 40-nm, 28-nm and 20-nm technology nodes measured at different frequencies. There are three important trends to be noticed here.

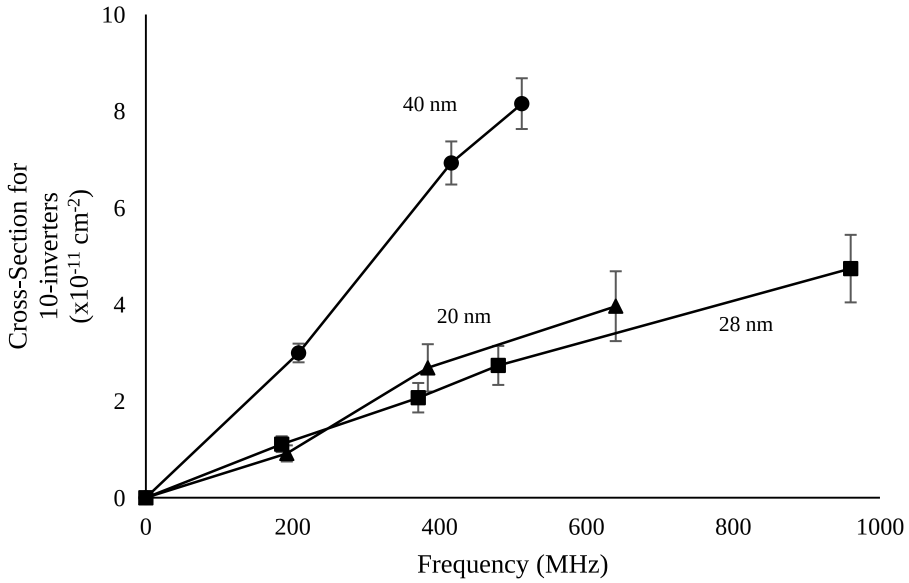


Figure 3-4 The logic cross-section for the circuits in all three technologies 40 nm, 28 nm and 20 nm increase with frequency. The slope of the logic cross-section as a function of frequency decreases from 40 nm to 28 nm. But is almost the same for the 20 nm node.

Firstly, a clear increase in logic cross-section as a function of frequency for circuits fabricated in each technology is observed. Technology scaling in general though, results in a reduction in the logic cross-section from 40-nm to 28-nm. However, the decrease in logic cross-section from 28-nm to 20-nm is not as significant as that from 40-nm to 28-nm. In the past, technology scaling has been shown to result in a decrease or saturation of the latch and memory cross-section [Dixi-11]. However this phenomenon has not been investigated extensively for logic circuits. In the following paragraphs, reasons for the trends seen in Figure 3-4 and their implications for failure-in-time (FIT) rate calculation for circuits fabricated in future technology nodes are discussed.

Estimating the combinational logic SER involves calculating the effects of sensitive area and the different masking factor at different gates across the circuit. The generalized expression for combinational logic SER is given as [Seif-01]

$$SER = \frac{\phi}{T_{clock}} \sum_{n}^{nodes} A_n \sum_{i}^{Q} prob(Q_{i,n}) \Delta q \sum_{i=t_{inj}}^{T_{clock}} upset_{j,i,n} \Delta t \qquad \textbf{3-3}$$

where SER is defines as the observed SER (from simulations) as the average over all upsets for all collected charge Q_i, all injection time t_{inj} and all nodes n of the combinational logic circuit. Φ is the particle flux, A_n is the sensitive area for the particle/environment of interest, T_{clock} is the clock period, prob(Qi, n) is the probability that charge Qi is collected at node n at injection time t_{inj}. $Upset_{i, j, n}$ = 1 if and only if the SET generated due to collected charge Qi leads to a transient that is wide enough to be latched by the receiving flip-flop. This expression incorporates information about the different masking factors like electrical masking, temporal masking and logical masking implicitly. For example when a single-event transient is produced at a node in the circuit, the capacitance or the restoring drive at that node may result in an SET that may not be wide enough to be latched by the flip-flop or the amplitude may not exceed the required threshold for propagation (generally atleast Vdd/2). In other cases this transient may be attenuated as it propagates through the logic. Similarly, just as the charge deposited may not result in a transient wide enough to be latched, the transient also may not arrive at the latching window of the flip-flop. Both these factors would lead to the SET being temporally masked. Finally, the SET at the struck node may be prevented from propagating to the latches due to the logical conditions in the circuit, in which case it would be logically masked.

In this work, the most dominant factors are incorporated in the logic cross-section as follows

$$Cross - Section\,(\sigma) = \sum_{i=1}^{n} A_i \cdot EM_i \cdot TM_i \cdot LM_i \qquad \textbf{3-4}$$

where *A* is the sensitive area, and *TM, EM and LM* are the temporal, electrical and logical masking factors respectively at each node. The temporal masking factor determines the proportion of transients that arrive during the latching interval of the receiving flip-flop and meet the setup-and-hold condition to result in an error. Electrical masking is related to the attenuation of pulses as they propagate through logic chains. Logical masking results due to input conditions on certain gates that prevent transients to propagate further in the circuit. Of these factors only the sensitive area, temporal and electrical masking factors are technology dependent. Logical masking factor is circuit design dependent and purely a Boolean property. The temporal masking factor depends on the SET pulse-width, setup-hold time of the flip-flop and the clock period. The temporal masking factor can be estimated as [Alex-11], [Nguy-03], [Lide-94]

$$TM = \left\{ \begin{array}{ll} 0 & if\ t_{SET} \leq w \\ \dfrac{t_{SET} - w}{T_{clk}} & if\ w < t_{SET} \leq T_{clk} \\ 1 & if\ t_{SET} > T_{clk} \end{array} \right\} \qquad \textbf{3-5}$$

where T_{SET} and *w* are the SET pulse-width and latching window respectively. The setup and hold time or latching window is the same for all the nodes in the circuit, the only difference being the SET pulse-width. Thus the *TM* of a node for a fixed charge deposition value can be approximated as [Shiv-02]

$$TM = \left\{ \frac{t_{SET}}{T_{clk}} \right\} \qquad 3\text{-}6$$

where t_{SET}, is the single event transient (SET) pulse-width, and T_{clk} is the clock frequency.

Thus the technology dependent cross-section can be expressed as

$$\sigma = \sum_{i=1}^{n\ nodes} A_i \cdot \frac{t_{SET_i}}{T_{clock}} \cdot EM_i \qquad \textbf{3-7}$$

Thus the cross-section $\sigma = f(A, t_{SET}, EM)$. Besides, in Equation 4, the product of the terms in parenthesis comprising of A, t_{SET}, EM and LM is the slope of the logic cross-section as a function of frequency. The technology dependent factors that influence the logic cross-section are the only the sensitive area, SET pulse-width, and electrical masking. In order to determine the impact of each of these factors and explain the results shown in Figure 3-4, qualitative simulations were performed using SPICE and calibrated 3-D TCAD models.

3.2.1 Impact of Sensitive Area and SET Pulse-width

In order to estimate the scaling trends of the sensitive area and the SET pulse-width, 3D-TCAD simulations were used. 3D-TCAD simulations were setup in such a way that single-event strikes were simulated on an OFF NMOS transistor. The restoring device was a PMOS transistor in an inverter configuration. The widths of the transistors were as shown in Table 3-2. The physical drain area was estimated using the layout and the drain extension values from the SPICE models of the process design kit (PDK).

The alpha particles emerging from the Po-210 source used to irradiate the circuits were close to mono-energetic (6 MeV) and isotropic. In such cases the maximum linear energy transfer (LET) of the particles is not likely to exceed 1 MeV/cm^2-mg [Gadl-08]. The single-event strikes with LET = 1 MeV/cm^2-mg were simulated in raster scan fashion on the OFF NMOS transistor and the single event pulse-width was monitored. The single-event pulse-widths along the cut-line for the 20-nm NMOS transistor case is shown in Figure 3-5. As seen in Figure 3-5, the single event pulse-width values reduces rapidly around the edge of the drain depletion edges. This is because very little charge is collected by the reverse biased drain region due to diffusion. As a result the charge collection area is limited to the area of the drain itself. Besides, since the SET value measured in TCAD is fairly constant around the center of the drain region, the SET pulse-width for a strike anywhere in the drain region can be approximated using the maximum SET pulse-width observed. In this case this value for the 20 nm inverter was ~8 ps. It is important to remember that the pulse-width recorded in TCAD and shown in Figure 3-5 is not guaranteed to be the precise pulse-width in the physical circuit due to differences in calibration, supply voltage drops across the chip that can influence SET pulse-widths etc. However we can rely on the *qualitative trend* in comparing the SET pulse-widths for the different technologies.

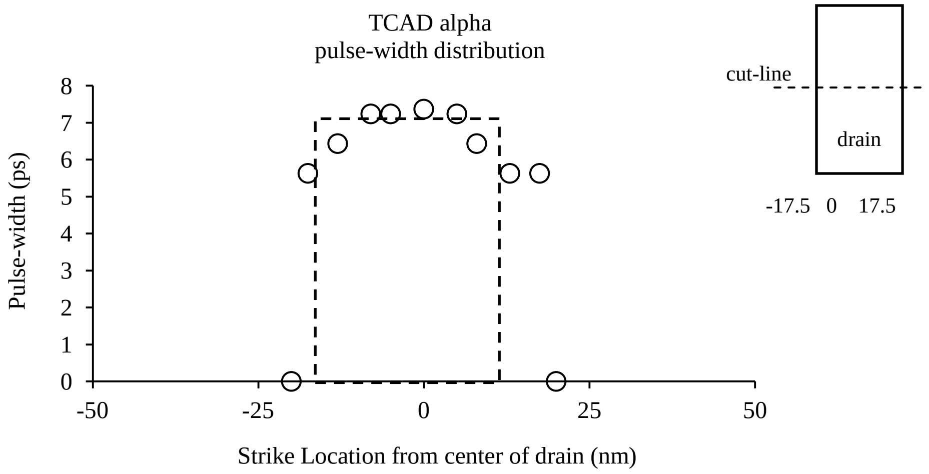


Figure 3-5 The SET pulse-width is plotted as a function of the distance from the center of the drain (0 in inset) along the cut-line shown. It is clear that the SET pulse-width reduces to 0 at the edge of the drain depletion region for alpha particles. Thus the drain area itself is a reasonable approximation of the sensitive area (A) in Equation 2. Similarly since the SET pulse-width rolls off very sharply towards the depletion edge of the drain, the maximum value of the SET pulse-width is a reasonable estimate of the SET pulse-width for strikes anywhere in the drain region. This is indicated using the dashed rectangle.

The procedure described above was repeated for the 28-nm and 40-nm inverter cases with the NMOS transistor struck in TCAD. The same assumptions about the area (sensitive area = drain region) and the SET pulse-width (maximum SET pulse-width) recorded in TCAD are made. The product of the struck NMOS transistor area (sensitive area *A* in Equation 4) and the SET pulse-width (t_{SET} in Equation 6) recorded from TCAD was calculated. This product ($A{\cdot}t_{SET}$) of the sensitive area and pulse-width for each technology yields the results shown in Figure 3-6. This product includes the first two terms of Equation 6 that influence the slope of the logic cross-section versus frequency curve.

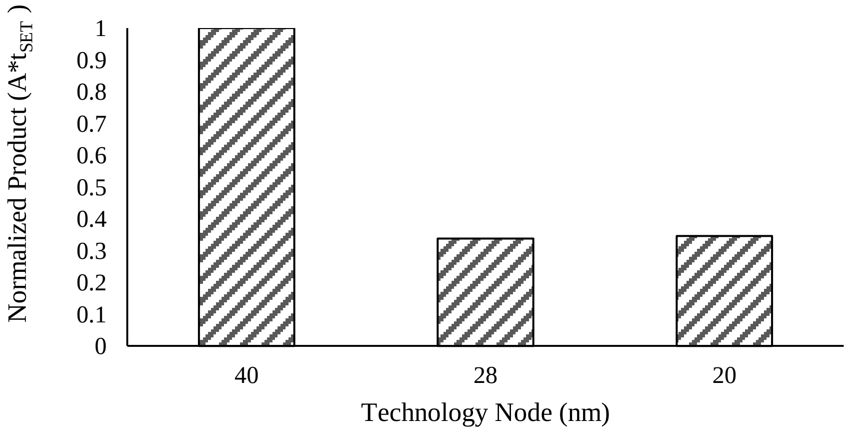


Figure 3-6 The product of the sensitive area and the SET pulse-width reduces from 40 nm to 28 nm, but increases marginally from 28-nm 20-nm. This is mainly because the charge collection area for the NMOS transistors is larger due to the choice of widths shown in Table II. However the SET pulse-width reduces across all three technologies since the restoring drive decreases marginally for the transistors PMOS transistors chosen. The overall product of area and SET pulse-width reduces from 40-nm to 28-nm but is comparable for 28-nm and 20-nm.

The results in Figure 3-6 are qualitatively in line with those in Figure 3-4. Due to technology scaling two important effects occur. Firstly the size of both NMOS and PMOS transistors reduces from 40 nm to 28 nm. As a result the active area for alpha particle strikes to result in upsets, decreases. On the other hand the current drive per µm increased with technology scaling from 40 nm to 28 nm. Increased restoring drive results in shorter transients. As a result the cumulative effect of technology scaling is that both, the active area and the SET pulse-widths decrease from 40 nm 28 nm. This trend is observed while scaling from 40-nm to 28-nm for both the experimental results in Figure 3-4 as well as the simulations in Figure 3-6. On the other hand however, the NMOS transistor widths in the 20-nm circuits were larger than those of the 28-nm NMOS transistor widths. This choice was made to evaluate the relative impact of drain area on the soft error sensitivity of combinational logic circuits compared to other masking effects and to determine whether the soft error cross-section scales well with area. Due to the choice of larger NMOS transistors in 20-nm compared to 28-nm but a marginally higher drive current that reduces the SET pulse-width, the product of the sensitive area of the drain and the SET pulse-width remains nearly same for these two technologies. In other words the decrease in SET pulse-width due to scaling from 28 nm to 20 nm is compensated by the increase in area.

3.2.2 Impact of Electrical Masking on Logic Cross-Section

Apart from the sensitive area and the SET pulse-width, the electrical masking also affects the SET pulse-width and subsequently the logic cross-section. For modern logic designs, usually the number of logic gates per latch stage (pipe-line stage) does not exceed 10 [Shiv-02(2)], [ARM-11], [Inte-10]. With such a small number of gates in a logic path, the effects of electrical masking are expected to diminish [Shiv-02]. Additionally, smaller gate delays reduce the impact of electrical masking on pulse-width propagation [Mass-08]. In particular the transient propagates with minimal attenuation if it exceeds the rise and fall time of the succeeding gates [Mass-08]. To characterize electrical masking, a 10-stage logic chain with uniform FO4 load was simulated using the 40 nm, 28 nm and 20 nm PDK. The average pulse-shortening was less than 1 ps with charge deposition values up to 20 fC. Thus, electrical masking can be neglected in comparison to the sensitive area scaling and transient pulse-width scaling due to differences in current drive while estimating the logic cross-section. In fact the FO4 delays were 22 ps, 16 ps and 11 ps for the 40 nm, 28 nm and 20 nm technologies. Thus electrical masking, if any, is likely to decrease with technology scaling. Thus it is possible, that SETs in the 20-nm circuits are attenuated less compared to 28-nm and 40-nm circuits.

Thus, the key factors that affect the alpha particle logic cross-section in Equation 4 are primarily the product of the sensitive area and the transient pulse-widths. Ordinarily with technology scaling, the active device sizes decrease and the drive currents increase, leading to reduction in the logic cross-section. However for the data set shown in Figure 3-4, the logic cross-section for the 20-nm node is comparable to the 28-nm circuits. This is because larger than minimum sized devices were used which compensate the effects of reduced SET pulse-widths for the 20-nm circuits. The trends in Figure 3-6 generally replicate the results obtained experimentally and shown in Figure 3-4. The important result however, is that the trends produced by the models support the experimental results. Technology scaling would consistently result in reducing the logic cross-section if the areas of the individual transistors in 20-nm are also reduced.

3.3 Comparison of Alpha-particle logic and latch cross-sections

From the point of view of chip-level SER, the contribution of both logic errors and latch errors need to be considered. Figure 3-7 shows the ratio of combinational logic errors to latch errors for each technology node. The logic SER of 10 inverters was normalized by the latch cross-section for each technology. Again, the trend in the ratio of logic to latch errors follows the same trend as observed in Figure 3-4. However the major difference here is that the 20-nm logic SER to latch SER curve is closer to the 40-nm curve than the 28-nm curve (as observed in Figure 3-7). The primary reason for this is that the latch cross-section decreases significantly from 40-nm to 28-nm. However the decrease in cross-section is not as significant from 28-nm to 20-nm. The DFF cross-section is indicated in Table 3-3 for each technology. These results are consistent with those that suggest that the SRAM SER/bit decreases or saturates with scaling [Dixi-11].

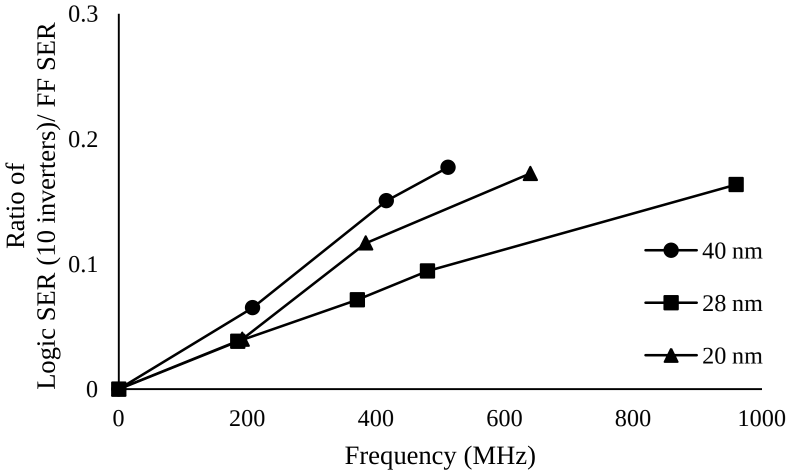


Figure 3-7 Ratio of logic SER to latch SER. As the latch SER scaling trend is different from logic SER trend, 20 nm logic SER to latch SER ratio is higher than 28 nm.

Table 3-3 Standard NAND gate D flip-flop cross-section for 40 nm, 28 nm and 20 nm

Technology	Cross-Section (cm^2)
40 nm	4.6x10^{-10}
28 nm	2.9x10^{-10}
20 nm	2.3x10^{-10}

Thus, although both the logic and latch cross-sections decrease with scaling, if the rate of decrease is dissimilar the ratio of the two could vary from one technology to the other. In this work, experimental results show that the logic SER scales at a different rate with technology compared to the latches, mainly due to differences in transistor sizes and their single-event effect (SEE) sensitivity. These two factors result in a lot of variation in the proportion of logic SER to latch SER for the three technology nodes considered. It is therefore important to carefully consider the total contribution of logic SER and latch

SER for the frequency of operation. As an example, at 20-nm, if the operating speed is 2.5 GHz (which is well within the operating range of modern ICs), the logic SER from 10 inverters in this case will be nearly equal to latch SER. It is however crucial to note that the total SER at any given frequency decreases due to scaling, because both the latch and logic cross-sections decrease with scaling. But logic SER could become the dominant contributor to the total SER at much lower frequencies with scaling.

3.3.1 Impact of Logical Masking

The logic SER contribution decreases due to the effects of logical masking. The 4-bit comparator was used as a test-bed to evaluate the effects of logical masking on the logic SER and compare with the latch SER. The comparator was used because it represents an average sized circuit in terms of logic depth and size. The number of gates in the circuit (~30) were comparable to the average number of gates per output (~25) in case of ISCAS-85 benchmark combinational logic circuits [ISCAS-85]. The impact of logical masking for two different input combinations were tested and the results are illustrated in Figure 3-8. The condition A = ‘0000’ and B = ‘1000’ represents the case where very few logic gates (high-level of logical masking) are used to establish if B>A. On the other hand the condition where A = ‘1000’ and B = ‘0000’ utilizes almost all the logic gates to compute the result (less logical masking). Results suggest that for the case where logical masking is highest (A = ‘0000’ and B = ‘1000’) the logic SER can be between 0.5-3% of the latch SER which is quite negligible. In cases where the logical masking is low (A = ‘1000’ and B = ‘0000’), the logic SER can be as much as 20-30 % of the latch SER at 500 MHz. Thus logical masking can result in the logic SER varying up to an order of magnitude for different inputs.

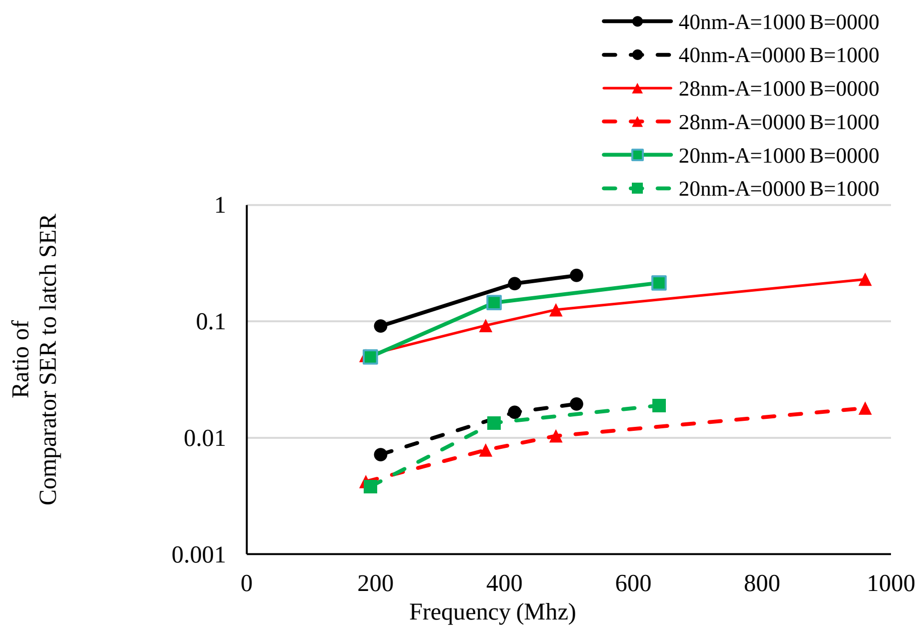


Figure 3-8 SER which is quite negligible. In cases where the logical masking is low, the logic SER can be as much as 20-30 % of the latch SER at 500 MHz.

3.3.2 Impact of Latch Design

The combinational logic soft error rate is a strong function of the logical masking factor. The differences in cross-section due to differences in input conditions and masking factors leads to differences in the ratio of the combinational logic SER to latch SER by as much as a factor of 15 as shown in Figure 3-8. Similarly the design of the latch itself can introduce differences in this ratio. Again, the technology trends of the latch cross-section will affect the ratio of the combinational logic and latch SER too. Figure 3-9 shows the measured latch cross-section of two different latches. The latch cross-sections differ by as much as 5X at 40 nm and almost 6X at 28 nm. Thus scaling led to a stronger reduction in cross-section for the hardened latch than for the soft latch. The combinational logic SER for 10 inverters was normalized to both these latches and the results are plotted in Figure 3-10. The key point to be stressed here is that scaling leads to a decrease in both the latch and logic cross-section. But as the cross-section of the hard latch is lower, the ratio of logic cross-section of 10 inverters to the hard latch cross-section is higher than the ratio of logic to soft latch cross-section. However, regardless of the latch design, the ratio of logic to latch cross-section is higher for the 40 nm designs compared to the 28 nm designs. Again, this ratio depends on both the logic scaling rate as well as the latch scaling rate with technology. The ratio of logic to latch as function of technology could change if the latch cross-section with scaling, decreases at a rate that is faster than that for the logic cross-section with technology.

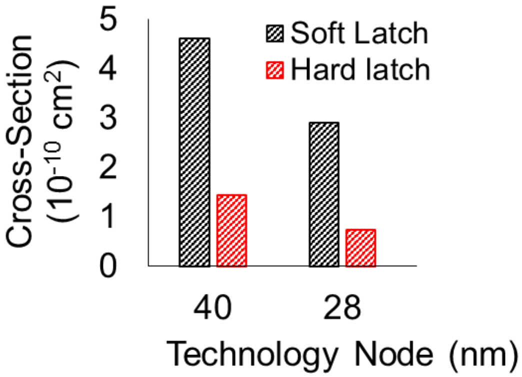


Figure 3-9 Latch cross-section for two different (soft and hard) latches in 40 nm and 28 nm.

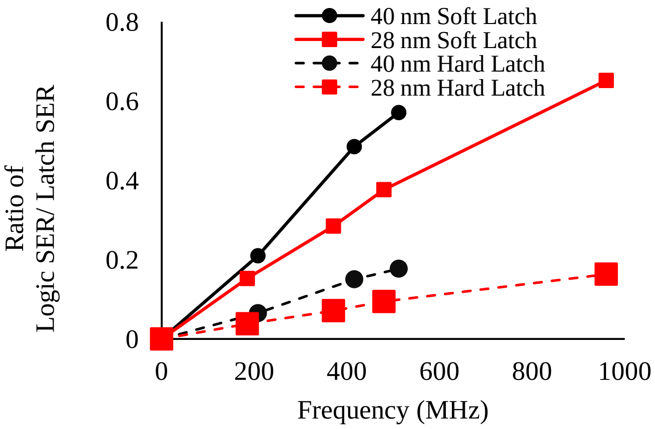


Figure 3-10 Ratio of logic cross-section to latch-cross section is higher with the use of hard latches. The ratio of logic to latch cross-section decreases from 40 nm to 28 nm implying that the rate of scaling of logic is faster than that of latches regardless of the latch design used.

3.4 Heavy-Ion Irradiation Results

Heavy-ion irradiation was performed at Texas A&M Cyclotron and Lawrence Berkeley National Lab. The 40 nm, 28 nm and 20 nm circuits were tested as functions of Linear Energy Transfer (LET) at very low frequency (2 KHz) and at 500 MHz to measure the flip-flop and logic cross-sections, respectively. The 40 nm circuits were not tested at high frequency. The flip-flop cross-section consistently decreases as a function of technology scaling as shown in Figure 3-11. Sensitive area scaling is the primary reason for this decrease in the cross-section with technology scaling. The ratio of logic cross-section (10 inverters) to that of the flip-flop, plotted in Figure 3-12, however, shows two interesting trends.

1. Nature of the curve (LET dependence): The nature of the curve suggests that initially the ratio of logic to latch cross-section increases linearly with LET but

then increases sharply at higher-LETs. This is observed in the case of both 28 nm and 20 nm circuits. This suggests that this trend is real and not an artifact of experimental error or inaccuracy.

2. High-LET anomaly: At very high LET, the ratio of the logic to latch cross-section for the 20 nm circuits is higher than that for the 28 nm circuits.

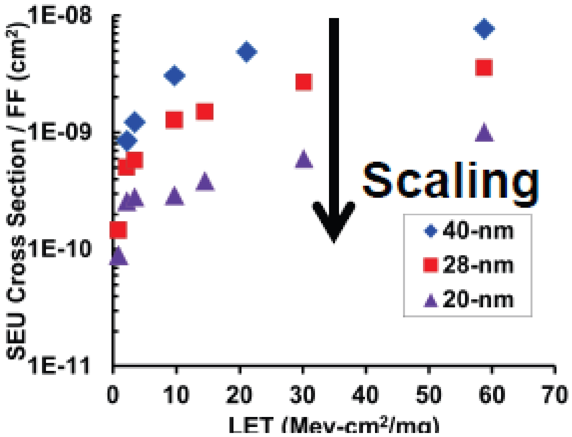


Figure 3-11 Ratio of logic cross-section to latch-cross section is higher with the use of hard latches. The ratio of logic to latch cross-section decreases from 40 nm to 28 nm implying that the rate of scaling of logic is faster than that of latches regardless of the latch design used.

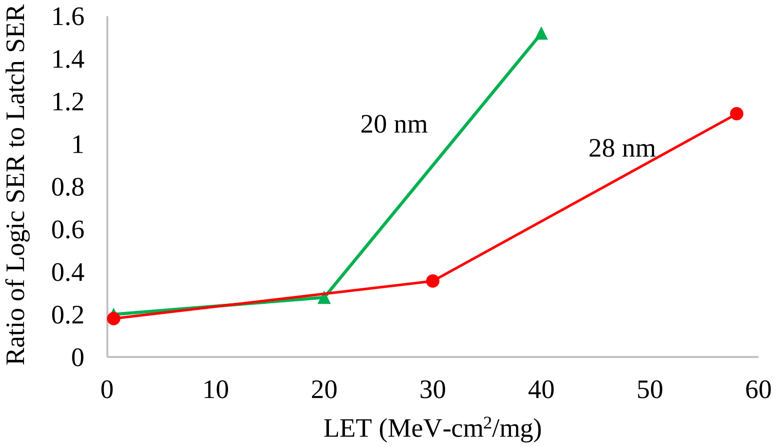


Figure 3-12 Ratio of logic cross-section to latch-cross section is higher with the use of hard latches. The ratio of logic to latch cross-section decreases from 40 nm to 28 nm implying that the rate of scaling of logic is faster than that of latches regardless of the latch design used.

3.4.1 Nature of the curve (LET dependence)

The flip-flop cross-section as function of LET is well-known and follows the Weibull curve as shown in Figure 3-11. The growth of the cross-section as function of LET is initially exponential and then saturates. This behavior can be captured by a piecewise function as follows :

$$\sigma_{flip-flop} \;\; \alpha \left\{ \begin{array}{ll} e^{\alpha LET} & if \; LET < LET_c \\ \sigma_{sat} & if \; LET > LET_c \end{array} \right\} \qquad \textbf{3-8}$$

where the cross-section σ depends on the LET exponentially up to a certain critical LET_c and then saturates to a constant value σ_{sat} beyond this LET value.

The logic cross-section depends on the sensitive area as well as the SET pulse-width. The two factors are difficult to extract from the experimental measurement of the logic cross-section of 10 inverters at a high frequency because the probability of striking the circuit and generating transients that latch is included implicitly. However, SET measurement circuits have been used in the past to characterize the pulse-width and the cross-section. These circuits do not operate on a clock but instead measure all the transients above a certain width that are created. A numerical count of the number of transients normalized by the fluence is the raw cross-section for combinational logic SETs, ie, the rate at which SETs are generated. The latching probability indeed depends on the pulse-width and the clock frequency. The frequency in the case of the experimental results reported in this work are concerned was fixed. The only two factors that change with LET are the raw cross-section and the SET pulse-width and both are expected to increase. Results from [Nara-08] suggest that the raw combinational logic cross-section also grows exponentially with LET. These are shown in Figure 3-13. The latch cross-section saturates because striking even outside the sensitive drain area leads to enough charge collection for the latch to upset. Beyond a certain distance diffusion charge collection is limited and very little or no charge is collected for an upset. However, in case of logic, the struck gate only produces a wider transient in response to higher LET which deposits more charge. On the other hand the pulse-width was experimentally measured to increase linearly with the LET. Simulations also confirm this trend [Dasg-07]. Thus the logic cross-section is proportional to the product of the sensitive area and the SET pulse-width. As shown in Figure 3-13, the sensitive area is exponentially dependent on the LET and the pulse-width varies linearly with LET. The logic cross-section in Equation 7 can be expressed as follows.

$$\sigma_{logic} \ \alpha \ (A_i) \cdot (t_{SET_i})$$

Accounting for the LET dependence of sensitive area and pulse-width we get,

$$\sigma_{logic} \ \alpha \ (e^{\beta LET}) \cdot (\lambda LET) \qquad \textbf{3-9}$$

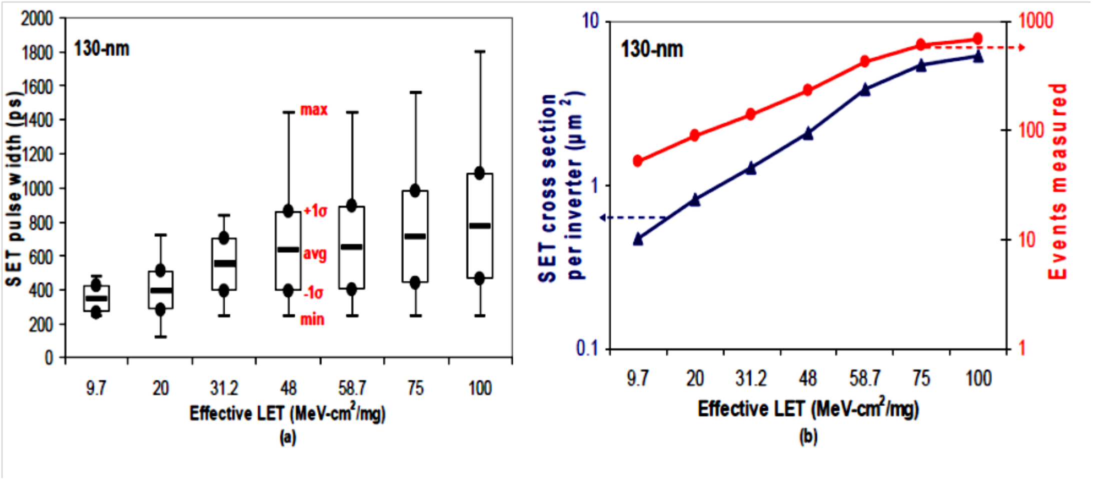


Figure 3-13 Box plot indicating (a) average maximum and minimum SET pulse-width as a function of LET for 130-nm process and (b) SET cross-section per inverter and number of events measured as a function of effective LET [Nara-08].

Based on these observation, the functional dependence of the flip-flop and logic cross-section on the LET has been captured in Equation 8 and 9. Therefore the ratio of the logic cross-section to latch cross-section can be expressed as

$$\sigma_{flip-flop} \quad \alpha \left\{ \begin{array}{ll} \dfrac{(e^{\beta LET})\cdot(\lambda LET)}{e^{\alpha LET}} & if\ LET < LET_c \\ \dfrac{(e^{\beta LET})\cdot(\lambda LET)}{\sigma_{sat}} & if\ LET > LET_c \end{array} \right\} \qquad \textbf{3-10}$$

For LET values less than LET$_c$, it is reasonable to assume that the effect of the exponentials have no net effect on the ratio of logic cross-section to latch cross-section leaving only a linear dependence. This could be the reason for a linear dependence of the ratio of logic to latch cross-section as seen in Figure 3-12. For a value, of LET greater than LETc, however, the latch cross-section saturates while the logic cross-section continues to increase. As a result the ratio of the two quantities increases sharply as seen in Figure 3-12. Thus the LET could have very different impacts on the logic and latch cross-sections. In general, the ratio of the logic to latch cross-section for the heavy-ions is higher than the alpha particles. In fact the ratio is 0.3 for LET = 20 MeV/cm^2-mg for the 20 nm circuits at 500 MHz. Indeed at higher frequencies, this ratio is expected to increase linearly with frequency as observed in the alpha particle irradiation results.

3.4.2 High-LET anomaly

It is seen in Figure 3-12 that the ratio of the logic cross-section to the latch cross-section is higher for the 20 nm circuits compared to the 28 nm circuits, especially at higher LETs. As explained earlier, the circuits were designed to have sizes and drive currents that would result in approximately similar cross-sections. However the high-LET results seem to suggest a substantial increase for the 20 nm circuits compared to 28 nm. More than one reason could be the cause here. The layout was done using an automated process which could introduce additional pulse quenching effects in the 28 nm designs compared to the 20 nm designs [Ahlb-08]. The placement of well taps too, strongly affects charge dissipation at high-LETs [Amus-08]. Both these factors can significantly affect the effective cross-section and pulse-widths of the two circuits relatively speaking. With limited experimental results and lack of knowledge of the layout pulse quenching and bipolar enhancement effects are strong candidates to explain the differences between 20 nm and 28 nm differences at high-LET.

3.5 Conclusions

This work shows that for the terrestrial environment, scaling results in a reduction in *both*, the SER for latches as well as that for logic gates. However, the proportion of logic SER to latch SER for a given circuit depends on several parameters like frequency of operation, logic size, topology etc. in addition to latch SER. For the circuits tested in this work, the logic SER for benchmark 10 inverters is about 20% of the latch SER at 500 MHz for the 20-nm node, whereas it is about 10% of the latch SER at 40 nm node. For an average-sized circuit like a 4-bit comparator the logic SER can be anywhere from ~1% to 20% of the latch SER at 500 MHz. At higher frequencies, the logic SER will certainly be comparable to the latch SER and could exceed it as well. It therefore becomes imperative to estimate the logic and latch SER accurately to best determine the hardening strategy to reduce the total chip-level SER. If logic errors dominate, flip-flop hardening alone will not reduce the overall SER. Results presented in this work can be extended to predict the logic cross-section of arbitrary circuits. These results will lead to efficient logic soft error mitigation strategies for future technology nodes. In the following chapters, various power-mitigation schemes for combinational logic soft error mitigation are presented.

Chapter IV. Circuit Partitioning for Power-Aware SER Mitigation

In the previous chapters, the impact of various factors on the combinational logic SER and the different mitigation techniques have been discussed. Some of the common strategies employed at gate-level is to selectively increase sizes of certain critical transistors in the design to reduce the transient pulse-widths. This however results in area and more importantly power overheads. In this work, the co-optimization of logic SER and power is emphasized. A formal framework based to mitigate combinational logic soft errors and reduce power consumption of arbitrary logic circuits is presented. Some of the key research findings from this work include:

1. Combinational logic mitigation and power minimization can be jointly achieved by targeted reduction in the number of sensitive nodes of the circuits.
2. Specific conditions to achieve near-optimal reduction of the combinational logic SER and power consumption with minimum area overheads is presented
3. Repeated application of the presented techniques results in improved soft error resilience and power minimization but can be shown to achieve a theoretical maximum.

4.1 Shannon Expansion Theorem

Several power reduction techniques exist to achieve low power operation at different level of abstraction. In this work, the reduction in dynamic power consumption is stressed through circuit and architectural approaches. Minimizing power consumption especially for high-speed circuits in active mode mainly involves reducing the dynamic power consumption. The dynamic power consumption is dependent on several factors and is given by Equation 1

$$P = (\sum_{i=1}^{n=nodes} \alpha_i \cdot C_i) V^2 f \qquad \textbf{4-1}$$

where C_i, is the switching capacitance of node i among n nodes of the circuit, that has a switching probability α_i. V is the supply voltage for a circuit which is clocked at a frequency of f. Reducing one or more of the factors in Equation 1 results in a reduction in the dynamic power consumption of the circuit. The work discussed in this paper primarily focuses on reducing the switching probability of certain nodes in the circuit. Shannon's Decomposition Theorem or Shannon's Expansion Theorem is a well-known technique to effectively reduce the probability (α) of circuit nodes [Lava-95]. Any Boolean function can be represented in a multiplexed form controlled by a single variable as follows [Lava-95]:

$$f(x_1, x_2, x_3, \ldots x_n) = x_1 \cdot f(1, x_2, x_3, \ldots x_n) + \overline{x_1} \cdot f(0, x_2, x_3, \ldots x_n) \qquad \textbf{4-2}$$

$$f(x_1, x_2, x_3, \ldots x_n) = x_1 \cdot CF_1 + \overline{x_1} \cdot CF_2$$

where $CF_1 = f(1, x_2, x_3 .. x_n)$, $CF_2 = f(0, x_2, x_3 .. x_n)$, are the co-factors obtained by evaluating the original expression for $x_1 = 1$ and 0 respectively. The expression in the above equation is formally called the Shannon Decomposition or Shannon Expansion about the variable x_1. Here x_1 is called the control variable or partitioning variable.

Depending on the logic value of the control variable, the appropriate co-factor is selected to compute the output. For any input condition of the circuit, only one of the two co-factors is required to compute the output. Thus, such an expression can be structurally translated into a multiplexed form where the two inputs of the multiplexer are the cofactors and the multiplexer select line is control variable selected for Shannon expansion or partition. Shannon expansion can be repeatedly applied within each co-factor to obtain multiplexed forms of the co-factors themselves. Figure 4-1 illustrates how Shannon expansion can be used to partition circuits. The individual co-factors can be obtained by setting a = 1 and a = 0, a being the control variable in this case. The original circuit and the Shannon equivalent are shown in Figure 4-1.

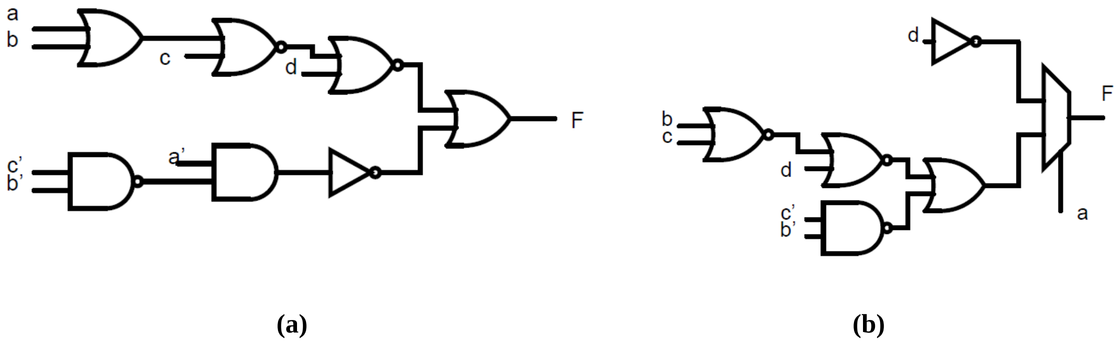


Figure 4-1 Original circuit (a) and its Shannon Equivalent implemented with multiplexers (b)

From the above analysis certain observations can be made.

1. The number of gates (size) of the individual co-factors is ≤ to the original circuit.
2. The sum of the gates (size) of the co-factors in the Shannon equivalent circuit is ≥ the original circuit.

3. In general, the size of the two co-factors need not be identical and depends on the individual Boolean expression.
4. The worst-case delay of the circuit may increase due to the addition of the multiplexer.

Each of the above factors is influenced by the choice of the variable that used to partition the circuit. Depending on the choice of the variable, the area of the partitioned circuit will have different sizes relative to the original circuit. The variables can be selected to minimize the area of the partitioned circuit and/or to minimize delay of the resultant circuit. The above observations are useful in understanding how power and SER can be reduced with the use of Shannon's expansion. Partitioning circuits into co-factors using Shannon expansion results in only one of the two co-factors being actively used for computing the output. Power savings can be achieved in three distinct ways.

1. Partitioning the circuit allows logic minimization in each of the co-factors thus reducing the number of switching nodes leading to lower dynamic power consumption.
2. As only one co-factor is used for computation, the inputs to the other co-factor can be disabled or even power-gated in certain cases [Aldi-94]. Thus if the co-factors generated from partition are smaller than the original circuit, then fewer nodes switch in each of the co-factors while the other co-factor is gated and contributes to only an increase in the leakage power. This is illustrated in Figure 2.
3. Partitioning the circuit effectively results in better path balancing. When multiple paths that converge have unequal delays, spurious transitions can

occur leading to glitching power dissipation. Shannon expansion if performed correctly using the right variable to partition the circuit can result in delay balancing, thus reducing the glitching power consumption [Maha-00].

Figure 4-2 provides an illustration of how power can be saved with the use of Shannon Decomposition of co-factors. The given function can be expanded into its co-factors with respect to an input variable, in this case V. Depending on the value of the variable, only one co-factor sub-circuit is computed, while the other is disabled by the use of AND gates at the input side of the two co-factors. The gating variable used for the AND gate is the control variable used for partition itself. When the variable V = 1, CF_1 is selected and the output is computed. On the other hand since V = 1 (V'=0), the other co-factor is gated on the input side using the AND gates with dominating value of V' = 0 used to prevent any switching activity in the internal nodes of this this co-factor (CF_2). Thus the internal switching activities of the overall circuit are reduced as only one co-factor switches for any value of the control variable. Accordingly, the power consumption is reduced considerably. In this work, the gating of input co-factors is used so that power can be saved along with combinational logic SER reduction.

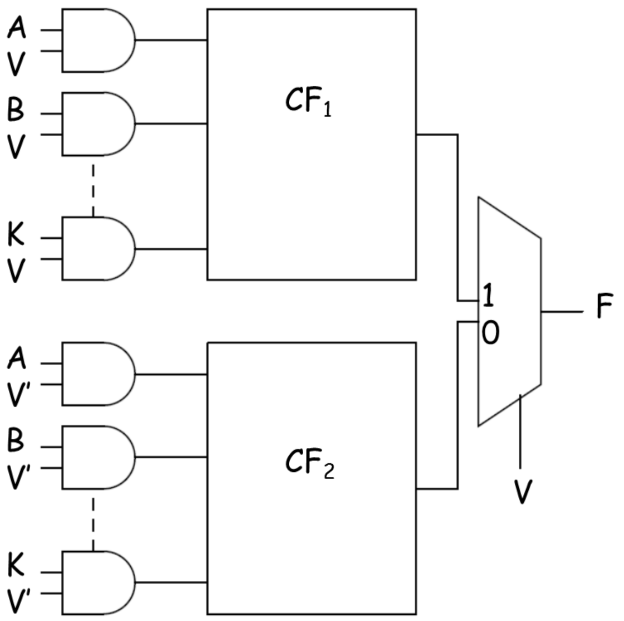


Figure 4-2 Input-gating of the co-factor that is not selected is achieved by using AND gates. If the variable select CF1 then the AND gates allow the inputs to propagate to the co-factor CF1. On the other hand, as the other co-factor is guarded with AND gates with complementary value of the controlling variable, the nodes in co-factor CF2 retain their previous value and do not switch.

4.2 Shannon's Expansion Theorem to reduce logic SER

When circuits are synthesized from Boolean descriptions of functions, the choice and placement of gates influences the combinational logic SER. For certain input values only few of the gates in the entire circuit are needed to compute the output. Single-event transients at the other gates however, can propagate to the output and result in single-event errors if they are latched by the receiving flip-flops. Minimizing the presence of such gates can lead to a significant reduction in the combinational logic SER through the elimination of single-event transients from certain gates. Shannon expansion allows logic minimization within the co-factors themselves which results in the removal and reduction of the redundant gates. This can be understood through the following example. The example circuit and its Shannon equivalent are shown in Figure 4-3. The variable ‘a’ is used to partition the circuit. Consider the case where, b=c=d=0 for simplicity. When a = 1, SETs due to strikes at 5 nodes can propagate to the output. On the other hand, in the Shannon equivalent, only 2 nodes, including those within the multiplexer that are vulnerable. The logic conditions are shown in Figure 4-3. Thus a reduction of 60% in the sensitive area is achieved through logic minimization for this co-factor.

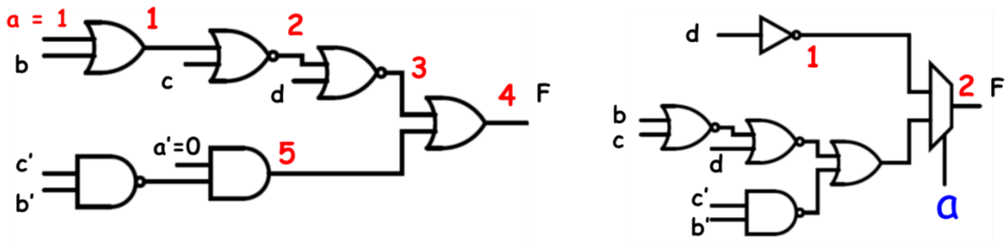


Figure 4-3 Sensitive nodes (in red) for the original circuit and its Shannon equivalent. The number of sensitive nodes reduce from 5 to 2 due to partition.

Similarly, for the case where a = 0 and b=c=d=0, SETs due to strikes at 6 nodes can propagate to the output. On the other hand, in the Shannon equivalent, 4 nodes, including those within the multiplexer that are vulnerable. The logic conditions are shown in Figure 4-4. Thus a reduction of 16% in the sensitive area is achieved through logic minimization for this co-factor.

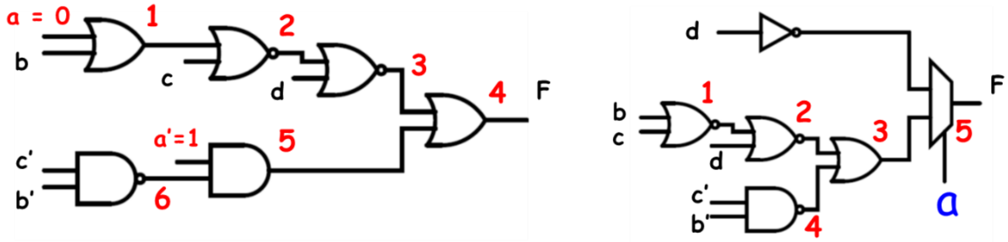


Figure 4-4 Sensitive nodes (in red) for the original circuit and its Shannon equivalent. The number of sensitive nodes reduce from 6 to 5 due to partition.

Comparing between these two cases illustrates that Shannon partition allows logic-minimization within each co-factor. This allows for sensitive area reduction within each co-factor. As only one of the two co-factors is active for any input value for a, a significant reduction in the total logic SER can be achieved through Shannon partition. This comparison also highlights the fact that the partition using certain variables may result in co-factors that are of unequal size. In the extreme cases very small co-factors may be achieved and in other extremes very large co-factors whose size is almost the same as the original circuit may be obtained. Thus the key task must be to partition the circuit in a way that the maximum logic minimization is achieved.

4.3 Framework to Evaluate the Power and SER of partitioned circuits

Estimating the combinational logic SER involves calculating the effects of sensitive area and the different masking factor at different gates across the circuit. The generalized expression for combinational logic SER is given as [Seif-01]

$$SER = \frac{\phi}{T_{clock}} \sum_{n}^{nodes} A_n \sum_{i}^{Q} prob(Q_{i,n}) \Delta q \sum_{i=t_{inj}}^{T_{clock}} upset_{j,i,n} \Delta t \qquad \textbf{4-3}$$

where SER is defines as the observed SER (from simulations) as the average over all upsets for all collected charge Q_i, all injection time t_{inj} and all nodes n of the combinational logic circuit. Φ is the particle flux, A_n is the sensitive area for the particle/environment of interest, T_{clock} is the clock period, prob(Qi, n) is the probability that charge Qi is collected at node n at injection time t_{inj}. $Upset_{i, j, n}$ = 1 if and only if the SET generated due to collected charge Qi leads to a transient that is wide enough to be latched by the receiving flip-flop. This expression incorporates information about the different masking factors like electrical masking, temporal masking and logical masking implicitly. For example when a single-event transient is produced at a node in the circuit, the capacitance or the restoring drive at that node may result in an SET that may not be wide enough to be latched by the flip-flop or the amplitude may not exceed the required threshold for propagation (generally atleast Vdd/2). In other case this transient may be attenuated as it propagates through the logic. Similarly, just as the charge deposited may not result in a transient wide enough to be latched, the transient also may not arrive at the latching window of the flip-flop. Both these factors would lead to the SET being temporally masked. Finally, the SET at the struck node may be prevented from propagating to the latches due to the logical conditions in the circuit, in which case it would be logically masked.

In this work, the most dominant factors are incorporated in an SER metric as follows

$$SER\ metric\ = \sum_{i=1}^{n} A_i \cdot EM_i \cdot TM_i \cdot LM_i \qquad \textbf{4-4}$$

Where *Ai, EM_i, TM_i, LM_i,* are the electrical temporal and logical masking factors respectively. For alpha particles and terrestrial environments, the drain area is a good estimate of the OFF transistor drain area as has been shown in this work earlier [Limb-12] [Dasg-07]. It has been shown that SETs propagate unatttenuated through logic gates if the delay of the gates is less than the SETs [Mass-08]. For the modern technologies, for example 40 nm, Fo4 delays are in the neighborhood of 15 ps - 20 ps. This delay was much smaller than the SET pulse-widths generated even from alpha particle strikes with 10 fC charge deposition (close to alpha particles). Additionally similar simulations results in earlier parts of this thesis also bear out this fact. Thus the effects of electrical masking can be safely ignored. In other words *EM* can be safely assumed to be unity for these technologies. The temporal masking factor depends on the SET pulse-width, setup-hold time of the flip-flop and the clock period. The temporal masking factor can be estimated as

$$TM = \left\{ \begin{array}{ll} 0 & if\ \ t_{SET} \leq w \\ \dfrac{t_{SET} - w}{T_{clk}} & if\ w < t_{SET} \leq T_{clk} \\ 1 & if\ t_{SET} > T_{clk} \end{array} \right\} \qquad \textbf{4-5}$$

where T_{SET} and w are the SET pulse-width and latching window respectively. The setup and hold time or latching window is the same for all the nodes in the circuit, the only difference being the SET pulse-width. Thus the *TM* of a node for a fixed charge deposition value can be approximated as

$$TM = \left\{ \frac{t_{SET}}{T_{clk}} \right\} \qquad \textbf{4-6}$$

The logical masking factor is the proportion of faults that appear at the output relative to the total number of faults injected at that node. This is a technology independent factor and purely dependent on the Boolean property of circuits. Thus the SER metric used in this work is

$$SER\ metric = \sum_{i=1}^{n\ nodes} A_i \cdot \frac{t_{SET_i}}{T_{clock}} \cdot LM_i \qquad \textbf{4-7}$$

For the simulation framework presented in this thesis, a combination of C# and Verilog was used to partition and synthesize the circuits in different ways. Synopsys Design Compiler is then used to synthesize the circuits to estimate the area, power and speed of the resultant circuits. The FreePDK Faraday 45nm library was used to perform synthesis. The procedure for partitioning and SER and power estimation is divided into two distinct parts as shown in Figure 4-5.

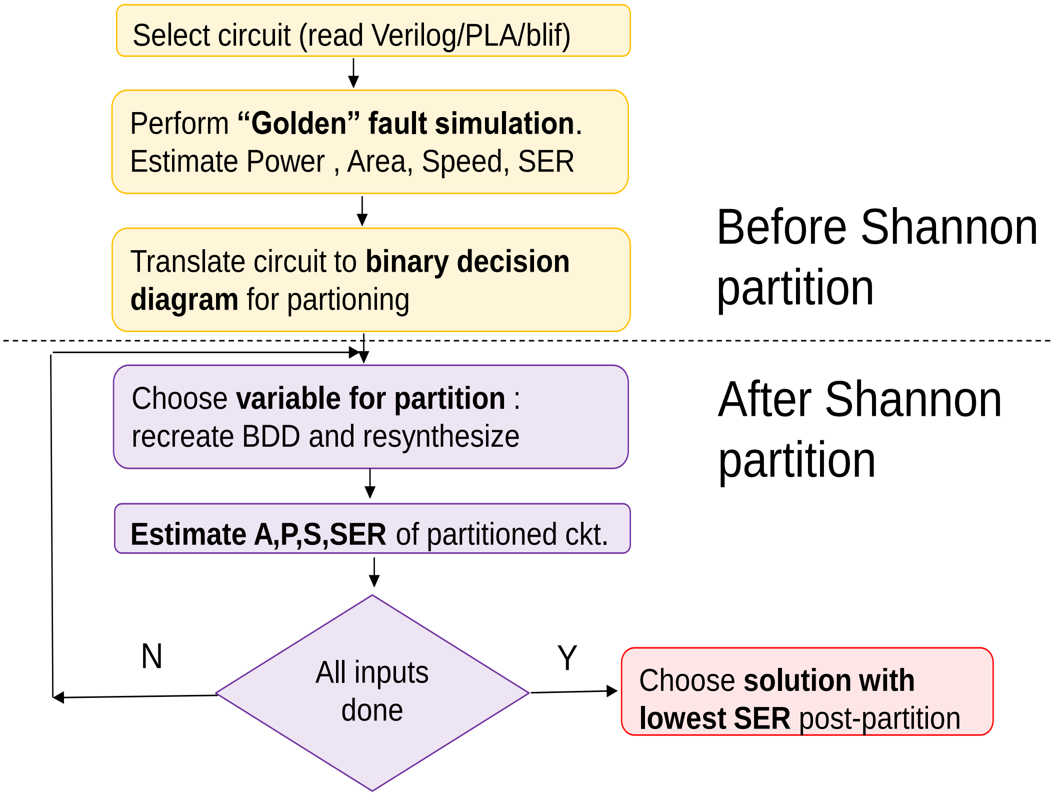

Select circuit (read Verilog/PLA/blif)
Perform "Golden" fault simulation. Estimate Power , Area, Speed, SER
Translate circuit to binary decision diagram for partioning
Before Shannon partition
Choose variable for partition : recreate BDD and resynthesize
Estimate A,P,S,SER of partitioned ckt.
After Shannon partition
N
All inputs done
Y
Choose solution with lowest SER post-partition

Figure 4-5 Flowchart for procedure to synthesize circuits, partition circuits and compare the SER/area/power.

The circuit descriptions are read in either Verilog, programmable logic array (PLA) or Berkeley Logic Interchange Format (BLIF). The circuit is then transformed into Boolean Decision Diagrams (BDDs) for easy graph manipulation and partition. Simultaneously, the circuit is synthesized using the FreePDK Faraday 45nm library using only 2-input NAND, NOR, and NOT gates. The synthesis tool, Synopsys Design Compiler is then used to calculate the area, power and speed of the design. The unconstrained design is provided inputs that have signal switching activity of 50%. In other words, the switching probability of each node is 0.5. The clock is set to 50 MHz so that there are no set-up and hold time violation.

Following this the circuit is partitioned using the variable of choice using the BDD representation. The partitioned circuits are then again synthesized separately. It is important to note that, during post-partition synthesis, the co-factors must be synthesized separately rather than allow the tool to group the co-factors. If this care is not taken, the synthesis tool will resynthesize to produce the original circuit itself due to logical equivalence. The co-factors are then combined using multiplexers for each output. Shared logic that appears in both co-factors is replicated for simplicity. Now, the major advantage of Shannon partition is that the co-factor that is not being actively used can be gated to eliminate switching activity in that co-factor. Gating in this context means using an AND gate (as shown in Figure 4-2) whose one input is the original circuit input and the other input is the variable that is used to perform the circuit partition. The other co-factor is gated in a similar fashion when not required in active computation. The only difference is that in this case the complement of the variable that is used for the partition is used. The partitioning and SER and power analysis was performed exhaustively for certain circuits. The results of partitioning circuits and improvement in power and SER is shown in Figure 4-6. As seen in Figure 4-6, when the circuit is partitioned using several variables, the reduction in power and SER for different variables is very different. In fact the reduction in both can vary from as much as 5% to 55% in case of the cordic circuit shown in Figure 4-6. Three different cases are presented to illustrate three different aspects of such an exercise to partition circuits to achieve a lower SER and power.

Case I: Cordic circuit: In the case of this particular circuit 25 inputs are exhaustively used to partition the circuit. The improvement in power and SER appear to be highly correlated. Intuitively this could be attributed to the fact that power and SER depend on certain variables that affect both. For example, when the number of switching nodes reduces the power consumption decreases. Similarly when certain gates are eliminated the sensitive area reduces. Thus the power and SER are related through the effective reduction in the sensitive area or sensitive nodes. The degree of correlation depends on the number of gates reduced and their individual impact on the power and SER.

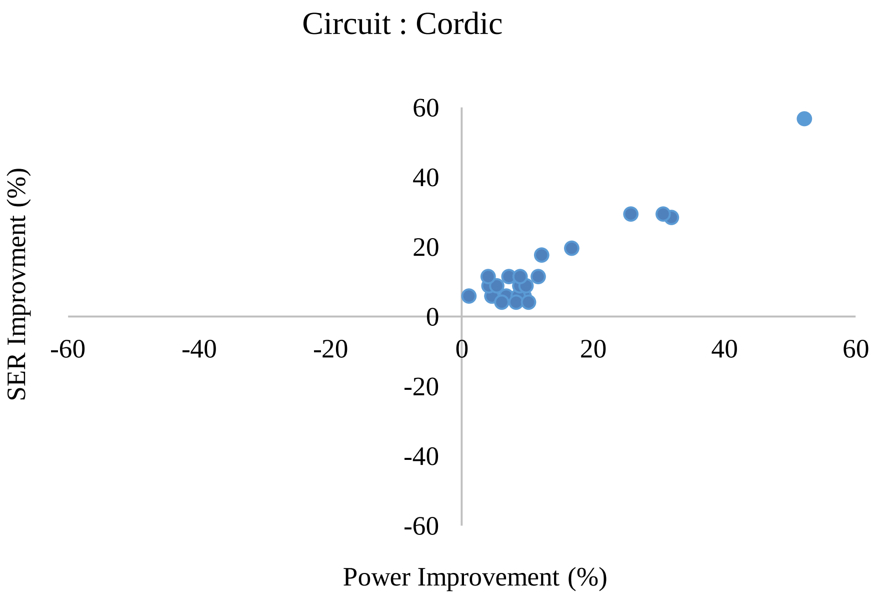


Figure 4-6 Relation between power improvement and SER improvement for different input variables for cordic circuit.

Case II: Parity circuit: This circuit has fewer inputs (8) that in the earlier circuit. However there are several differences between this circuit and the previous case. Firstly, the degree of correlation is not as high as that in the earlier case. The improvement in power, on an average is more than the improvement in SER. The primary reason for this is as follows. When the tool synthesizes the circuit, certain primary inputs to gates are very close to the outputs while others are further away from the outputs. When the inputs are closer to the outputs, the reduction in power can be quite significant [Deva-94]. On the other hand, the reduction in SER is limited. This is illustrated in Figure 4-7. Consider the example in Figure 4-8. When ‘a’ =1 and the input is far away from the output, SETs at nodes 1, 2 and 3 can propagate to the output of the circuit. However, if ‘a’ were close to the output, for example in the position of ‘d’ then SET at previous nodes would be effectively masked. Thus the presence of inputs further away from the outputs leaves several gates that can be potentially struck by ions which will increase the logic SER sensitivity. On the other hand, if the inputs are close to the output post-synthesis, then the reduction in SER may not be significant. Thus not all circuits are alike in terms of the power-SER improvement that can be achieved using Shannon' Theorem. In fact a trade-space exists in the careful synthesis of circuits so that the placement of inputs can be manipulated to allow for greater reduction in power or SER.

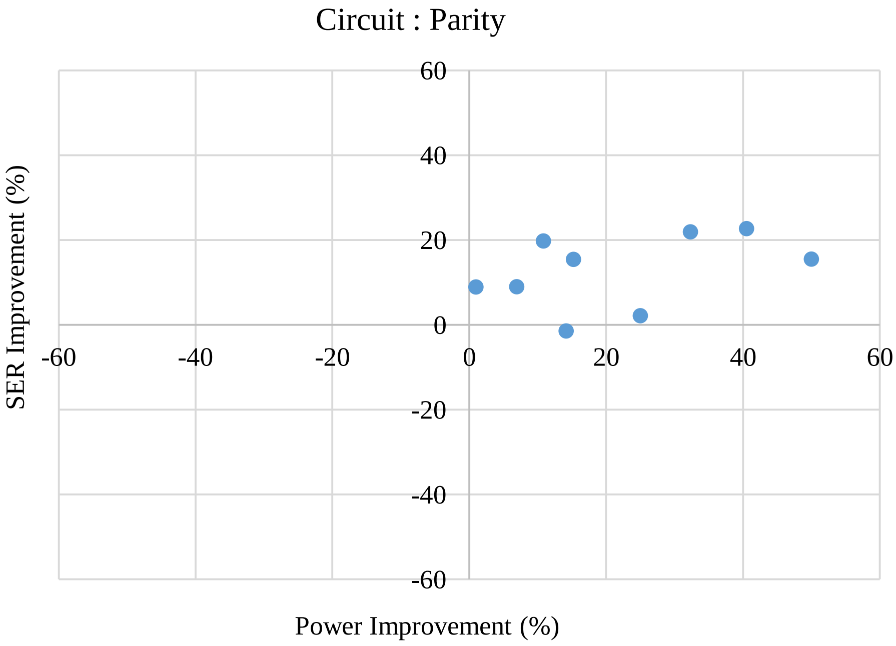


Figure 4-7 Relation between power improvement and SER improvement for different input variables for Parity circuit.

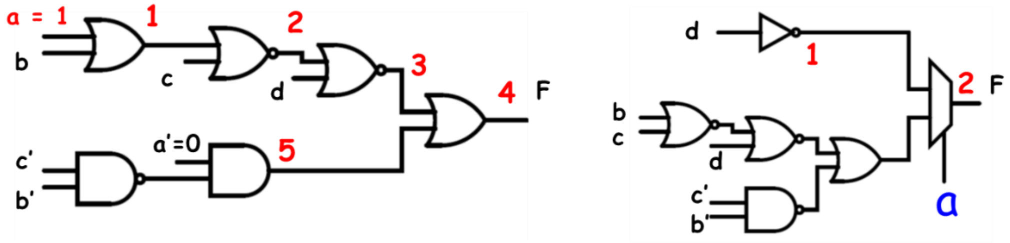


Figure 4-8 Sensitive nodes (in red) for the original circuit and its Shannon equivalent. The number of sensitive nodes reduce from 5 to 2 due to partition.

Case III: ALU circuit: For the ALU, almost all the inputs have the same impact on the power and SER. This is so because the contributions of all the input variables to the functions are similar. Hence partitioning the circuit using on or the other variable has a similar impact on the power and SER reduction. The results of partition for the ALU circuit are shown in Figure 4-9.

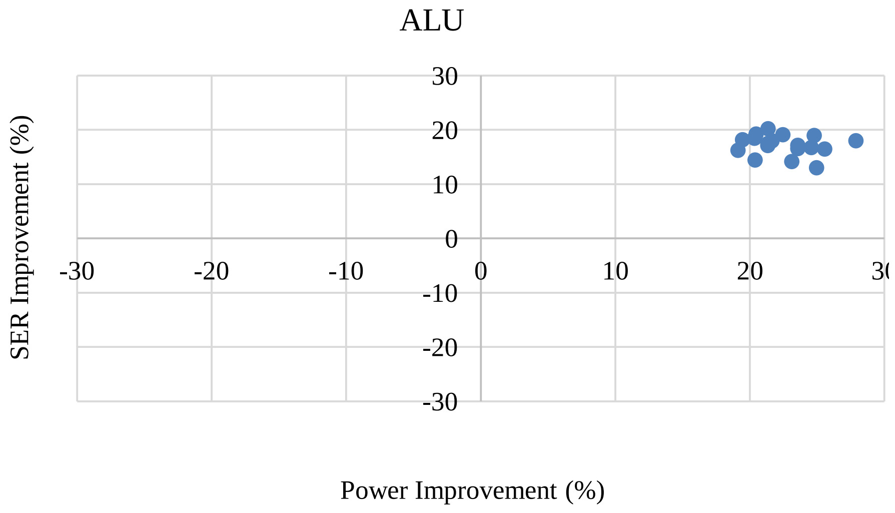


Figure 4-9 Relation between power improvement and SER improvement for different input variables for ALU circuit.

The observations from the above cases can be summarized as follows

1. SER reduction and power reduction can be simultaneously achieved using partition achieved by Shannon's Expansion Theorem.
2. The relationship between these two parameters is mainly through the elimination of unnecessary switching transitions due to logic minimization and reduction in the sensitive area achieved by the same.
3. The amount of reduction depends upon the choice of the variable. Judiciously choosing the right variable can lead to maximum reduction in the power and the SER.

   These observations can be used to guide the process of choosing the best candidate variable to achieve maximum reduction in SER and power.

4.3.1 Heuristic algorithm for control variable selection

In the Shannon expansion scheme the size of the cofactor sub-circuits notably depends on the input variable selected as the control signal of the mux and latches. To avoid an exhaustive search, a heuristic algorithm is proposed which selects the most beneficial control variable from among all the inputs for a given logic function. The heuristics are based on the empirical observation that the size of the co-factor sub-circuits for a selected input is inversely proportional to the number of appearances of the variable in the cubic representation of a given circuit. The technique is summarized as follows. First, the algorithm collapses a given logic function into cubic representation. This is available from Synposys Equation Editor. The number of variables for each cube is calculated to select the variable which most frequently appears in cubes. For the unate variable, the weight of the variable becomes the number of cubes containing the variable. For the binate variable, the proposed algorithm computes the weight of the variable by summing the number of cubes containing the variable and its complement. When performing the Shannon expansion with respect to the selected variable, the cofactors of the selected variable have the minimal set of literals. This produces minimum-sized sub-circuits and leads to the reduction in power dissipation and SER. If more than one variable has the maximum weight, the proposed algorithm figures out the number of literals for each cube containing the variables by calculating the number of variables contained in the cube, then selects the variable with the largest value as the optimal input variable.

Applying the above heuristic, the choice of the variable is indicated in the figures below. For the cordic circuit, the heuristic results in the best variable as far SER reduction is concerned. On the other hand, in case of the Parity circuit, the SER reduction achieved through the use of the heuristic does not achieve the maximum reduction in SER. For all the circuits analyzed in this work the heuristic adopted resulted in a reasonable solution as far as SER reduction is concerned. The chosen variable using the heuristic is highlighted in red in Figure 4-10 and Figure 4-11 for the cordic and parity circuits respectively.

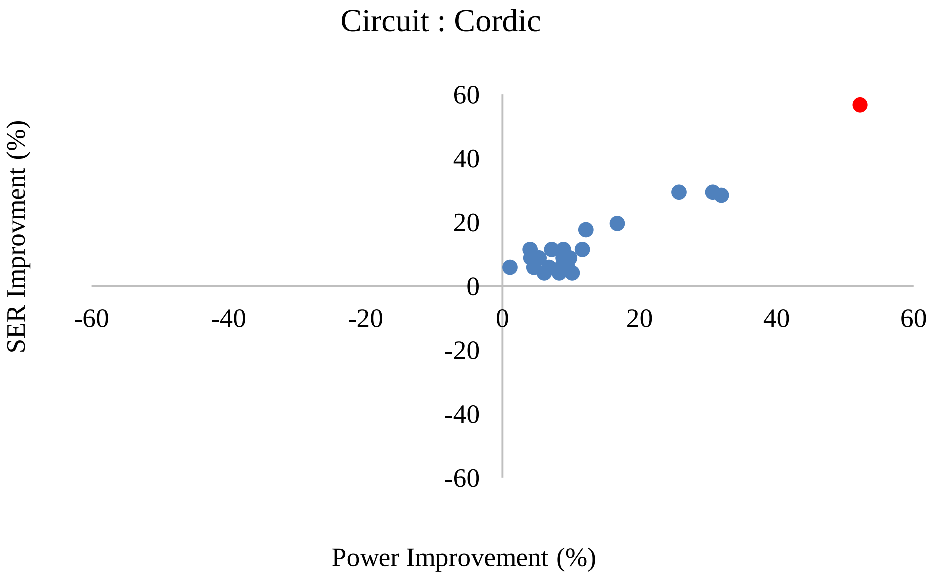


Figure 4-10 Selection of variable (red data point) using heuristic shows that the SER improvement is maximum.

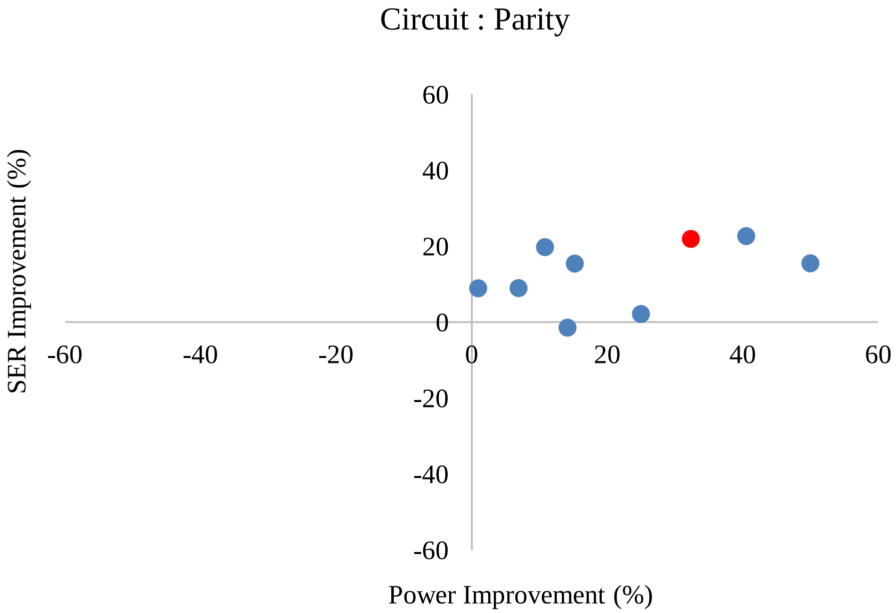


Figure 4-11 Selection of variable (red data point) using heuristic shows that the SER improvement is close to optimal and reasonably good solution results.

4.3.2 Trade-off between Increase in Area and SER reduction

Partitioning the circuits into co-factors results in area overheads. The area of the resulting circuit after partitioning must also be minimized. The different variables chosen for partitioning result in different circuits. The area overhead of these different circuits resulting from partition is plotted in Figure 4-12. In this work, minimizing the power consumption and the SER was the key metric. While doing so however, area overheads are inevitable. For all the circuits analyzed, the area of the partitioned circuits was always higher than the original circuits.

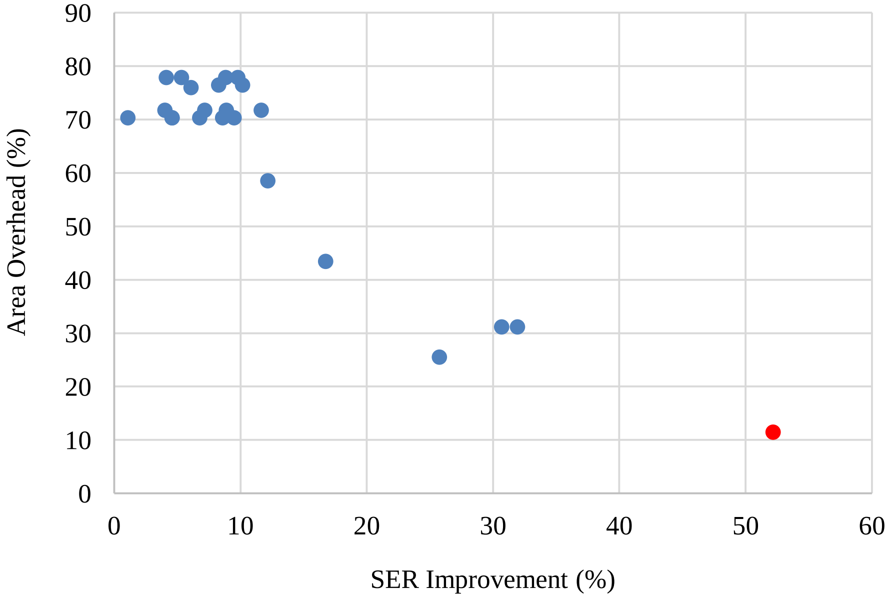


Figure 4-12 Trade-off between area overhead and SER improvement. For all input vectors studied, the area overhead ranged between 10 and 70 %.

4.3.3 Multi-variable Partitioning

Apart from using a single variable to partition circuits, multi-level partition was also studied. At each stage the heuristic was applied to select the best variable for partition. The results of SER reduction using multi-variable reduction is shown in Figure 4-13. The key observation from here is that beyond a certain number of partitions the gains from partitioning circuits actually reduce. This is because the overhead from the introduction of multiplexers eventually leads to an increase in the sensitive area and SETs. Secondly, as the number of partitions increase, significant reduction in the size of the co-factors cannot be achieved. As a result, eventually the gains from partition saturate and even decrease. Similarly the power improvements also begin to decrease as the number of levels of partitioned is increased. Interestingly, the power gains reach a maximum with only 2 levels of partition. This can be attributed to two reasons. As the number of partitions increase the amount of shared logic between co-factors (does not appear in the same cube as the control variable and its complement in the Boolean expression) gets replicated. This adds to the leakage power overhead. Secondly, the additional circuitry also adds switching and leakage power overheads. Thus while in the case of SER mitigation the presence of the inactive co-factor does not affect the SER because SETs in the inactive co-factor are masked, the leakage power from inactive gates quickly increases and offsets gains in switching power reduction. Thus in the approach to reduce *both* power and combinational logic SER, power may be the factor that limits the amount of reduction achievable in the combinational logic SER.

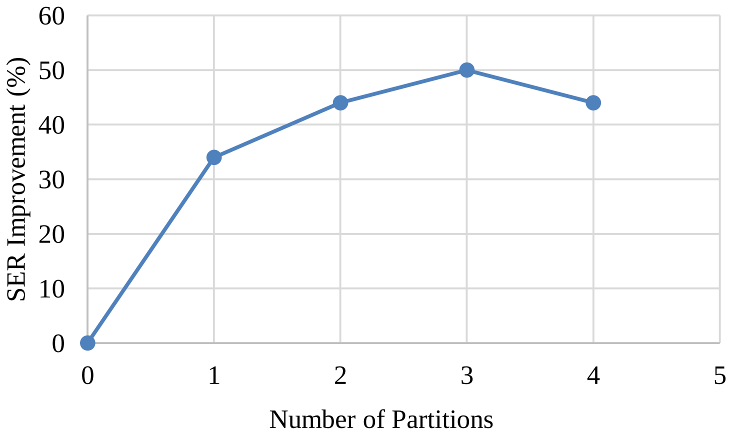


Figure 4-13 Improvement in SER reaches a maximum for 3 levels of partition. 0 partitions corresponds to the original circuit.

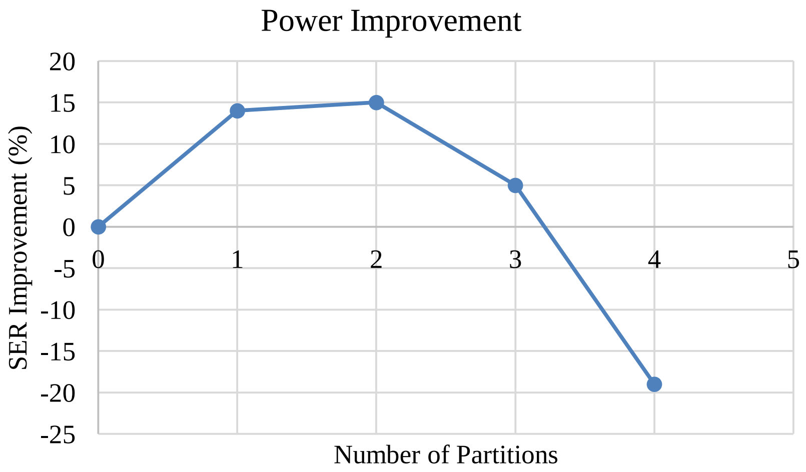


Figure 4-14 Improvement in SER reaches a maximum for 3 levels of partition. 0 partitions corresponds to the original circuit. The power costs from adding additional circuitry and leakage power from the different inactive co-factors limits the improvement in power possible as far as multi-level partition is concerned.

4.4 Summary

Shannon's Expansion Theorem can be effectively used to partition circuits into co-factors. Logic minimization within the individual co-factors can also be achieved. The beauty of this technique is that only one of the two co-factors so produced from single-variable partition, is actively required for computation. Thus the inputs to the co-factor that is not required for computation can be gated or guarded. Thus through logic minimization and gating of co-factors power consumption can be minimized. Logic minimization has the serendipitous benefit of reducing the number of sensitive nodes in the design and thus reducing the soft error sensitivity. Similarly partitioning the circuit into co-factors effectively leads to masking of SETs from the co-factor that is not actively needed for computation. The caveat to performing Shannon Expansion of the circuit is that the right variable must be chosen to partition the circuit. If care is not exercised, then the power consumption and combinational logic SER can also increase. The task of choosing the variable through exhaustive search is intractable. An elegant heuristic that provides near-optimal solutions for power minimization and SER mitigation is presented. The use of the heuristic results in massive reduction in computational effort as well. Results from this work clearly illustrate that achieving low-power consumption along with logic SER minimization is possible. Upto 60 % reduction in logic SER and 40% reduction in dynamic power consumption is possible through the approach presented in this work.

Chapter V. Kernel-Based Shannon Expansion for SER Mitigation

In the previous chapter the application of Shannon's Expansion theorem to improve the combinational logic soft-error sensitivity of arbitrary combinational logic circuits was presented. The primary advantage of implementing circuits using Shannon partitioning for combinational logic soft errors is that the effective number of switching nodes decrease thus resulting in a reduction in power consumption. If the partition of the circuit is performed correctly, the reduction in switching nodes also leads to an effective reduction in the sensitive area resulting in combinational logic SER mitigation. The circuit implementation of Shannon's Expansion using variables to partition circuits leads to multiplexed forms for the Boolean expressions. In this chapter, the possibility of improving the gains from employing Shannon's Expansion by using sub-circuits or sub-functions instead of just variables is used to partition the circuit. The approach relies on identifying parts of the circuit that do not affect the output under certain input conditions. Under these conditions, dynamic power consumption can be reduced in the idle sub-circuits by disabling or preventing signal transitions at the inputs of these idle sub-circuits. By isolating the idle sub-circuits from the part of the circuit that is actively used for computation of the output(s), single-event transients (SETs) in any of the idle sub-circuits can also be prevented from affecting the output(s). Thus soft-error mitigation in combinational logic can be achieved. Secondly if the conditions under which only a small part of the original circuit is used for computation and the rest of the circuit is disabled and isolated, greater reduction in dynamic power consumption and combinational logic soft errors can be achieved. Circuits utilizing this technique were fabricated in a 20-nm bulk CMOS process and exposed to alpha particles. Results clearly show the effectiveness of the proposed design technique in reducing logic soft error rates along with dynamic power consumption. Additionally factors that could lead to a trade-off between power minimization and logic soft error mitigation are also discussed.

5.1 Background: Kernel-Based Shannon Expansion

In this section, Shannon Expansion Theorem is revisited. The concepts of variable-based Shannon Expansion and Kernel-based Shannon expansion are explained.

5.1.1 Background on variable-based and kernel-based Shannon Expansion

A given Boolean expression can be decomposed using Shannon Decomposition Theorem into its co-factors as follows: [Lava-95]

$$f(x_1, x_2, ..x_n) = x_1 \cdot f(1, x_2, ..x_n) + \overline{x_1} \cdot f(0, x_2, ..x_n) \quad \mathbf{5\text{-}1}$$

$$f(x_1, x_2, x_3..x_n) = x_1 \cdot f_{x_1=1} + \overline{x_1} \cdot f_{x_1=0} \quad 5\text{-}2$$

$$f(x_1, x_2, x_3..x_n) = x_1 \cdot CF_1 + \overline{x_1} \cdot CF_2 \quad 5\text{-}3$$

where, x_1, x_2, ..x_n are the input variables. The original function f is evaluated by fixing the value of variable x_1 as 1 to obtain sub-circuit $f\ (1, x_2, x_3, \ ...x_n)$ and x_1 as 0 to obtain sub-circuit $f(0, x_2, x_3, \ ...x_n)$. Formally, $f_{x1=1}$ and $f_{x1=0}$ are the co-factors of x_1. The circuit is usually synthesized with a multiplexer with the variable x_1 as the control input as explained in the previous chapter. Power saving comes from the ability to block nodal transitions in the idle co-factor with the use of either AND gates or transmission gates at the inputs. Similarly, as the effective sensitive area reduces to a single co-factor, combinational logic soft error reduction is also possible.

This theorem can be extended to use a function or combination of more than one variable, instead of a single variable, to partition a given circuit. Instead of a single variable, a *kernel* or sub-function can be used to partition the circuit. In such a case, Equation 1 can be expressed as

$$f(x_1, x_2, x_3..x_n) = K \cdot f_{K=1} + \overline{K} \cdot f_{K=0} \qquad \text{5-4}$$

$$f(x_1, x_2, x_3..x_n) = K \cdot CF_1 + \overline{K} \cdot CF_2 \qquad \text{5-5}$$

Where *K* is the kernel sub-function of *f* and is used to partition the circuit into co-factors. The output can be expressed in the multiplexed form by putting *K=0* and *K=1* in the original expression. The kernel consists of a subset of inputs of the circuit. Such a partition is shown in Figure 5-1 where the multiplexer is controlled by the kernel whose output, *K, is the controlling input for the multiplexer* instead of a single variable. The advantage over the use of a single variable is that in certain cases very small co-factors that are selected more often can be achieved using kernel-based partition compared to variable-based partitions.

Power savings result from being able to block transitions in the co-factors. However, in this case, the size of the kernel must be kept small so that switching activity is minimized. Similarly, the logic cross-section must account for the overhead from the kernel as well. In general, the size of the kernel can be minimized by choosing as few variables as possible that influence the output most often. In this work, the kernel is chosen with the objective to save power for the partitioned circuit and evaluate the effects on the overall soft error rate.

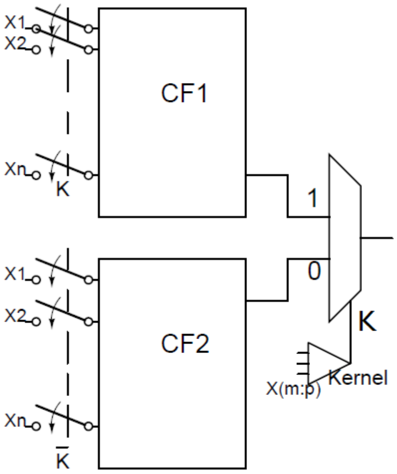


Figure 5-1 Generalized approach to decompose and partition circuits into co-factors using a kernel sub-function or sub-circuit. The kernel consists of a subset of the inputs to the circuit. Power can be saved by gating input transitions to the co-factors depending on the output of the kernel.

5.1.2 Choosing Kernel using Boolean Difference Metric

The right choice of kernel is essential in saving power and achieving reduction in logic soft errors. A technique called pre-computation, originally presented in [Aldi-94] achieves data-dependent power reduction at the sequential logic or combinational logic level using a well-known Boolean algebraic property called the Boolean difference (BD) to determine the appropriate kernel. BD is widely used in Automatic Test Pattern Generation (ATPG) routines for fault detection and sensitization [Tiwa-98], [Sell-68], [Aker-59]. The idea behind the use of Boolean difference is that it can be used to identify those conditions for which the output depends on very few of the total inputs and thus utilize only a few gates to compute the output. The Boolean difference for a function $f(x_1, x_2, x_3..x_n)$ for a variable x_1 can be calculated as

$$BD_{x_i} = f_{x_1=1} \cdot \overline{f_{x_1=0}} + \overline{f_{x_1=1}} \cdot f_{x_1=0} \qquad 5\text{-}6$$

Consider a function

$$f = x_1 x_2 + x_3$$

$$BD_{x_i} = (x_2 + x_3) \cdot \overline{x_3} + \overline{(x_2 + x_3)} \cdot x_3$$
$$BD_{x_i} = x_2 \cdot \overline{x_3}$$

The Boolean difference can also be calculated for multiple variables. A full description of the Boolean difference, its properties and its application in fault tolerance and testability is provided in [Aker-59].

When the Boolean difference evaluates to 1, i.e. if BD_{x1} =1 then any changes in the input variable x_1, can be observed at the output. For example when x_2 = 1 and x_3 = 0, f = $x_1(1)+0$ = x_1. In other words the original function itself is only dependent on input variable x_1. Hence, by building the kernel using the Boolean difference the co-factors that depend on certain variables and those that are independent of certain variables can be produced. Thus very small co-factors that depend on few inputs can potentially be achieved using this metric. If the size of the resultant co-factors is much smaller compared to the original circuit substantial reduction in power can be achieved as explained earlier. If combinational logic soft error reduction is to be achieved through kernel based partitioning, then the size of the kernel and the co-factors must be small. Similarly, if the co-factors are of unequal size then the probability of selecting the smaller of the two co-factors is also important and must be maximized. Thus the kernel-based partition relies on a combination of effective partition and logical masking to reduce the error cross-section in the average case. Any partition that results in a reduction of the total cross-section after partition must satisfy the following cost function in Equation 7, below

$$P_{CF1} \times CS(CF1 + K) + P_{CF2} \times CS(CF2 + K) < CS(original) \qquad 5\text{-}7$$

Where P_{CF1} is the probability of selecting the co-factor C_{F1} and P_{CF2} is the probability of selecting co-factor C_{F2}. $CS(C_{F1}+K)$ and $CS(CF_2+K)$ are error cross-section of the co-factors and kernel respectively. Also $P_{CF1} + P_{CF2} = 1$. Thus for any given input condition, the circuit that is vulnerable to single-event effects is any one of the co-factors and the kernel.

The choice of the Boolean difference as the kernel to disable input transitions is common to this work and [Aldi-94]. However a key distinction between this work and [Aldi-94] is that in this work, the circuit is physically partitioned into two different co-factor circuits so that the effective error cross-section can be reduced for certain input stimuli. In [Aldi-94] the Boolean difference-based kernel is only used to disable signal transitions at input variables that are not needed for output computation for certain input stimuli. The original circuit and its structure is not modified in any way. In such cases no improvement in the combinational logic soft error rate (SER) will be seen.

5.2 Kernel Based Partition For 4-Bit Comparator

Based on the above discussion, Boolean difference based kernel partition was applied to a 4-bit comparator circuit. The comparator produces a logic 1 whenever a 4-bit unsigned binary number $A(A_{3:0}) > B(B_{3:0})$, and 0 otherwise. The comparator was used because its logic size and depth is comparable to modern pipeline circuits [Gunt-08]-[Hris-02]. Secondly the Boolean expression of the comparator is suitable to be partitioned using the approach explained in the previous section. The output expression for a 4-bit comparator is given as

$F = G_3 + G_2E_3 + G_1E_3E_2 + G_0E_3E_2E_1$ 5-8

Where

$G_i = A_i\overline{B_i}$ **5-9**

$E_i = A_iB_i + \overline{A_i}\,\overline{B_i}$ **5-10**

Applying the Boolean difference metric for the two input combination, A_3 and B_3 we get [Aldi-94],

$BD = \overline{A_3}B_3 + \overline{A_3}B_3$ **5-11**

Denoting this as the kernel K,

$K = \overline{A_3}B_3 + \overline{A_3}B_3$ **5-12**

Then the output function of the comparator can be rewritten as

$F = G_3 + G_2\overline{K} + G_1\overline{K}E_2 + G_0\overline{K}E_2E_1$ **5-13**

Because

$E_3 = \overline{K}$

Substituting K=1 and K=0 in Equation 13 we get the co-factors of the decomposition as

$CF_1 = G_3$ **5-14**

$CF_2 = G_3 + G_2 + G_1E_2 + G_0E_2E_1$ **5-15**

When the kernel K evaluates to 1, the term G_3 in co-factor will always be 0 and can thus be dropped from the co-factor in which case, the co-factor CF_2 then reduces to a 3-bit comparator. The omission of G_3 is by observation and may not necessarily apply to all circuits that are partitioned using a kernel. The multiplexed version of the 4-bit comparator using the kernel based on Boolean difference is shown in Figure 5-2. The smaller co-factor consists of a single AND gate (CF_1) while the larger co-factor consists of a 3-bit comparator (CF_2).

In this circuit, when A3≠B3, the output can be completely specified using the most significant bits only and does not depend on other inputs. This is shown in bold in Table 5-1 where A3B3 = 01 or 10.

Table 5-1 Truth Table for co-factor selection using kernel

A_3	B_3	Comparator output	Circuit selected by kernel
0	0	Depends on lower bits (2:0)	*CF2*
0	**1**	**0**	*CF1*
1	**0**	**1**	*CF1*
1	1	Depends on lower bits (2:0)	*CF2*

As the output of the comparator is 1 only when A_3 = 1, the sub-circuit CF_1 is merely the ANDed product of A_3' and B_3. In the other input cases of A_3B_3, the remaining inputs bits are needed to compute the output. If $A_3 = B_3$, the larger co-factor must be used as shown in Figure 5-2.

Thus the Boolean difference metric used to develop the kernel has two advantages. 1) The size of the co-factors produced are smaller than the original circuit and they can be computed using fewer than the total number of inputs 2) The smaller of the two co-factors is selected for 50 % of the input vectors.

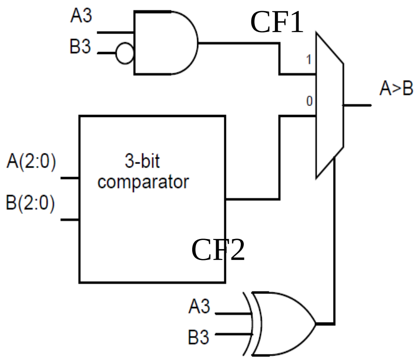


Figure 5-2 Kernel based partition applied to a 4-bit comparator. The smaller co-factor is selected whenever the most significant bits of the comparator A_3 and B_3 are unequal. In other cases the rest of the bits are required for computation and the larger co-factor is selected. The kernel is computed using the Boolean difference metric for inputs A_3 and B_3. Input switches for co-factors are not shown.

Assuming that all inputs have equal probability of being at 0 or 1, for 50% of input conditions for numbers A and B, only a small number of gates (sub-circuit CF_1) is needed to decide whether A>B or not. In the other cases, the output computation requires more gates that are part of the 3-bit comparator. Thus in the average case the logic cross-section is expected to improve. Consider a simple illustrative case for the 4-bit comparator without partition. For those cases where $B_3 > A_3$, the output is always logic low. In the case of the ordinary 4-bit comparator, each of the minterms in Equation 8 will be 0. However the SETs on any of the gates that compute these minterms could produce a transient logic 1 at the output. Thus although the output is 0 and can be easily computed without the need for lower order bits, SETs on the other gates can lead to errors. On other hand, in case of the comparator implemented using kernel based partition, for cases where $B_3 > A_3$, the effective cross-section reduces to very few gates comprising of co-factor CF_1, the kernel and the multiplexer output. If the cross-section of this combination is less than that of the 4-bit comparator cross-section without partition then logic SER reduction can be achieved. In other words the resultant partition must satisfy Equation 7. The power savings for 50% of the input cases will also be significant as only a small number of logic gates switch for 50 % of the cases. In fact, such reduction in SER and power is possible in an arbitrarily large comparator. Note, if 4 variables, A_3, B_3, A_2, B_2 are selected then the output can be computed for 75% of the input vectors. However in this case the size of the co-factor and kernel will be different.

5.3 Test Circuit Description and Experimental Details

The C-CREST approach to measure the combinational logic cross-section has been explained in earlier chapters [Ahlb-08]. It is summarized once again for the convenience of the reader. The data source shown in Figure 5-3 can be used to provide random input patterns as well as static patterns to the circuit. The Circuit-Under-Test (CUT) consists of a shift register design with logic circuits interleaved with flip-flops. The logic circuit is separated from the critical path between flip-flops so that a circuit of arbitrary size or depth can be implemented. Two xor gates are used as control circuits at the output of each logic block. One flip-flop along with the associated logic and control circuit comprises a single stage. The C-CREST design consists of 2,056 of such stages to improve the error statistics. The errors recorded by the error detection circuit can be due to direct strikes on the combinational logic circuits as well as direct strikes on the flip-flops. To separate the errors due to latched SETs from logic and direct strikes on flip-flops, another shift register chain was built with no logic included. This allows errors from both sources to be identified independently. The D flip-flops (DFF) used in all the shift-register chains consisted of NAND-gate based conventional DFF design. All the error detection circuits and counters are protected against single event errors using Triple Modular Redundancy (TMR). The clock is provided by a high-speed Phase Locked Loop (PLL). The PLL was capable of being operated up to 1.2 GHz with low-noise characteristics and was also hardened against upsets.

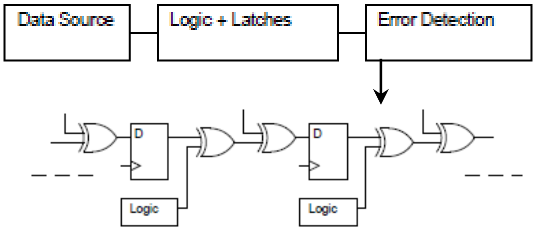


Figure 5-3 C-CREST circuit implemented on 20 nm technology nodes. A single stage consists of a flip-flop and logic block.

Two separate C-CREST circuits were implemented. A baseline 4-bit comparator circuit with no partition was implemented as logic in the first C-CREST shift register chain. The 4-bit comparator forms the block labeled 'Logic' in Figure 5-3. To reduce silicon area requirements, only the smaller co-factor (CF$_1$ in Figure 5-2) of a partitioned 4-bit comparator was implemented as logic in the second C-CREST shift register chain. The average of the cross-sections of the 4-bit comparator and the small co-factor is a reasonable estimate of the effective cross-section of 4-bit comparator with partition. In fact, the 4-bit comparator is slightly bigger than a 3-bit comparator (~1.2X) and would marginally overestimate the cross-section of the 3-bit comparator, so in reality the cross-section of a partitioned circuit would be lower than the mere average of the 4-bit comparator and the smaller co-factor. The control logic consisting of xor gates shown in Figure 5-3 was added to both the logic circuits. Their area and sensitivity is close to that of the xor (for kernel) and multiplexer (selection logic) combination as shown in Figure

5-2. The layout area for peripheral circuits was about 10% of the 4-bit comparator circuit.

The circuits were irradiated with 6 MeV alpha particles from a Polonium-210 source with an activity of 500 µCi, at room temperature. The flux of particles was $7.2\text{x}10^5$ particles/mm^2/s. The alpha source was placed at a distance of less than 1 mm from the dies during testing. The size of the alpha source was 1.4 cm^2, and the maximum die size was 3 mm x 3 mm. To account for both inter-die and experimental variability, measurements were repeated several times at each frequency and logic input value. Besides the high activity allowed thousands of errors to be recorded for each data point. Testing was conducted in accordance with JEDEC specifications [JEDE-06]. The nominal operating voltage was 0.9 V. The test results are reported at a fixed frequency of 416 MHz.

5.4 Experimental Results & Simulations for Power Consumption

In this section, the experimental results from alpha particle testing of the two different circuits is reported. Subsequently the improvement in power is reported for different comparators.

5.4.1 Alpha Particle Logic SER

Figure 5-4 shows the alpha particle logic cross-section for the two circuits for different inputs. The different input conditions for which the 4-bit baseline comparator circuit as well as the smaller co-factor circuit CF_1 were tested are listed in Table 5-2.

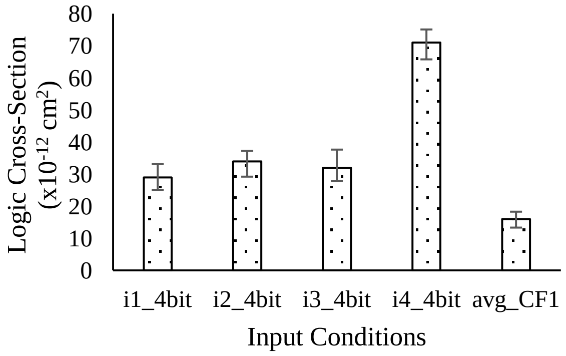


Figure 5-4 The logic cross-section for different input patterns of the 4-bit comparator are consistently higher than that of the small co-factor (CF_1) circuit. When the most significant bit (A_3=B_3) is equal a large number of gates are vulnerable [Input combination i4_4bit]. In other cases, due to logical masking fewer gates are vulnerable. [Input combinations i1_4bit, i2_4bit, i3_4bit].

Table 5-2 Input vectors tested for the 4-bit comparator and smaller co-factor CF1 in Figure 5-2

Input	Value of A, B
i1_CF2	A=1000 B=0000
i2_CF2	A=0100 B=1111
i3_CF2	A=0110 B=1001
i4_CF2	A=0000 B=0001
avg_CF1	01 & 10

As expected, the 4-bit comparator cross-section is consistently higher than that of the smaller co-factor (CF_1 co-factor) for all the input conditions tested. Importantly, however, the cross-section of the comparator is very different for different input conditions. For those conditions in which the most significant bit of inputs A and B is not equal (A_3≠B_3), the cross-section is lower than that of the case where the most significant bits are equal (A_3=B_3). This is mainly because different input conditions lead to different logical masking effects in the comparator itself and not all gates are sensitive to transients. When A_3=$B_{3,}$ (A_3B_3 = 00 in this case) a higher number of gates are sensitive to transients as explained earlier. Thus the average logic cross-section of the comparator needs to be estimated. This was done as follows.

The relative sensitivities in terms of logical masking for the four different input conditions of the comparator were estimated using fault injection. The logical masking factor for each input condition was calculated by injecting faults at each node in the circuit and recording the proportion of faults that propagate to the output. As the circuit was small, this exercise was also repeated exhaustively for all the input conditions and all nodes to estimate the average logical masking factor. As Figure 5-5 shows, different inputs have different masking factors. The results from Figure 5-4 and Figure 5-5 are also in qualitative agreement as far as effects of logical masking on the cross-section are concerned. In fact, i2(A=0100 B=1111) is the condition under which the logical masking is highest (least number of SETs propagate) and i4(A=0000 B=0001) is the condition under which the logical masking is lowest. The average logical masking factor of these two conditions (40%) is close to the average logical masking for the whole circuit (32%). As exhaustive testing over all input conditions is impractical, the cross-section of the 4-bit can be reasonably estimated as the average of the cross-section for these two input conditions. This value is approximately $5x10^{-11}$ cm^2 (average of i2 and i4 in Figure 5-4). On the other hand, in a partitioned circuit, the smaller cross-section would be active for 50 % of the cases and the larger cross-section would be active for the remainder of the 50 % of the cases. Therefore, for a partitioned circuit the cross-section would then be the average of the larger and smaller co-factor. This value is approximately $3.4x10^{-11}$ cm^2 (average of above value and smaller co-factor avg_CF1 from Figure 5-4). This is expressed as the left-hand side of Equation 7. Thus, with the use of kernel based partition, the cross-section reduces by 30 %. Thus the kernel based partition technique to partition circuits is a useful approach to reduce the logic cross-section in certain cases.

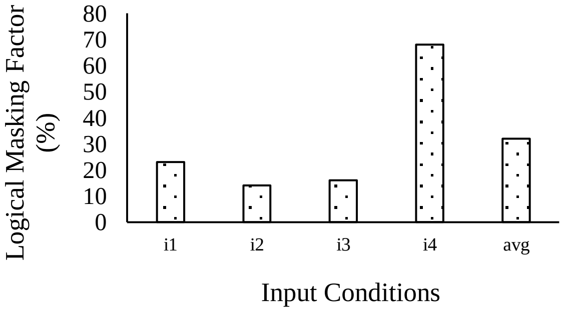


Figure 5-5 The logical masking factor for the 4 input conditions under which the comparator was tested. i4(A=0000 B=0001) is the condition for which maximum transients propagate to the output. i2(A=0100 B=1111) is the condition for which least number of transients propagate to the output, i.e., the logical masking is highest for this input condition.

5.4.2 Simulations for Power Consumption

Simulations to calculate dynamic power consumption were performed using Cadence Spectre 6.1. The simulations were performed at a frequency of 416 MHz and supply voltage of 0.9 V. As shown in Figure 5-6, the total power consumption of a partitioned circuit when compared with the original circuit without partition shows an improvement of only about 3% in the 4-bit comparator case. The primary reason for this is that the overhead due to the introduction of the kernel and gates at the input to disable transitions add to dynamic and leakage power overhead. The power improvement however steadily increases as the size of the comparator is increased as shown in Figure 5-6. In fact for an N-bit comparator where N is large, the improvement in logic SER mitigation and power consumption will approach 50%.

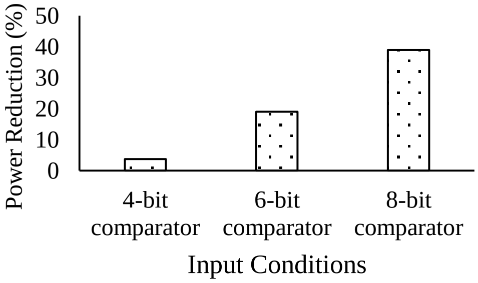


Figure 5-6 The power consumption for 4- 6- and 8-bit comparators was estimated. For smaller sized comparators (4-bit for example), the additional overhead due to the kernel and input switches adds to dynamic and leakage power overhead. As the size of the comparator increases, the improvement in power is quite significant.

5.5 Summary

In this work, a powerful technique to mitigate combinational logic soft errors along with power minimization through the use of circuit partition is illustrated. The technique for partition has its basis in a low-power design philosophy which isolates and disables idle sub-circuits from active ones as frequently as possible to save power. This can be serendipitously applied for soft error mitigation as well because isolating the idle sub-circuits from the active ones minimizes the likelihood of SETs from the isolated sub-circuits affecting the output. This idea was applied to a 4-bit comparator wherein 30 % decrease in logic cross-section was observed. For comparator circuits, as the size of the comparator grows, large savings in SER and power are possible. While this work shows that kernel-based partition can be applied to reduce *both* the power and soft error cross-section of certain circuits, it must be emphasized that the results may not be applicable to every combinational logic circuit. The impact of some key factors must be evaluated while partitioning any circuit to achieve lower power and/or combinational logic SER mitigation: 1) The choice of the kernel affects the size of the co-factors. Different kernels will produce very different co-factors. In general the problem of choosing the best kernel is NP-complete and heuristics may be adopted for kernel selection [Choi-02]. 2) The size of the co-factors and kernel must be small compared to the original circuit. If this is not achieved, the combinational logic SER could increase at the expense of power reduction. In fact there could also be a trade-off between power reduction and logic SER reduction based on the choice of kernel used for partition, inputs selected for the kernel etc.

Chapter VI. Circuit-Level Implementation of Shannon Expansion for SER Mitigation

In the previous chapters the application of Shannon's Expansion theorem to improve the combinational logic soft-error sensitivity of arbitrary combinational logic circuits was presented. The primary advantage of implementing circuits using Shannon partitioning for combinational logic soft errors is that the effective number of switching nodes decrease thus resulting in a reduction in power consumption. The simple variable based expansion can be extended to kernel or function based partition to generate very small co-factors. The circuit implementation of Shannon's Expansion leads to multiplexed forms for the Boolean expressions. Such multiplexed forms can be implemented using different circuit-level logic families to explore the possibility of improving the gains from employing Shannon's Expansion. Additionally the kernel-based partition idea is also implemented to evaluate the maximum possible gains from Shannon decomposition at the circuit level.

The observation that Shannon's Expansion relies on the use of repeated multiplexed forms of Boolean expressions is used as the basis for the use of transmission-gate logic to implement circuits. Transmission gates are often used in high-speed circuits to replace large area-hungry standard cell CMOS multiplexers. This chapter is organized as follows: A brief summary is provided to Shannon Expansion theorem for the benefit of the reader. The implementation of Shannon's theorem through the use of different logic styles is explained. This is followed by description and discussion of experimental results that demonstrate how this technique can be used to reduce the SER of adder circuits implemented in TSMC 20 nm node. The delay, area and power of the different circuits is analyzed and the pros and cons of designing circuits using the Shannon technique are discussed.

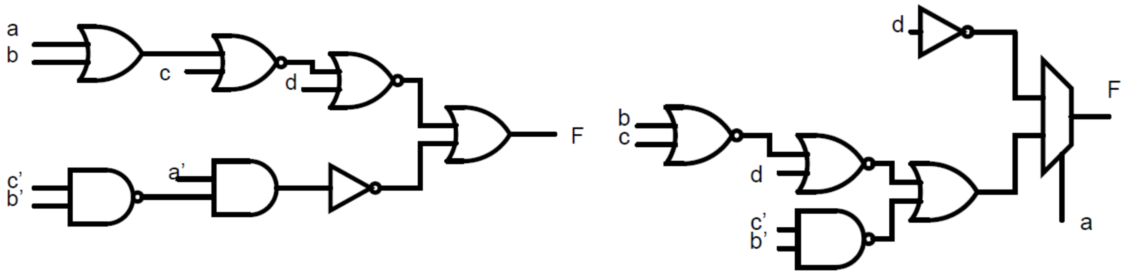
6.1 Background: Shannon Expansion Theorem

Shannon Expansion Theorem is based on George Boole or Claude Shannon's original theory on switching circuits. It can be used for logic synthesis and optimization [Lava-95]. Arbitrary Boolean expressions can be expressed as follows.

$$f(x_1, x_2, x_3, ...x_n) = x_1 \cdot f(1, x_2, x_3, ...x_n) + \overline{x_1} \cdot f(0, x_2, x_3, ...x_n)$$

$$f(x_1, x_2, x_3, ...x_n) = x_1 \cdot CF_1 + \overline{x_1} \cdot CF_2$$

where $CF_1 = f(1, x_2, x_3..x_n)$, $CF_2 = f(0, x_2, x_3..x_n)$, are the co-factors obtained by evaluating the original expression by setting $x_1 = 1$ and $x_1 = 0$ respectively. Here x_1 is called the control variable or partitioning variable. Depending on the logic value of the control variable, the appropriate co-factor is selected to compute the output. For any input condition of the circuit, only one of the two co-factors is required to compute the output. Thus, such an expression can be structurally translated into a multiplexed form where the two inputs of the multiplexer are the cofactors and the multiplexer select line is control variable selected for Shannon expansion or partition. Shannon expansion can be repeatedly applied within each co-factor to obtain multiplexed forms of the co-factors themselves. Figure 6-1 illustrates how Shannon expansion can be used to partition circuits. The individual co-factors can be obtained by setting a = 1 and a = 0, a being the control variable in this case. The original circuit and the Shannon equivalent are shown in Figure 1 (a) and Figure 1 (b) respectively.



(b) **(b)**

Figure 6-1 Original circuit (a) and its Shannon Equivalent (b) implemented with multiplexers.

6.2 Circuit implementation of Shannon's Expansion Theorem

Shannon's partition inherently allows the use of multiplexers to implement Boolean functions. In fact, Shannon expansion can be successively applied to partition the co-factors themselves using every variable in the circuit. In other words, the whole circuit can be decomposed using a network of *only* multiplexers. Such an implementation strategy under ordinary circumstances would lead to a huge increase in the area of circuits. However this multiplexed form is ideal for certain functions. The implementation of multiplexed functions is most commonly performed using transmission gates as fast multiplexers in high-performance processors [Cell-05].

In this thesis, Shannon expansion was practically implemented using transmission gate structures. The key advantage of transmission gates is the smaller capacitance which allows faster implementations of certain function. The lower capacitance and compact implementation also translates into lower dynamic power consumption. The primary disadvantage of transmission gates is that the design must be carried out carefully to avoid floating nodes that can lead to erroneous operation. More importantly, it is difficult to determine the delay of circuit paths that implement transmission gates especially in series because transmission gates are nothing but an R-C low pass filter. Implementing such networks in series results in a complex transcendental equation to determine the delay. The complexity of calculating the delay increases as more transmission gates are added and the number of poles in the system increase. On the other hand, the delay of standard CMOS gates can be accurately calculated using well known concepts like logical effort [West-09]. From the point of view of single-event effects, there are two important considerations. The smaller transmission gates mean that the sensitive area potentially reduces. On the other hand, the smaller capacitance on the gates and lower drive means the critical charge required for single-event transient generation is lower. Thus with the implementation of certain circuits with Shannon's expansion theorem using transmission gates, there are potential trade-offs between the power and area improvement and the single-event sensitivity. This is the first instance of systematic comparison of the various factors involved in designing circuits using transmission gates implemented using Shannon's Theorem and testing their single-event sensitivity. In the following sections comparisons are made between a standard cell adder and a "Shannon Adder" implemented using transmission gates. The experimental set-up for alpha particle and heavy-ion experiments are explained. Subsequently, simulations are used to understand the implications of using transmission gates with Shannon's expansion theorem are explored.

6.2.1 Circuit Description

The choice of circuit for test purposes in this work was critical. The ideal circuit to compare the use of transmission gate logic compared to standard CMOS implementation must meet two conditions. 1) The circuit designed using two different techniques (Shannon with transmission gates and standard CMOS) must still be representative of commonly used circuits in microprocessors etc. Then the results of comparison can be extended to other circuits for the SER mitigation and power reduction benefits. 2) The circuits must be simple enough to be experimentally tested exhaustively yet easily so as to compare the experimental results with simulated fault injection campaigns. The adder circuit was chosen for this exercise for its universal application in a wide range of circuits. Besides, at its most basic form the full adder has only 3 inputs and only 8 different input combinations. Thus testing such a circuit exhaustively would not be challenging. Two different flavors of the full adder circuit were designed and tested in 20 nm bulk technology. The standard cell CMOS adder consisted of And-Or-Inverter (AOI) gates. The sum and carry functions are shown in Figure 6-2 and Figure 6-3 respectively. The XOR gates consisted on 4 NAND gates each for the sum function. All the gates were sized to achieve equal rise and fall time. The minimum sized inverter against which the rise and fall delays were benchmarked had dimensions of W_P/W_N = 220 nm/180 nm.

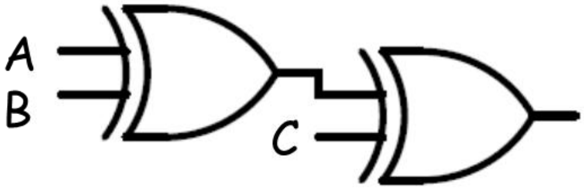


Figure 6-2 Sum generation circuit of the Standard cell adder. Internally the circuit consisted of NAND gates.

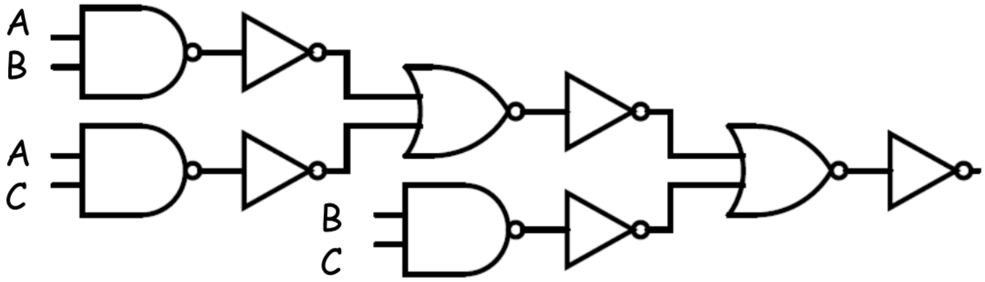


Figure 6-3 Carry generation circuit of the Standard cell adder. Internally the circuit consisted of NAND gates.

The "Shannon Adder" consisted of transmission gate and the size of all the transistors used in each of the transmission gate designs was W_P/W_N = 220 nm/180 nm. The implementation of the Shannon Adder is shown in Figure 3. From the circuit shown below, it is clear that a combination of transmission gates and inverters is used. The primary purpose of the inverters was to restore the nodal values to full rail signal, in case of any degradation of the signals when they pass through the transmission gates. The transmission gates themselves implement multiplexer functions. For example the sum output of an adder is implemented using just 4 transistors. The sum expression for the adder is

$$Sum = A \otimes B \otimes C$$

This can be decomposed using Shannon's theorem as follows:

$$Sum = \overline{A}B + A\overline{B}$$

Here the controlling or partitioning variable is B and A and A' are the co-factors resulting from the partition. Thus the XOR function lends itself conveniently to be implemented using Shannon Expansion. Similarly the carry of the adder is expressed as

$$Carry = A \cdot B + B \cdot C + A \cdot C$$

In such a form, Shannon Expansion would not be very convenient. Hence the expression is transformed into a form shown below. This is a very important transformation and effectively results in a kernel now becoming the controlling function. In this case the sum function generated earlier as the XOR of inputs A and B is used as the kernel to partition the carry function. Thus transmission gate implementation is coupled with the kernel-based partitioning approach to achieve a very small and compact carry implementation.

$$Carry = (A \otimes B) \cdot C + (\overline{A \otimes B}) \cdot B = S \cdot C + \overline{S} \cdot B$$

Here, the expression is now transformed into a form that resembles the variable-co-factor notation of Shannon's Theorem. Here the control variable is now S (sum) and the co-factors are merely the inputs C and B respectively. Thus expanding the expression using Shannon's Theorem allows a very compact implementation using transmission gates. Also note that the sum expression is used a control variable for the carry expression. This this a form of multi-variable expansion of Shannon's Theorem. The area and transistor count of the two circuits are summarized in Table 6-1.

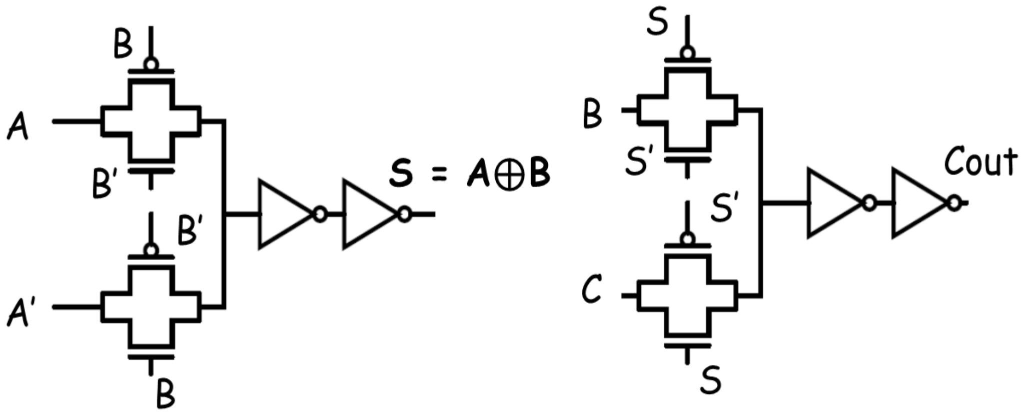


Figure 6-5 Shannon implementation of the sum (S) and carry outputs using transmission gates.

Table 6-1 Comparison of Standard CMOS and Shannon Adder

Circuit	Number of Transistors	Drawn Area	Min W_P/W_N
Std. CMOS Adder	28	0.525 μm^2	220nm/180nm
Transmission Gate Adder	16	0.240 μm^2	220nm/180nm

6.3 Test Circuit and Experimental Details

As has been described in Chapter 3, the C-CREST approach was used to measure the combinational logic cross-section of the two different adder circuits tested. Two different combinational logic circuits were implemented using the C-CREST scheme. The logic in the first C-CREST circuit consisted of the standard cell adder. The logic in the second C-CREST circuit consisted of the transmission gate adder. The circuits were irradiated with 6 MeV alpha particles from a Polonium-210 source with an activity of

500 µCi, at room temperature. The circuits were tested at different frequencies and voltages at room temperature. As in all the experimental test procedures and set-ups, JEDEC testing standards were followed and statistical variability was limited by collected hundreds of errors in multiple experimental runs.

6.4 Alpha-Particle Experimental Results

The test circuits were tested under a variety of frequency and voltage conditions. The frequency dependence of logic soft errors is well-known and observed in both cases shown below for an operating voltage of 0.95 V. As seen in Figure 6-6, the cross-section of the Shannon Adder is lower by about 35%. The Shannon implementation did provide improvement in terms of measured cross-section. The primary reason for this is that due to the smaller size in terms of area, the sensitive area of the Shannon transmission gate adder is lower for almost every input combination.

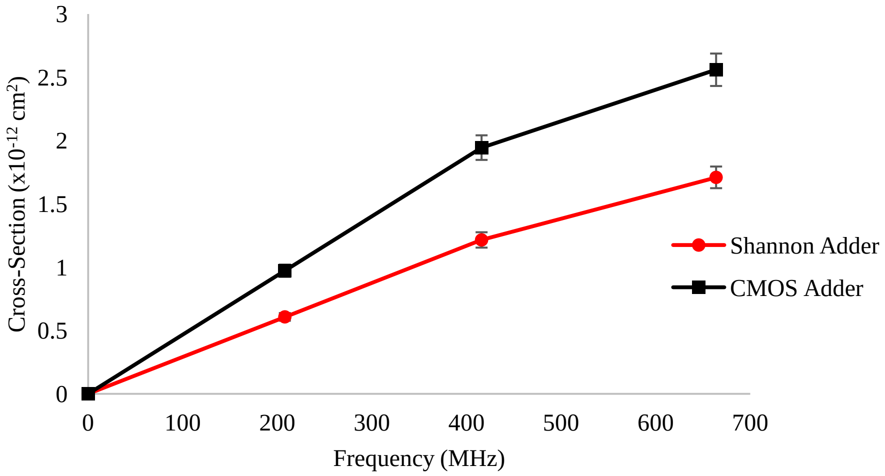


Figure 6-6 Comparison of the alpha particle logic SER of the conventional CMOS and Shannon implementation shows that the Shannon implementation improves the logic SER by ~35 %.

However the improvement is not as much as the difference in area (Table 6-1) of the two circuits would suggest. The area of the Standard cell adder is about 2.3X higher than that of the Shannon transmission gate adder. However the difference in cross-section for the most sensitive input is merely 35 %. The SET pulse-widths were investigated to calculate their impact on the logic cross-section. It is well known that differences in drive current and capacitances introduces differences in pulse-widths for different gates. Different gates and corresponding input conditions have different SET pulse-width distributions. The SET pulse-width of three representative gates used in the circuits described above was estimated for different values of charge deposition using the Bias-Dependent current pulse model [Kaup-09]. It is seen that the SET pulse-widths of the transmission gate adder are higher than those of inverters and the NAND gates. In fact the SET pulse-width of the transmission gate adder is about 3 times as much as the inverter in spite of the fact that their physical area is exactly the same. The pulse-widths for the different gates is shown in Figure 6-7.

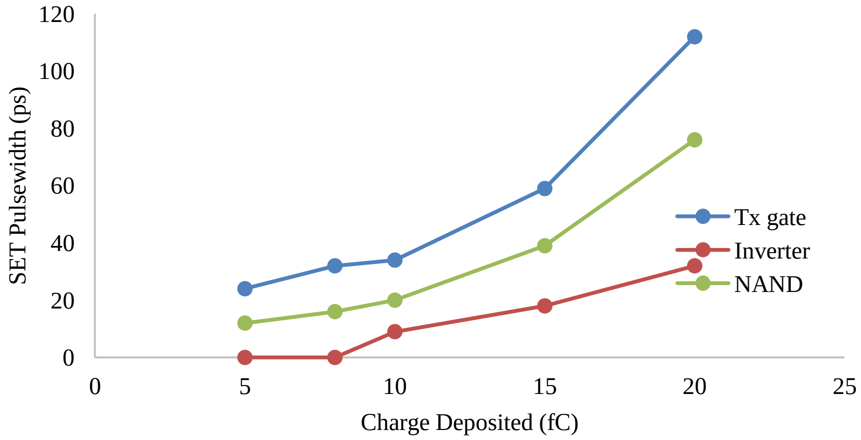


Figure 6-7 SET pulse-widths for various gates and structures for charge deposited of 10 fC.

The reason for this apparent difference in the pulse-widths can be explained as follows. Consider the case of an inverter where the PMOS transistor is ON and the NMOS transistor is OFF. If the NMOS transistor is OFF then the PMOS transistor restores the node to the original value. The pulse-width is indeed inversely proportional to the restoring drive. In fact the SET pulse-width is proportional to the RC delay time taken by the PMOS transistor to restore the node to its original value. In the case of the inverter shown in Figure 6-8, the RC delay = R_pC. On the other hand consider the case of the transmission gate and inverter combination. As Figure 6-5 shows every transmission gate output node is preceded by an inverter to restore any degradation in signal swing. Consider the case where the NMOS transistor part of the transmission gate is struck. The NMOS and PMOS transistors of this transmission gate are ON and PMOS transistor of the previous stage is ON. In this case the restoring drive is provided by the ON PMOS transistor. This current flows through the PMOS (part of the inverter) and the parallel combination of resistances of the PMOS and NMOS of the transmission gate. This is shown in Figure 6-9. Thus the time taken to restore the node can be calculated using the Elmore delay model

$$\tau = R_p C_1 + (R_p \ + (R_p \parallel R_n))C_2$$

Setting Rp=Rn = R which is a reasonable approximation (matched drive from PMOS and NMOS) and C1 = 2C and C2 = C, (transistor sizes almost the same) we get.

$$\tau = \frac{7}{2} RC$$

Thus the RC delay or the time to restore the node (SET pulse-width duration) of the transmission gate case is about 3.5 time that of the inverter, mainly because of the introduction of additional resistive and capacitive paths for the PMOS restoring current. As a result the pulse-widths of the transmission gates are longer than those of the inverter or even the NAND gate.

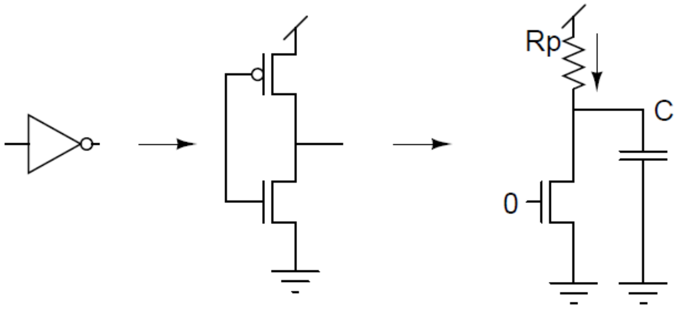


Figure 6-8 Inverter with ON PMOS transistor. The restoring drive is responsible for nodal capacitance being restored to original value.

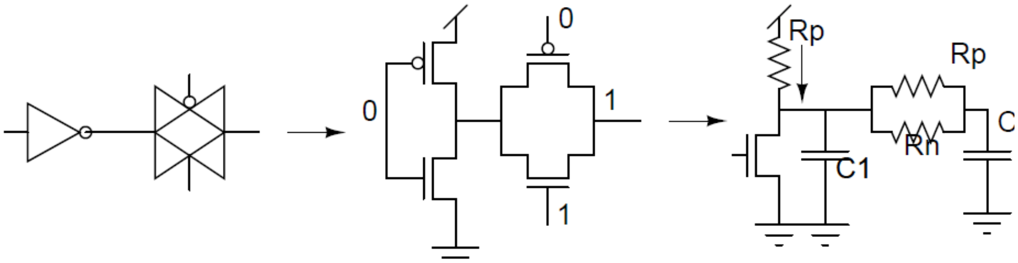


Figure 6-9 Transmission gate + inverter combination where the NMOS of the transmission gate is struck. The restoring drive encounters additional resistive and capacitive paths which increases the restoring delay or SET pulse-width.

These results emphasize the fact that the combined effect of sensitive area and pulse-widths must be accounted for while considering the use of transmission gates. Although the sensitive area clearly reduces, the pulse-widths. In order to gauge the impact of *both* the sensitive area and the SET pulse-width, ensemble Monte-Carlo simulations were performed in which every node of the standard cell adder and every node of the Shannon Adder were struck with a fixed charge deposition of 10 fC using the Bias-dependent model [Kaup-09]. The product of the sensitive area and pulse-width was monitored for each node and the cumulative sum of product of area and pulse-width was recorded for the whole circuit. This is plotted in Figure 6-10. The difference in the product of the sensitive area and the SET pulse-width is about 25 %, which is close to the 35 % difference in logic cross-section observed experimentally. Clearly this points to the fact that although the physical area of the circuit is lower with the use of transmission gates, the SET pulse-widths of the nodes in the transmission gate adder could be higher than those from the CMOS adder. As a result the overall logic cross-section depends on the relative differences between these two factors and must be considered very carefully while employing transmission gate based design.

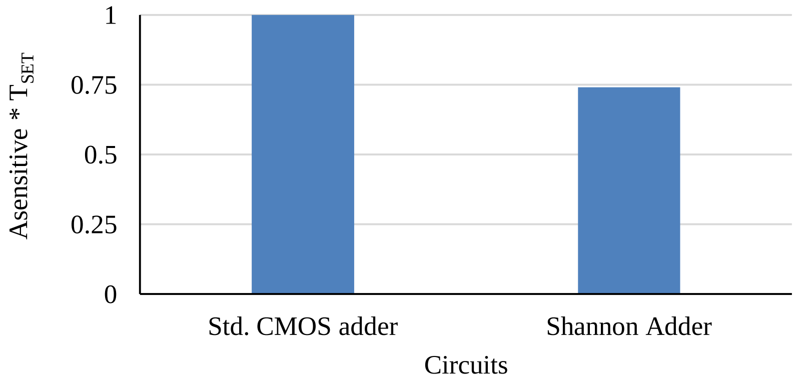


Figure 6-10 Sum of product of sensitive area and SET pulse-width with strikes simulated at each node of the two circuits (charge deposited = 10 fC).

In summary the Shannon implementation of adders using transmission gates is a powerful way to reduce the logic SER. The Shannon implementation importantly results in only one of the two co-factors being active for all the inputs and also ensures that there are restoring transistors for every node in the circuit. Besides, the sensitive area reduces substantially. However, careful attention must be paid to the SET pulse-width distribution of the individual nodes where transmission gates are used compared to standard CMOS gate implementation. If the restoring current is weak then the pulse-widths can be very long. As a rule of thumb, cascading transmission gates is not recommend because of the poor restoring strength introduced by the additional RC combinations. Additionally the noise issues due to signal swing degradation and charge sharing may arise if every node is not carefully protected with CMOS restoring devices like inverters. But the forced inclusion of these inverters also offsets some of the area benefits of a pure transmission gate implementation as well. Next, the power and delay improvement of Shannon implementation is considered.

6.4.1 Delay and Power Analysis

A. Adder delays.

Transmission gate implementations are traditionally faster than standard CMOS gates due to the lower capacitances and fewer circuit nodes. This is borne out in the delay analysis of the two adders simulated at different voltages and plotted in Figure 6-11.

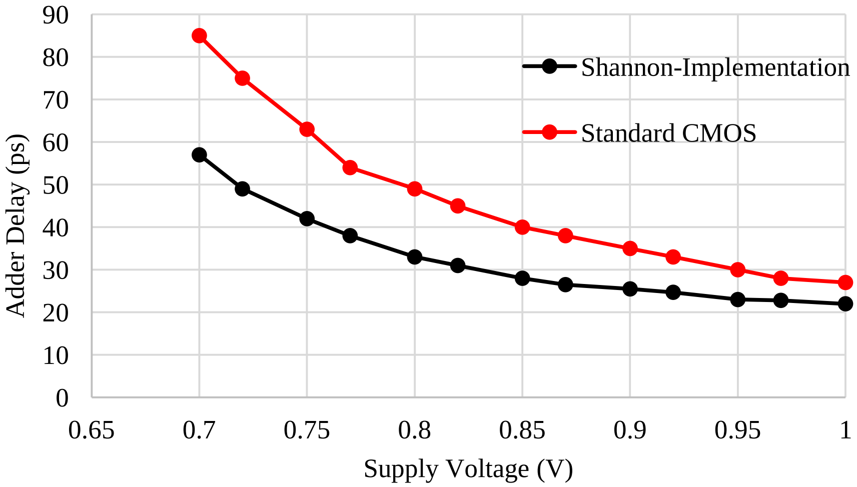


Figure 6-11 Worst case adder delays measured for the standard CMOS and Shannon implementation

B. Dynamic Power Consumption and Power-Delay Product of the Adders

The power-delay product is a useful metric to compare two different logic families or two circuits built using two different logic families. The dynamic power was calculated by applying 1000 random patterns at a frequency of 1 GHz to both the adders at nominal voltage (0.9). The worst-case delay was measured as in the previous section. The resultant power delay product is plotted in Figure 6-12. The power consumption and the delay of the Shannon Adder was much lower than that of the standard cell adder. As a result the power delay product too, is much lower. In fact, a minimum is seen at a voltage of about 0.85 V which indicates that this is the lowest energy point of the Shannon Adder.

The logic cross-section for the Shannon Adder at this voltage and 416 MHz is ~ 1.41x10^{-11} cm^{2} as shown in Figure 6-13. At the same time, the minimum energy point for the standard cell adder appears to be ~0.98 V. At this operating voltage and frequency of 416 MHz, the logic cross-section of the standard cell adder is ~ 1.32x10^{-11} cm^{2} which is comparable to the logic cross-section of the Shannon Adder. Thus the two cross-sections are comparable at their minimum energy points. This is a very interesting result and needs to be investigated. However thorough investigation this is beyond the scope of this work.

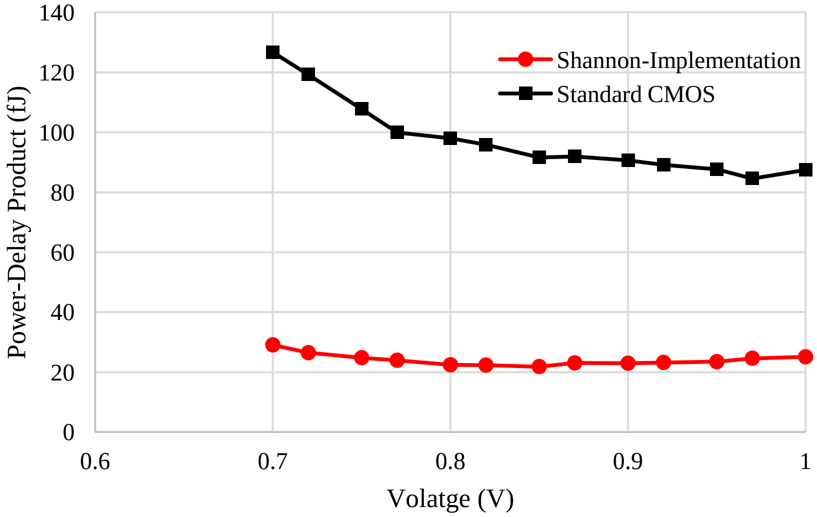


Figure 6-12 : Power-delay product of the Shannon Adder is much lower compared to the standard CMOS implementation

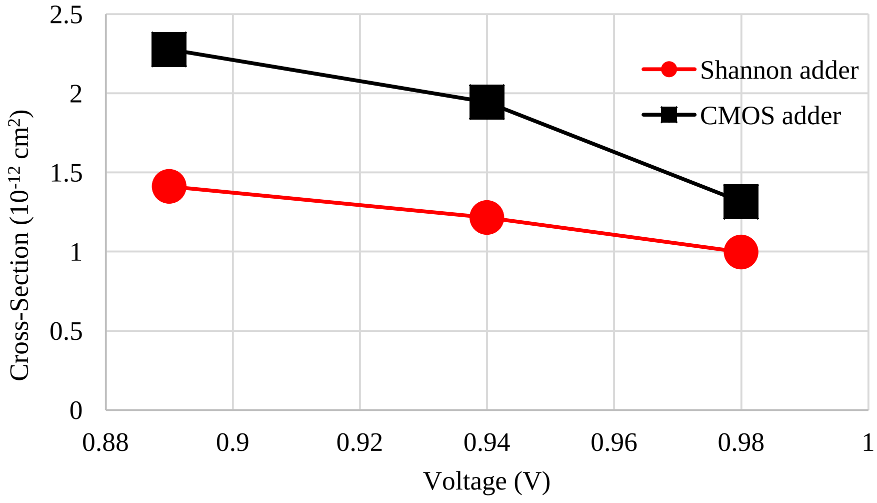


Figure 6-13 Logic SER as a function of voltage for the two implementation schemes.

6.5 Summary

Shannon's theorem can be conveniently implemented using transmission gates given their suitability to build multiplexed functions. The key advantages of adopting a transmission gate approach is that of faster speed, lower power consumption and lower logic SER. The expected reduction in sensitive area from the use of transmission gates with Shannon's theorem, however is not realized because the transient pulse-widths due to strikes on transmission gates are much longer compared to standard CMOS gates. If transmission gates are cascaded in series, the SET pulse-widths will increase because of the reduced current drive to restore the struck node. These key pitfalls must be evaluated before adopting a transmission gate style to reduce power and SER. For the circuits tested in this work, Shannon implementation using transmission gates offers a powerful alternative to standard CMOS implementation in terms of faster speed, less power and area along with higher soft error reliability.

Chapter VII. SER Mitigation Using Low-Power Pipelining

In previous chapters, the impact of different gate and circuit level approaches to achieve power-aware combinational logic SER mitigation have been presented. In this section a powerful approach at a higher level of abstraction to achieve SER mitigation along with power minimization is discussed. In the design of VLSI circuits, the tools available to architectural designers or software developers are different from those available to circuit designers or device engineers. Modern-day designs rely heavily on the use of silicon-intellectual property (IP) available in the form of portable high-level RTL designs that individual designers can synthesize in different ways. On many occasions however, even synthesis may not be feasible option. Architectural designers who work with such IP blocks can rarely modify the underlying architecture such as gate sizes, circuit design etc. of the IP, but must instead design the system most effectively by utilizing the different 'block-level' designs in the IP library. In such cases architectural or micro-architectural changes such as pipelining, retiming blocks, parallelizing computation, and smart placement of sub-system blocks can help improve the performance, reduce power and minimize area.

In this chapter, an approach to reducing the SER by modifying a low-power pipelining technique is first proposed by [Hris-02] is studied. Results suggest that modest amount of pipelining can help reduce the SER as well as the power consumption of the evaluated circuits. Pipelined ripple carry adders are used as a pathfinder to understand the different parameters like combinational logic masking factors, effects of voltage on SET pulse-widths, frequency, flip-flop design and count that influence the total SER of pipelined systems. This chapter is organized as follows : A brief introduction to basics of traditional pipelining are provided. The traditional pipelining approach which is focused towards increasing performance or throughput is contrasted with a low-power approach to pipelining which emphasizes constant throughput but low power consumption. Following this the SER for a low-power pipeline is evaluated and the impact of hardening the combinational logic on SER and power is discussed.

7.1 Introduction: Traditional Pipelining Basics

Pipelining is a well-known and well-researched concept in which the throughput or performance of microprocessors is improved. Microprocessors execute a given program instruction by instruction. Each instruction which is introduced is first fetched, decoded and then executed. This process is repeated for all the instructions. Consider a processor that does not implement pipelining and processes every instruction in one clock cycle. All activity in the processor occurs on the clock edge and is timed by the clock. This time duration is called T-state, machine state or clock period. Let us assume that an instruction requires one T-state for fetching and executing an instruction. Let the time required to do be = T. Thus a single instruction requires T units of time to be completed. Let there be 5 such instructions in a program. Then the program sequence is as shown in Table 7-1:

Table 7-1 : Program execution flow for non-pipelined processor

T		T		T		T		T	
F1	E1	F2	E2	F3	E3	F4	E4	F5	F5
1st instruction		2nd instruction		3rd instruction		4th instruction		5th instruction	

As a result the total time taken for the program = 5 instructions * T units = 5T. Thus the rate of processing instructions called the throughput or instructions per cycle (IPC) = 5/5T = 1/T

Now consider a pipelined implementation of this processor. In a pipelined implementation, the logic is divided into smaller parts each of which is active at the same time. In other words, the action of fetching and executing instructions is separated and performed by two separate functional units so that while one instruction is being executed, another instruction can be fetched. The advantage with such a technique is that the logic has been divided into smaller parts, so the clock period can be reduced. Let us assume an ideal value of T/2 for the new clock period in this case. The program sequence is now given in Table 7-2.

Table 7-2 Program execution flow for pipelined processor

T/2	T/2	T/2	T/2	T/2	T/2
F1	E1				
	F2	E2			
		F3	E3		
			F4	E4	
				F5	E5

As seen in the program sequence for the pipelined processor, the total time taken reduces to 6*T/2 = 3T. The reason for this is that instruction fetch and execution overlap and the different processor units for fetch and execute are active at the same time. Thus pipelining resembles an assembly line. The time taken can be expressed as 1*(T/2 + T/2) + 4*T/2 = 3T.

For a general processor where the degree of pipelining is k, where k is the number of pipeline stages, then the time to execute a program with N instructions would be kT + (N-k)T/k where T/k is the new clock period for the processor. If N >>k which is true for almost all realistic workloads and programs, the time taken is ~ NT/k. The time a non-pipelined processor with clock period T would take is NT. Thus the speedup possible in the ideal case is almost k. In other words the throughput improves by a factor of k for a pipelined system.

7.1.1 Pipeline Limitations and Impact on Power and SER

Pipelining improves the throughput of processors but there are certain factors that limit the improvement in throughput to less than the ideal and also certain unavoidable overheads. The limitations of pipelining are evident when branch instructions occur or instruction hazards occur. Branch instructions require execution to begin from a new address and loading of new instructions, which means that the pipeline must be flushed and filled again with new instructions. If branch instruction occur very frequently, the throughput of the processor may be affected severely due to repeated emptying of the pipeline. Hazards occur when certain instructions require successive pipeline blocks to share data or information about the instruction being executed. Thus certain execution units must be given more time to complete the task and provide the data safely to the next execution block. This requires the pipeline to be stalled. These stalls can add to the performance overhead reducing the overall throughput.

Apart from these factors, the inevitable cost of pipelining is an increase in dynamic power consumption due to higher frequency of operation. Additionally the number of flip-flops and the clock network and routing complexity increases. All these factors contribute to increased power consumption. As far as soft-errors are concerned, increasing the clock speed results in an increase in the combinational logic SER. Increasing the number of flip-flops also increases the flip-flop error rate. Thus, increased amount of pipelining results in a higher SER overall for the chip.

7.2 Pipelining for Low-Power

Beginning in the 90’s and into the early 2000’s, pipelining accompanied by increasing frequency was the primary method by which processor performance was improved. The trend towards deeper pipelines in microprocessors is seen in the development of Intel x86 family, with a factor of 7 reduction in logic depth per stage over the last decade [Hris-02]. In the early-2000’s, the gains from pipelining were quite prominent and processor frequencies and pipeline depths continued to increase. But longer pipelines lead to large penalties from hazards and stalls. The resulting reduction in instructions completed per cycle (IPC) reduces the performance advantage from greater clock frequency, with greater impact on codes with lower instruction-level parallelism (ILP). Early work by [Kunk-86] considered pipelining in vector supercomputers and found that 8–10 gate levels was performance-optimal for scalar code, and 4 gate levels for more parallel vector code. Several authors have investigated the performance-optimal pipeline depth for superscalar microprocessors [Hart-02, Hris-02, Spra-02], with a consensus in the range of 8–11 Fo4 delays for SPEC integer codes and around 6 FO4 delays for SPEC floating-point codes, which generally have higher ILP. These performance-optimal numbers ignore power as well as the design and verification complexity that would accompany such high-frequency designs (roughly twice the clock rate of existing systems [Spra-02]). Similarly the power costs and power density costs of pipelining limits the amount of pipelining that can be introduced. With emphasis in digital processor design changing towards low-power consumption the work by [Chan-92] proposed to use pipelining as a low-power tool but no attempt was made to determine the power-optimal pipelining strategy. Power-optimal pipelining was first explored in

[Heo-04]. The approach in [Heo-04] and related work is summarized here. The authors emphasized *fixed-throughput designs* for highly parallel computations and so do not include any performance loss from an increased frequency of pipeline stalls as pipeline depths increase. This is extremely relevant in the case of architectures like digital signal processing (DSP) pipelines and arithmetic intensive pipelines employed in graphics processing units (GPUs). DSP architectures and GPUs use a lot of adders, multipliers and other arithmetic constructs. Unlike microprocessors which involve heavy branching, DSP architectures do not need to branch very often. On the other hand, large data streams need to be processed and computed. Thus data parallelization and pipelining can be achieved very easily. In fact, just as instructions can be computed in pipeline fashion, data too can be processed and computed in pipelined fashion. For example 8-bit addition of two 8-bit numbers can be performed by adding the lower nibbles (lower 4 bits) together and then adding the result to the higher nibbles (higher 4 bits). Thus the operation of adding two 8-bits can be serialized into two separate 4-bit components. In traditional pipelining, the clock frequency would be increased to compensate for the reduced delay for 4-bit addition as against 8-bit addition. However this is accompanied with power overheads. However, if *fixed-throughput low-power operation* is required, then the frequency of operation does not need to be increased with pipelining. In such cases the amount of logic per stage reduces. As a result, the logic now gains more time to complete its operation. Consider the case of the 8-bit adder divided into 4-bits discussed earlier. If the frequency is 2 GHz for the 8-bit adder and then the 8-bit adder can be divided into two 4-bit components, the logic depth per stage reduces. However the time required to complete the 4-bit addition operation remains unchanged if the voltage is the same. But, now since the 4-bit adder has gained considerable slack to complete the ADD operation on 4-bits, the supply voltage for both the adder blocks can be lowered so that power can be saved. As the logic amount per pipeline stage decreases, the voltage can be scaled even more. Consider the case of low-power pipeline instruction flow. Shown in Table 7-3. The clock period is same as the original: T units. Pipelining allows simultaneous execution but the latency of the first instruction increases.

Table 7-3 Program execution flow for fixed throughput pipelined processor

T	T	T	T	T	T
F1	E1				
	F2	E2			
		F3	E3		
			F4	E4	
				F5	E5

As seen in Table above, the first instruction requires 2 cycles to complete after which each instruction can be overlapped and completed in 1 clock cycle. Thus the total time taken in 6T. Here, the latency for the first instruction is larger (2 cycles) than in the non-pipelined case (latency=1 cycle for non-pipelined processor). For a general pipeline where the frequency is not increased and there are no stalls/hazards, the throughput would be (k)T + (N-1)T = (N+k-1)T. Again if N >>k-1, then the throughput is ~NT. Therefore no major loss improvement in throughput is obtained from such a technique. The key point here is that the supply voltage can now be lowered for each execution block that is part of the pipeline. The delay condition required to be met to find the lowest voltage that can be used for the logic units is as follows:

$$T_{logic}(Vdd) + T_{flip-flop}(Vdd) = \frac{T_{logic}(V)}{N} + T_{flip-flop}(V) \qquad \textbf{7-1}$$

where, $T_{logic}(Vdd)$ is the combinational logic delay for the entire logic chain of the the non-pipelined system and $T_{flip\text{-}flop}(Vdd)$ is the flip-flop delay for non-pipelined system operating at nominal supply voltage *Vdd*. If the logic is divided into multiple units as in a pipelined system, then the voltage can be lowered in each stage. The resulting delay in each stage and for each flip-flop is $T_{logic}(V)$ and $T_{flip\text{-}flop}(V)$ respectively and N is the number of stages that the pipeline is divided into (pipeline depth). The difference between traditional pipelining and pipelining for low-power is illustrated in Figure 7-1 and Figure 7-2.

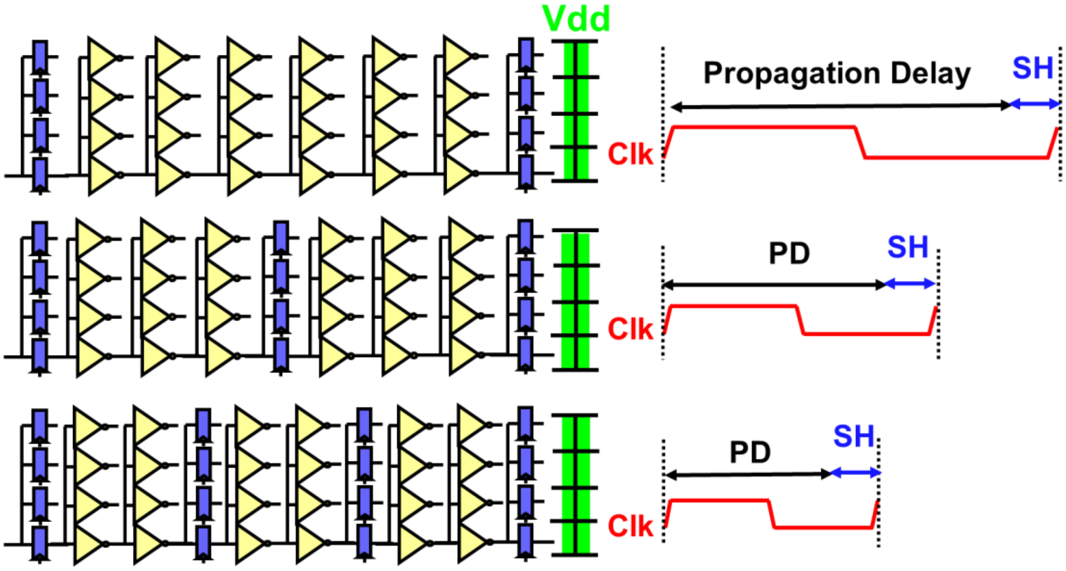


Figure 7-1 Traditional pipelining where the logic stages are divided into smaller sub-units and the frequency is increased (clock period decreases). The voltage is maintained at nominal supply level.

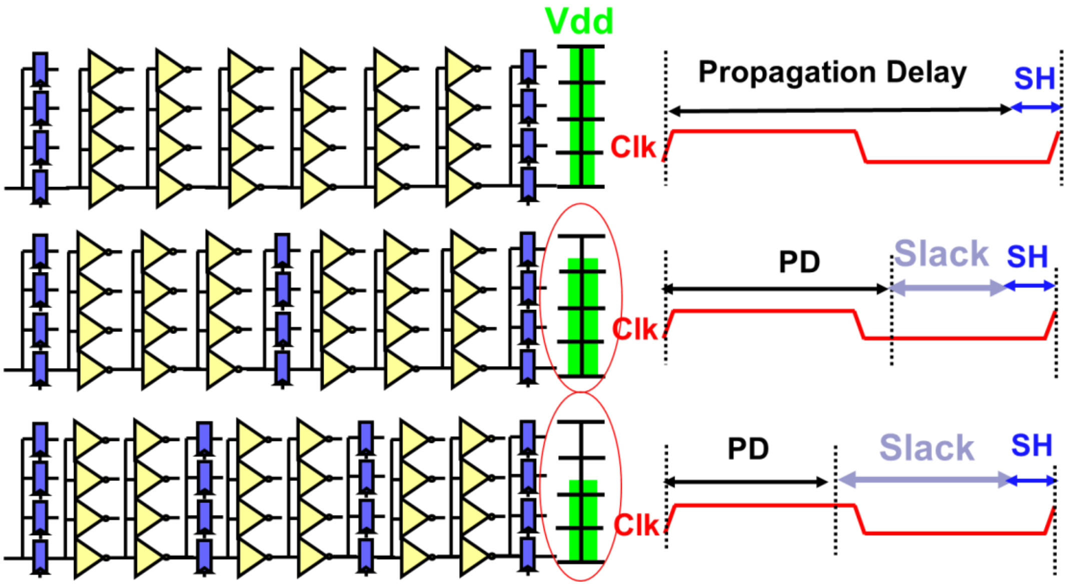


Figure 7-2 Pipelining for low power where the logic is divided into sub-units but the frequency is kept constant and voltage is lowered for each logic block.

7.3 Pipelined Structure Evaluated

As discussed earlier, pipelining can be used to increase performance and throughput as well to save power consumption with constant throughput. The objective in this work is to mitigate combinational logic with low power operation and minimal performance overheads. By implication the low-power pipelining approach is adopted to achieve low-power and SER. The target system for such approaches are DSP pipelines, arithmetic pipelines and ASIC computational circuits that do not need to branch very often. A 24-bit adder ripple carry adder as a testbed to understand how low-power as well as lower-SER can be achieved through pipelining. The adder was used for the following purposes:

1. Adders are universally used as fundamental building blocks in arithmetic and DSP pipelines.

2. The carry signal represents a pipeline signal that is propagated from one stage to another.
3. The logical masking characteristics of the adder could be exhaustively analyzed.
4. Experimental results to compare the logic SER two different adders as a function of voltage and frequency were available. This experimental data could be used in the models to estimate the improvement in SER and power with pipelining.

The 24-bit adder structure with different levels of pipelining is shown in Figure 7-3. The baseline structure has no pipelining and the 24-bit ripple carry adder forms the logic terminated by a flip-flop. As the pipeline depth (N) is increased the adder is decomposed into smaller logic blocks at each successive stage. For example N = 2, corresponds to two 12-bit adders forming the logic component and the number of flip-flops increases by a factor of 2. N=8 corresponds to eight 3-bit adders forming the logic interspersed with 8 flip-flops. There are some key things to remember about such a pipeline

1. A deeper pipeline corresponds to less logic between stages
2. The total amount of logic remains the same but the number of flip-flops grows with pipelining
3. The voltage at each stage can be lowered in accordance with Equation 1.

   Again, in the 24-bit structure only the carry-bit and carry generation circuit is utilized. Each adder also has sum generation which needs separate gates but the sum outputs are independent of each other and pipelining does not change the number of flip-flops or change the amount of logic for the sum generation. Thus the sum circuits are not relevant for this discussion.

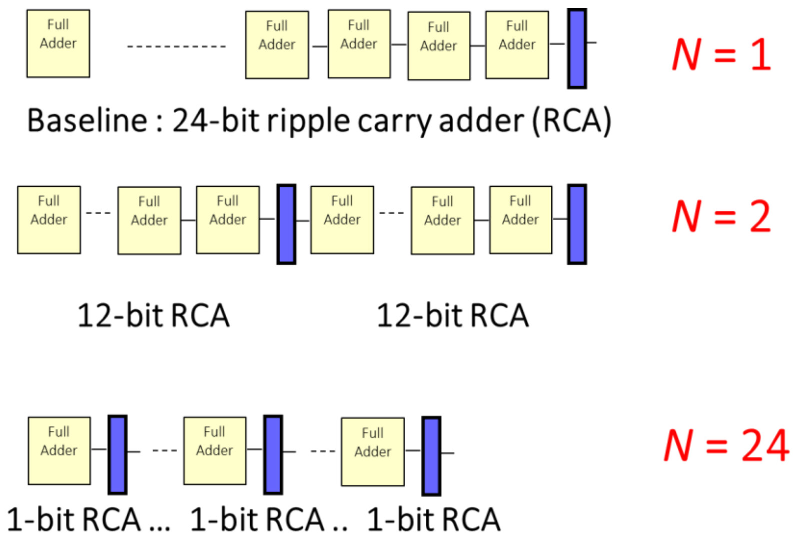


Figure 7-3 24-bit adder structure for power reduction and SER mitigation. The baseline is a non-pipelined version terminated by a flip-flop. Pipeline depth of 2 corresponds to 2 12-bit adders separated by a flip-flop. The extreme case is a pipeline depth of 24 with only one adder stage per flip-flop.

In this work, all the power calculations were performed in simulations and through the use of models. The frequency for these simulations was set to 2 GHz and a 20 nm PDK was used for simulation purposes. The SER estimation on the other hand was mixed approach where the impact of voltage on two different adders (standard cell and Shannon adder) was performed experimentally. The exhaustive logical masking calculations were performed using simulations. Both these factors were then used to model the total SER of the pipelined adder block. The adders used to experimentally evaluate the logic SER as function of voltage and frequency were implemented in a 20 nm bulk technology. The frequency of operation for the experiments was 416 MHz and voltage was varied from 1.1 V to 0.85 V. The carry circuit consisted of 5 gates (2 AND gates, 2 OR gates). The minimum size of the inverter used was Wp/Wn = 220 nm/180 nm. The size of the other gates was adjusted to achieve equal rise and fall time as that of the minimum sized inverter used. Thus the baseline consisted of 24 adders (each consisting of 5 gates) and 1 standard NAND gate D-flip-flop to receive the carry output. The delay of a single carry stage was 18.4 ps at a nominal voltage of 0.9 V. Thus the total delay through the 24 stages was 432 ps. In other words the logic delay was 432 ps for the worst case path where the carry ripples through the 24-bits. Thus, the clock frequency in simulation was therefore set to safe 2 GHz (clock period 500 ps) to ensure that the sum of the logic delay and the setup-hold time was less than 500 ps.

The following discussion in this chapter is organized as follows: The reduction in power by increasing pipeline depth for the 24-bit RCA is introduced. The total SER of such a pipeline is modeled using experimental data for SER as a function of voltage and differences in masking factors. Following this, the implications of hardening the individual adders in different ways for SER and power are presented.

7.4 Power Consumption with Pipelining

The objective in low-power pipelining is to keep the frequency the same and increase the pipeline depth. As the frequency is the same, the voltage in each logic block can be lowered. Using Equation 1, given below as well, the minimum possible voltage at which each pipeline logic block can be operated is given as

$$T_{logic}(Vdd) + T_{flip-flop}(Vdd) = \frac{T_{logic}(V)}{N} + T_{flip-flop}(V)$$

In this work, the voltage of the flip-flops is not changed because a logic dominated pipeline is assumed and the reduction in power consumption due to lower voltage on the logic is much more than that due to reduction in power from the flip-flops. Thus the above equation can be expressed as

$$T_{logic}(Vdd) + T_{flip-flop}(Vdd) = \frac{T_{logic}(V)}{N} + T_{flip-flop}(Vdd) \qquad \textbf{7-2}$$

The resultant voltages obtained using the above equation for different pipeline depths N is plotted in Figure 7-4. As the pipeline depth is increased, lower voltages can be used. The initial reduction in voltage is quite sharp but the reduction in voltage is not as significant beyond a pipeline depth of about 8. This indicates that even modest pipelining allows for significant reduction in voltage. This reduction in voltage directly impacts the power consumption as function of pipeline depth plotted in Figure 7-5. The power consumption for the 24-bit pipelined system can be modeled as

$$\begin{aligned} P_{total}(N) &= P_{switching}(N) + P_{leakage}(N) \\ P_{total}(N) &= P_{switching\,logic} + N^{\rho} P_{switching\,flip-flop} + P_{leak-logic} + N^{\rho} P_{leak\,flip-flop} \end{aligned} \qquad \textbf{7-3}$$

The total power was recorded in simulation for a frequency of 2 GHz and voltage V for different pipeline depths N. The switching components of logic and flip-flops were recorded separately. Ordinarily, pipelining would result in a linear increase in the number of flip-flops. However a more pessimistic approach has been adopted where the leakage and switching components are scaled by a flip-flop growth factor ρ = 1.2. So the number of flip-flops in the pipelined systems grow at the rate of $N^{1.2}$ rather than N^{1} (linear) [Heo-04]. The additional flip-flops are often required for synchronization related tasks with pipelining. The power consumption scaled up to a flip-flop growth factor of $N^{1.2}$ is plotted in Figure 7-5.

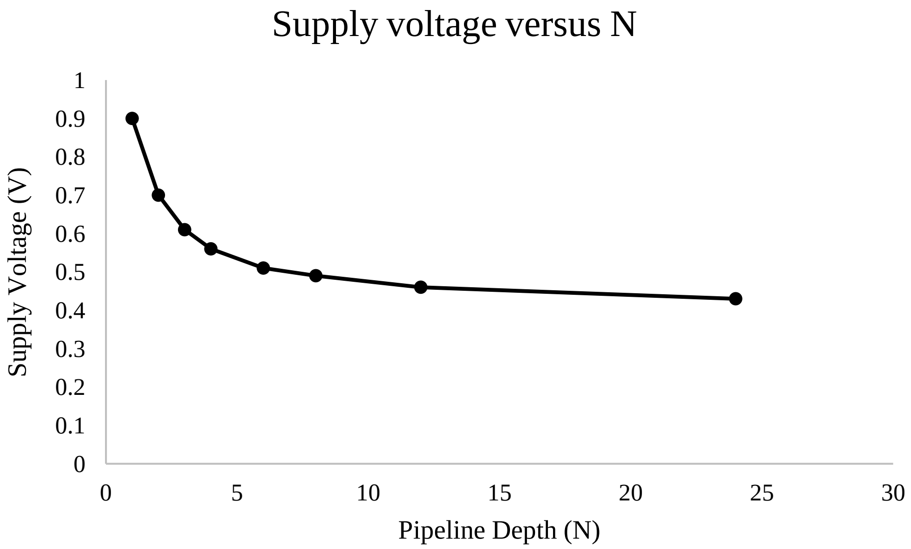


Figure 7-4 Supply voltage variation as function of pipeline depth. A lower voltage can be used for deeper pipelines but, the gains decrease beyond a pipeline depth of about 8.

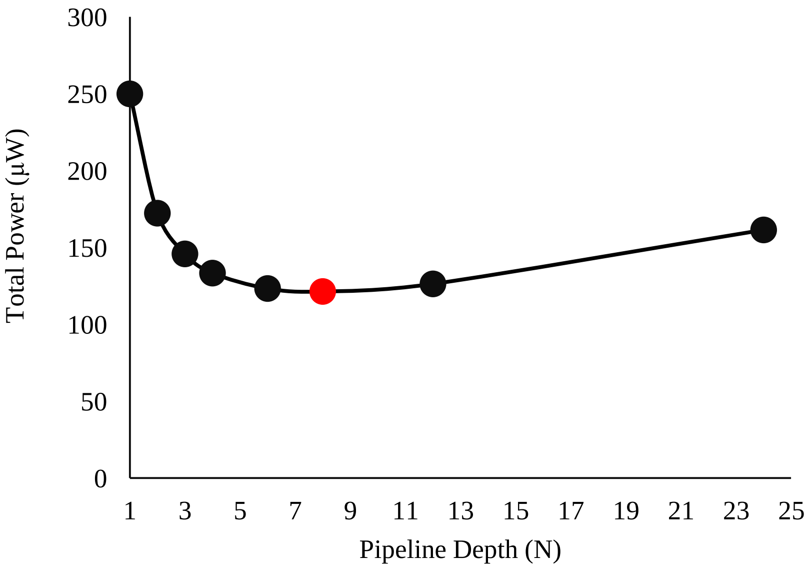


Figure 7-5 The total power consumption as a function of pipeline depth initially decreases. This is driven by reducing voltage which leads to switching power reduction. Beyond a pipeline depth of about 8, the power consumption starts increasing. This is due to increased leakage and active power contribution of flip-flops.

As seen in Figure 7-5, the total power consumption reduces as the pipeline depth in increased. The decrease in power consumption reaches a maximum at a pipeline depth of about 8. Even with a modest amount of pipelining the power consumption can be reduced by about 70% due to the V^2 relationship of switching power. Beyond a pipeline depth of 8, the leakage and switching contribution of increased number of flip-flops begins to dominate, leading to an increase in the power consumption. Thus the optimal pipeline depth for this particular design is about 8 stages. In other words the logic depth per stage is 3 adder blocks per flip-flop stage. 3 adder blocks corresponds to about 30 gates in the standard cell adder implementation explained earlier. This is reasonably close to the average number of gates per flip-flop in large circuits and ASICs (28) [ISCA-85]. Thus, a low-power pipelining approach can be used to lower the power consumption, but the increased power consumption from increased number of flip-flops limits the eventual power gains. Other logic circuits may yield different results based on individual value of logic switching capacitance, gates per flip-flop etc.

7.5 Estimating the SER of Pipelined Systems

For a pipelined system using a low-power approach as discussed above, two things are of critical importance as far as the SER of the pipelined system is concerned. Firstly, with increasing pipeline depth, the voltage can be lowered at each stage as a result of which the SET pulse-widths would increase [Dodd-03]. Secondly, the logic depth or amount of logic between individual flip-flop stages decreases. Logical masking reduces for shorter path lengths as the probability of masking SETs logically reduces as the number of possible blocking gates reduces. Thus with pipelining the effects of masking diminish. Thus these two factors need to be incorporated in estimating the combinational logic SER of pipelined systems. Apart from this, the number of flip-flops in the system increase which would increase the total flip-flop SER. As argued in previous chapters the assumption is made that the gate delays are small enough for transients to propagate unattenuated and electrical masking is not the dominant factor that affects logic SER estimates. Thus the total SER for a pipelined system of depth N is

$$\begin{aligned} Total\,SER\ (N) &= Logic\,SER(N) + Flip - flop\,SER\,(N) \\ Total\,SER\ (N) &= LM(N)\cdot SER_{logictotal}(V_N) + N^{\rho}SER_{1\,flip-flop} \end{aligned} \qquad \textbf{7-4}$$

where, SER(N) is the total SER for a pipelined system with depth N. The total SER is the sum of the logic SER and the flip-flop SER. The flip-flop SER grows with the flip-flops growth factor N^{ρ}. $SER_{1\ flip\text{-}flop}$ is the raw flip-flop and $SER_{logic\ total}$ is the SER of the entire logic in the pipelined implementation (24 adders) calculated from experimental results of a 1-bit adder as a function of voltage V_N which in turn is a function of the pipeline depth Figure 7-4. In a pipelined system with depth N, recall that the total amount of logic still stays the same. What differs is the amount of logic between individual flip-flop stages and the supply voltage. This can be incorporated separately in a logical masking function which is pipeline depth dependent (LM_N). In the following sections, the impact of pipeline depth on the voltage sensitivity of individual adders and the impact of pipeline depth on the logical masking is estimated.

7.5.1 Impact of Pipelining on Voltage Sensitivity of Adders

The 24-bit adder consisted of 1-bit adders. Due to a decrease in the voltage and the amount of logic per flip-flop stage, the impact on the SER must be modeled. To model and understand the impact of voltage reduction on the logic SER, a 1-bit adder was tested as a function of voltage for a fixed frequency of 416 MHz. The experimental setup used was the same as that introduced in the previous chapters and was based on the C-CREST technique. Figure 7-6 shows the impact of voltage variation on the combinational logic SER of the adder measured at 416 MHz.

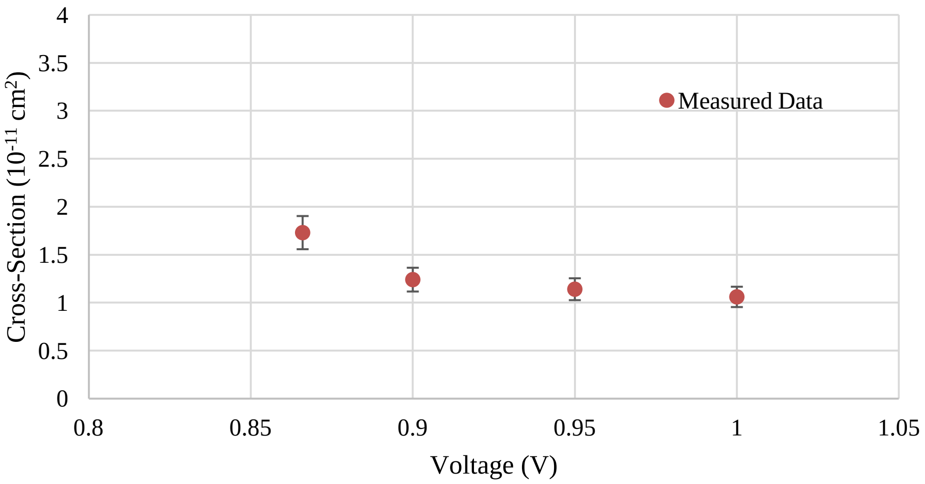


Figure 7-6 Measured alpha particle cross-section for 1-bit adder at different voltages.

Due to a decrease in the voltage the logic SER increases quite considerably. The primary reason for this is that the single-event transient pulse-widths increase. This increases the likelihood of transients latching. The range of voltages that could be tested were limited due to the fact that the on-chip PLL could not be operated at less than certain voltages. However the data can be fit using a reasonable model for the SET variation with the voltage. It is well known that the SET pulse-widths are inversely proportional to the restoring drive [Ferl-13]. In fact the charge stored by the capacitor must be discharged by the ion-strike and restored by the ON transistor(s). If the charge initially stored on the capacitor is *CVdd* where C is the output capacitance of the gate of interest and *Vdd* is the supply voltage then, the SET pulse-width is proportional to :

$$SET\ \ pulse-width\ \ \alpha \frac{Charge\,Stored}{Restoring\ \ Drive}$$

$$SET\ \ pulse-width\ \ \alpha \frac{C \cdot V}{\beta(V-|V_{thP}|)^{2}}$$

Thus the SET pulse-width can be fit to a general Equation of type

$$SET\ \ pulse-width\ \ \alpha \frac{\lambda \cdot V}{(V-|V_{thP}|)^{2}} \qquad \textbf{7-5}$$

where λ is a fitting constant which is technology dependent. The threshold voltage of the transistors was approximately 0.38 V in the linear region and 0.40 in the saturation region due to drain induced barrier lowering. We assume a value of 0.38 V. The distribution of the SET pulse-widths simulated as a function of the voltage, for a strike that deposits 10 fC of charge, is plotted in Figure 7-7. The SET pulse-widths were fit to the Equation 4 and the results are reasonable.

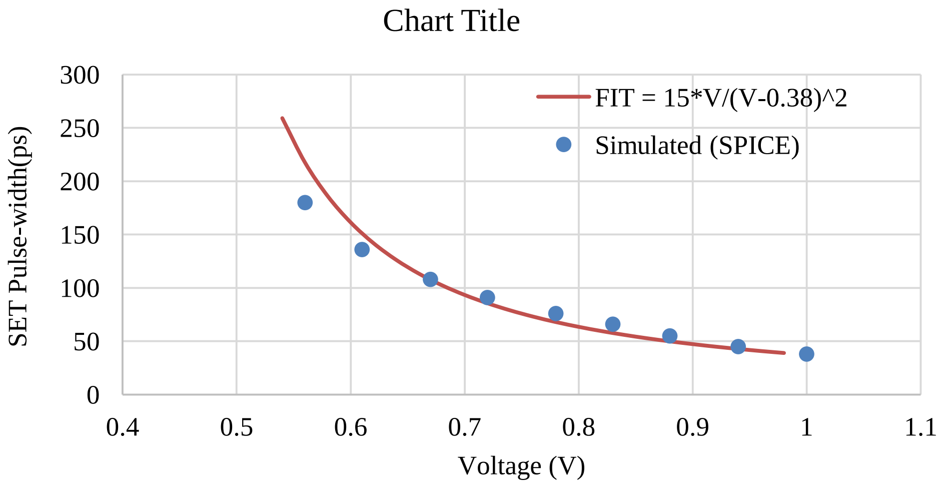


Figure 7-7 Simulated SET pulse-widths as a functions of voltage can be fit reasonably to the restoring drive current of the MOSFET.

The main reason for the increase in the voltage sensitivity of combinational logic is the increased SET pulse-widths. Using the above information about SET dependence on voltage, the experimental data from Figure 7-6 can be similarly fit to Equation 4. The fit to the data is plotted in Figure 7-8 and the fit is quite reasonable. Based on the fit to the data, the cross-section of a 1-bit adder as a function of voltage can be extrapolated for the different values of voltage shown in Figure 7-4. Similarly the 2 GHz cross-section which is relevant to the 24-bit adder framework and simulation can be obtained the by simply scaling the cross-section measured at 416 MHz by a factor of 4.8 (2GHz/416MHz). The extrapolated cross-section for different voltages and 2 frequencies is plotted in Figure 7-9. In this work, the worst case adder cross-section for the input conditions A = 0000 and B = 0000 are reported. The worst case cross-section is a reasonable estimate for the raw 1-bit adder cross-section for all input conditions. For a full adder with 3 inputs there are 8 input combinations. For the combinations where

ABC = {000, 001, 010, 100} the experimentally measured cross-section was nearly the same. This is so because the same number of gates is sensitive for each of these input conditions, regardless of logical masking conditions. For the cross-section {110, 101, 011} the experimentally measured cross-section was slightly lower than the above because transients at few gates do not appear at output. For ABC = 111, the cross-section is least because transients at almost gates are masked.

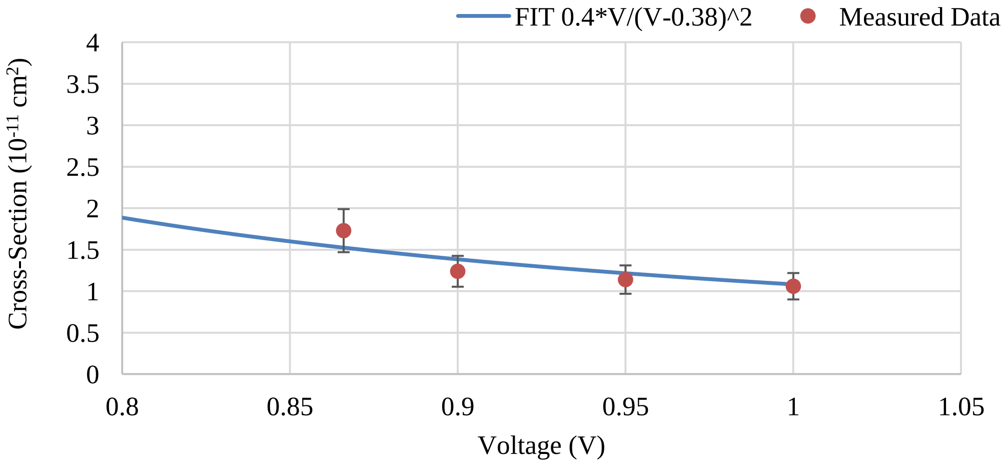


Figure 7-8 Measured data and the fit obtained from SET dependence on drive current as a function of voltage show good agreement.

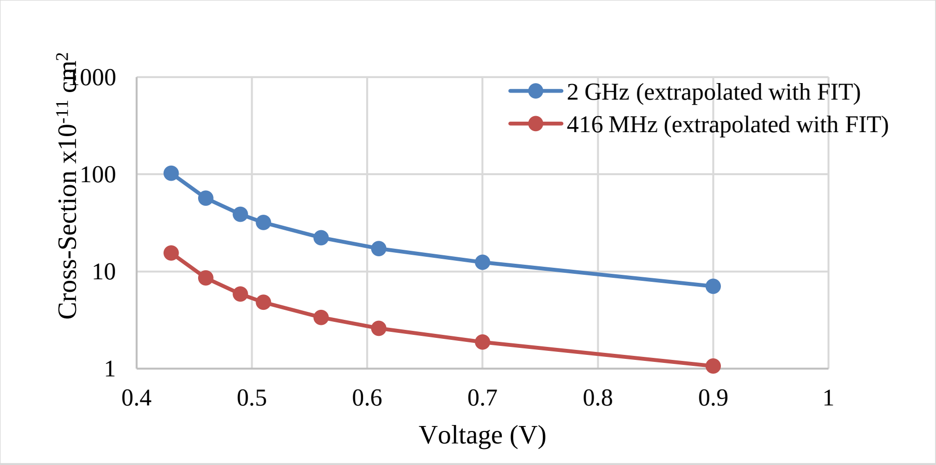


Figure 7-9 Estimated 1-bit cross-section plotted for two different frequencies.

7.5.2 Relation between Pipeline Depth and Logical Masking

The key factor that affects the logic SER when the logic depth per stage is reduced is the logical masking. In the baseline case where N = 1, a 24-bit serial ripple carry adder (RCA) is interfaced to a single flip-flop. In case of the pipelined versions of the baseline 24-bit adder, two 12-bit adders are used for pipeline depth N =2, three 8-bit adders for pipeline depth N =3 and so on. Thus at each stage, the number of adders per stage declines and decreases to 1 in the extreme case where pipeline depth N = 24. Thus, with pipelining, the depth of the logic per stage decreases. It is quite intuitive to see that when SE strikes occur deep in the logic chain (farthest from the flip-flop), the likelihood of their propagating to the output would be low because the logic in between could logically mask the errors. So in the baseline case for N =1, strikes far away from the final carry out, have a very low probability of propagating to the output. On the other hand strikes closer to the output flip-flop are much more likely to produce errors. This is more pronounced when the logic depth becomes smaller with pipelining. Thus when the logic is partitioned into smaller blocks, the amount of logic that can mask transients at each stage decreases considerably. Logical masking in this work was calculated as the proportion of faults that appear at the final carry output of a single block. So in the case of N=1 where no pipelining is implemented, random faults were simulated on the 24-bit RCA and the final carry was monitored. In the case of N=2, 2 12-bit RCAs are used. Logical masking in both logic blocks is similar. So faults are injected into only one of them and the carry signal of one of the 12-bit RCA adders is monitored. However, the calculation of the total SER of the system obviously incorporates both the adders. Combinational logic soft errors in this particular work are defined as SETs that propagate to any of the flip-flops at any time. In other words, the content of the whole pipeline, regardless of depth must be fault free at any given time instant for error free operation. The results of logical masking simulations on the adders of different sizes that are part of the pipeline of varying depth are shown inn Figure 7-10. The logical masking factor increases by 14X when the pipeline depth is increased from 1 to 24. Area-wise, there are 24 adders per flip-flop for N =1 and 1 adder per flip-flop for N =24. Thus the logic size per flip-flop is 24 time greater in the case of no pipelining. Consider a simple exercise of taking the product of logic masking with area. In this way, we get 24*0.05 = 1.37. Similarly, the product of logic masking factor and area for the case where N=24 is 1*0.725 = 0.725. But there are 24 such adder blocks. Therefore the contribution of errors from each adder block must be accounted for i.e., 24*.725 = 17.5. Thus the sum of the product of logical masking and area for the pipelined version (N=24) is a factor of 12 higher than the case where no pipelining is implemented (N=1). This simple metric gives the reader an idea about the sensitivity of the logic SER to diminishing logical masking effects introduced due to deeper pipelining.

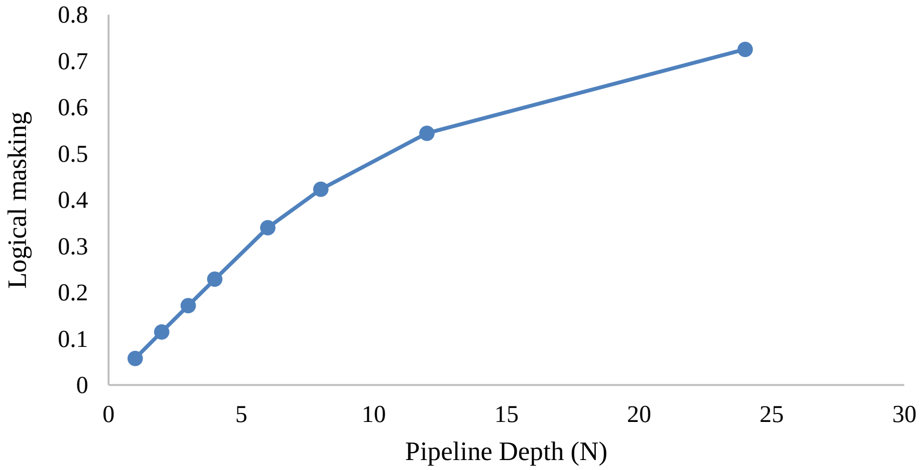


Figure 7-10 Logical masking effects are estimated for each combinational logic block interfaced to the flip-flop. For example for N = 1, faults are injected in 24 adders and the final carry is monitored. In case of N = 24, however, fault injection is performed on only one adder. So the results of logical masking are stage wise. The likelihood of transients being masked decreases with pipelining (value of logical masking factor increases). The increase in logical masking factor from N = 1 to N= 24 is more than an order of magnitude.

7.5.3 Modeling the SER with Logical Masking and Voltage Sensitivity Effects

In the previous sections the effects of increase in logic cross-section due to lower voltage and reduction in logical masking effects with pipelining were estimated. The logic cross-section and flip-flop cross-section can now be calculated for the 24-bit structure as a function of different pipeline depths. The strategy to do so is summarized once again

1. Experimental measure logic cross-section for different voltages and fixed frequency (416 Mhz) (Figure 7-6).
2. Fit the voltage variation data to SET dependence on voltage to allow extrapolation to voltages less than experimentally tested values (Figure 7-8).
3. Estimate the logic cross-section as function of pipeline depth for different voltages ($SER_{logictotal}$ (V_N) in Equation 3.
4. Scale cross-section to required frequency of operation. (logic cross-section scales linearly with frequency).
5. Calculate logical masking factors as function of pipeline depth (Figure 7-10 and *LM(N)* term in Equation 3.
6. Experimentally evaluate the flip-flop cross-section for fixed voltage (nominal).

The resultant logic, flip-flop and total cross-section as a function of pipeline depth is plotted in Figure 7-11. The logic cross-section and flip-flop cross-section both increase quite significantly. The logic cross-section increases due to lower logic masking effects with pipelining as well as increase in SET pulse-widths with pipelining. The flip-flop cross-section increase is driven by an increase in the number of flip-flops as $N^{1.2}$.

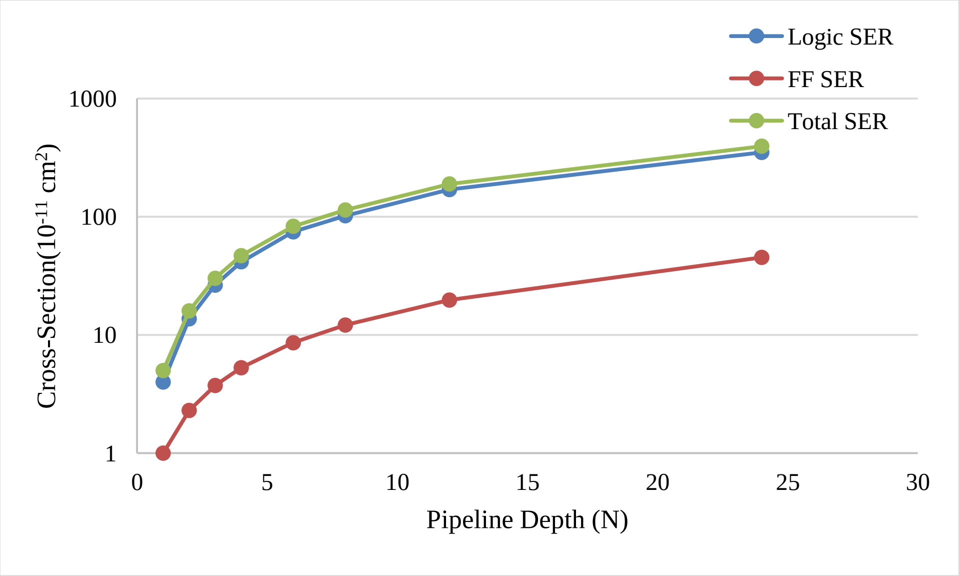


Figure 7-11 Logic and flip-flop cross-sections increase by 60X and 50X respectively.

7.5.4 Impact of hardening on Total SER

The impact of hardening the combinational logic is considered. The flip-flop SER remains unchanged. The total SER is expressed as

$$Total\ SER\ (N) = Logic\ SER(N) + Flip - flop\ SER(N)$$
$$Total\ SER\ (N) = LM(N) \cdot SER_{1-bit\,adder}(V_N) + N^{\rho} SER_{1\,flip-flop}$$

A hardening co-efficient is assigned to the logic cross-section to calculate the cross-section of the pipeline with logic hardening. Then the hardened SER is given by,

$$SER\ Hardened\ (N) = H \bullet LM(N) \cdot SER_{1-bit\,adder}(V_N) + N^{\rho} SER_{1\,flip-flop} \qquad \textbf{7-6}$$

where, H is the hardening co-efficient. For example H=0.1 corresponds to reducing the logic SER by 90%. The percentage improvement in SER compared to the baseline (N=1, no pipelining) due to different amounts of hardening is illustrated in Figure 7-12. These results provide some very interesting insights. Some of the key observations are listed below

1. Modest amount of hardening (H=0.7) does not result in any improvement in the total SER.
2. Even in the case of extreme amounts of hardening (H=0.01) the improvement is limited to 50% and the gains rapidly decrease with deeper pipelining. In fact, beyond a pipeline depth of 4, there is no improvement in SER.
3. Eventually the flip-flop SER and the growth of the flip-flops with pipelining limit the impact of hardening combinational logic.
4. Most importantly, these results highlight that circuit level hardening approaches (H=0.7 is close to improvement achieved through the use of Shannon Adders in Chapter V) may not carry through to the highest level of abstraction if sufficient cross-layer optimization for SER is not carried out.

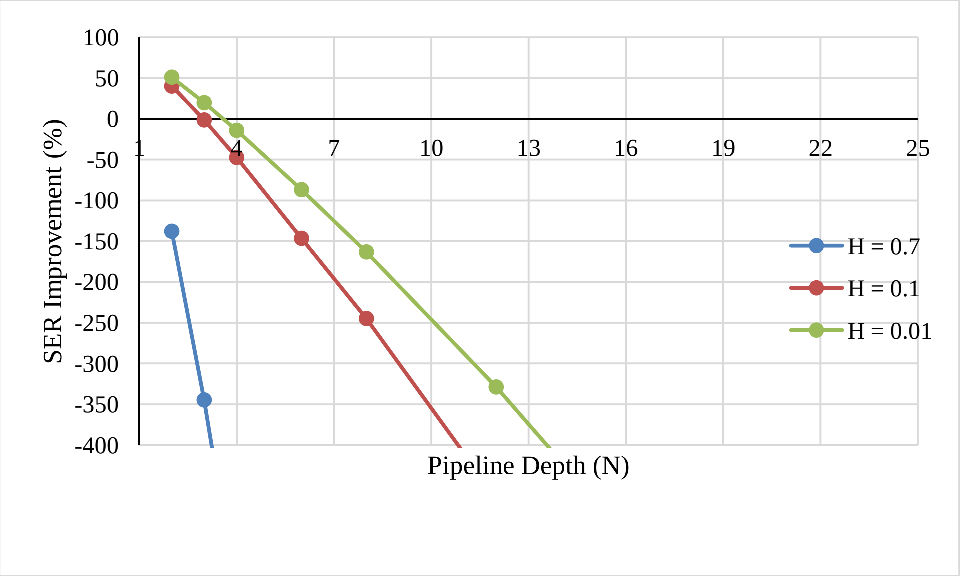


Figure 7-12 Percentage improvement in total SER for different amount of hardening shows that upto 50% improvement can be obtained in the best case (H=0.01). On the other hand, modest amount of hardening does not result in any improvement in the total SER. In all cases the growth of flip-flops limits the impact of hardening combinational logic and beyond pipeline depth of about 4, no improvement is seen.

7.5.5 Power Overheads from Hardening Combinational Logic

The power overheads from hardening must also be accounted for to understand the co-optimization of both power and SER. In the following results, the effects of hardening that lead to increased power consumption are illustrated. The example of the inclusion of an SET filter with the logic chain is used where the minimum operating voltage due to pipelining must now account for the presence of the SET filtering element. Secondly, power increase of 1.5 X is assumed due to the adopted hardening approach. The results of these comparisons are shown in Figure 7-13. This plot shows that even in the case where the SET filter is included and in the case where power increases uniformly by

1.5X, the reduction in supply voltage still results in a significant reduction in power compared to the baseline case with no pipelining (N=1).

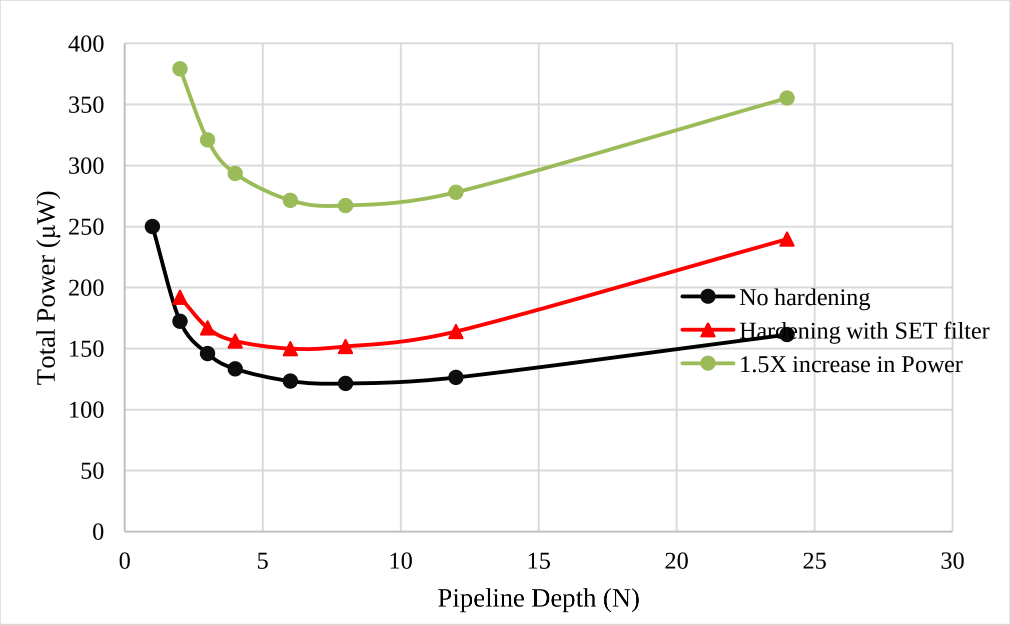


Figure 7-13 Power overheads from hardening are still less than the baseline case where no pipelining is used.

7.6 Summary

Pipelining for low-power introduces two important effects as far as the performance, power consumption and SER of pipelined systems are concerned. With a fixed throughput, lower power consumption can be achieved. However, this is achieved mainly by lowering the supply voltage for the combinational logic. Due to this, the SET pulse-widths increase leading to a direct increase in the logic SER. Similarly pipelining also results in lower logical masking effects thus increasing the logic SER. The increased slack from operating slowly can be traded off by incorporating harder combinational logic which can reduce the total SER of the pipelined system. However, care must be taken to ensure that the power overheads from the hardened logic do not offset the improvement obtained through pipelining in the first place. If this is taken care of, then, pipelining allows for lower SER and low power compared to the baseline case where no pipelining is incorporated.

In summary, if only pipelining for low-power is used, there would be a tremendous increase in SER. If only SER reduction is adopted without power considerations then in most cases the power overheads could be significant. However if a modest amount of pipelining is combined with SER mitigation approaches then *lower power and lower SER* can be achieved.

Thus, in this dissertation work, two major ideas that explore the power and soft error reliability are discussed. The first relates to the use of the Shannon expansion theorem for combinational logic protection. Shannon expansion, if performed appropriately can improve the combinational logic SER quite substantially as well as reduce the dynamic power consumption. This is achieved by partitioning circuits in such a way that the effective sensitive area is reduced and the switching activity of the circuit also reduces. Area overheads of up to 2X and small speed penalties can be expected with such a combinational logic mitigation approach. The second major idea relates to the use of pipelining to reduce dynamic power consumption. In this approach, the pipeline depth of datapaths or logic chains is increased while maintaining the frequency of operation. The amount of combinational logic in each stage decreases as the pipeline depth is increased. However as the clock frequency is not increased, the combinational logic computation time for each stage is less than the clock delay. The combinational logic computation time at each stage can then be increased so that it is close to the clock time period and satisfies the setup and hold timing constraints of the flip-flops, by lowering the operating voltage of the combinational logic. Lowering the combinational logic supply voltage reduces the dynamic power consumption substantially but is accompanied by a corresponding increase in the combinational logic SER due to lower supply voltage. Increased number of latches also increase the overall latch SER. The area overhead however comes from the increase in the number of latches and an increased latency for task execution. Thus there is a trade-off between power and soft error reliability. Designers may however choose to combine pipelining or Shannon expansion along with other hardening approaches (which may have power overheads) to meet the power and soft error reliability budget of the circuit. In the following chapter the impact of the different hardening and low power approaches proposed in this work are discussed in unison. This will provide designers a mean to choose from and combine a variety of different hardening and low-power approaches so that the effective power and SER of the circuit is improved. For example consider the hypothetical case in which, doubling the pipeline depth of a baseline circuit results in 50% power savings but increase the SER by 3X. On the other hand a hardening approach to improve SER leads to SER improvement by 10X with 20% power penalty. In isolation the first technique reduces power at the expense of SER and vice versa for the second. However if the two are combined the effective resul;t could lead to 40 % power reduction and 70 % SER reduction. In circumstances where power and SER are both important such trade-offs and design practices may become essential to reduce power and SER. The following chapter discusses the different aspects of hardening circuits and lowering power in isolation and then in unison to offer designers a variety of options and a guiding framework for low-power hardening. The associated area and performance penalties are also discussed.

Chapter VIII. Summary, Recommendations and Future Directions

"Designers will soon have to cope with a reliability wall..after having tackled the power wall (sic)"

-- Pradip Bose, IBM, Invited Talk, IRPS 2014.

As technology has scaled, semiconductor devices and circuits that operate at multi-GHz speeds consume lot of power and are prone to failures of different kinds. This dissertation has shown that combinational logic soft errors have emerged as a major threat as far as the reliability of high-speed ICs is concerned. At the same time, as the number of connected and battery powered devices grows around us, there is a concurrent need to minimize power consumption as well. Thus designers must ensure both low-power consumption and high reliability. However, while trying to achieve this goal there are two major drawbacks. Most approaches that improve the soft-error reliability result in power overheads as well. Popular approaches, such as increasing transistor sizes or nodal capacitances to mitigate transients, filtering transients, adding redundant circuits or computing repeatedly to ensure correct operation upon error detection, incur power overheads. On the other hand, standard approaches to reduce power consumption, such as reduction in supply voltage and capacitances, using smaller devices with lower drive, high-vt devices etc. inevitably result in higher SER.

As a solution, this dissertation proposes approaches, such as Shannon Expansion theorem that can be used to decrease the logic SER as well as the power consumption. However this approach could be limited by the area overhead and the fact that the SER and power benefit is circuit dependent. The improvements in SER and power were in the range of 10-45 % for different benchmark circuits analyzed. However, in many circumstances, it may be necessary to meet certain specific power and SER targets. In such cases, designers may have to combine one or more approaches to meet these targets. This chapter presents the designer some insights into how the different approaches described in this work and elsewhere in literature can be combined to “effectively” reduce the power consumption and SER of circuits to meet the power/SER specifications. This provides a holistic picture of hardening, power minimization and associated pitfalls. Area and/or performance and design effort penalties are also discussed.

The chapter presents a particular problem in the form of an SER and power specification and then discusses 3 different scenarios to try and meet this specification. The first section discusses different hardening schemes only. A brief review of the Shannon expansion approach is summarized and the key experimental results that show the improvement in power and SER for Shannon Adder circuits are presented. The results are presented in the context of a 24-bit adder for easy comparison with the pipelining results. Section 2 discusses the impact of pipelining on the SER and power consumption of a 24-bit adder circuit. Section 3 addresses the following questions in tandem 1) What is the impact on power consumption and SER when both Shannon approach and pipelining approach are combined? 2) When should pipelining be adopted in the design stage? 3) What is the observed impact on power consumption and SER when standard hardening approaches and pipelining are combined? The chapter concludes with general learning, key findings, guidelines and rubrics so that designers can make informed choices about the right hardening approach to reduce power and SER in the best way possible.

8.1 SER Mitigation and Low-power Approaches

Consider a specific design constraint: 40 % reduction in SER is required along with a 20 % reduction in power compared to a baseline circuit to meet the overall SER and power budget. From previous discussions it is known that conventional hardening techniques introduce power overheads. Similarly power reduction approaches, such as pipelining introduce power overheads. In the following analysis, SER hardening and power mitigation are considered in isolation first and then they are considered jointly to minimize the SER and power consumption. In this way a basic understanding about different factors that affect the power-SER trade-off and how different approaches can be combined for a better solution for power and SER reduction is provided. The examples considered here are not as rigorous as those presented in earlier chapters or representative of all circuits or design approaches, but only to illustrate a general approach to attacking the dual problem of power consumption and soft error reliability.

8.1.1 Goal 1: Only SER Mitigation with power overheads

In this section the best possible reduction in SER is sought and power overheads of different approaches are evaluated. The primary goal here is logic SER mitigation. Three different techniques are briefly discussed. The first deals with Shannon expansion, introduced in earlier chapters as an effective means to reduce the dynamic power consumption and the logic soft error rate. This is achieved by effectively partitioning the circuit such that fewer nodes are switching every clock cycle and the effective circuit sensitive area is reduced. Such a technique can be applied to all combinational logic circuits and can also be extended to partition the circuit repeatedly. Similarly circuit approaches such as implementing the circuit with transmission gates is permissible with Shannon expansion. The reader can refer to detailed discussion in Chapters 4, 5 and 6. In section 6.4 two different adder circuits were compared. A standard CMOS adder implemented using standard CMOS gates was compared to an adder based on Shannon expansion theorem and implemented using transmission gates. The results indicate that the combinational logic cross-section improves by 35 % or H=0.65X (recall hardening co-efficient from Section 6.4) compared to the standard cell adder case. At the same time, the power consumption reduces by 65 % (0.35X). The area of the Shannon implementation is however 2.3X less than the standard cell adder. It can be said reasonably that the percentage reduction in power and SER for 24-bit adder (refer to discussions in Section 7.5.3 for 24-bit adder details, simulation and modeling approaches and pipeline models) implemented with Shannon expansion and transmission gates will be similar to that obtained from a 1-bit adder. This is so because logical masking does not change from one stage to the other and it is assumed that electrical masking is negligible in both cases. The effective improvement in SER and power across different cases for the two 24-bit adders is shown in Figure 8-1. The size of the bubble is indicative of the area of the circuit. Along with the Shannon Adder, two other cases where the circuit is hardened by 90% (H=0.1X) and by 99% (H=0.01X) are shown. These cases correspond to the use of SET filters and layout based hardening (LEAP). Generally speaking SET filters are designed in a way that all the maximum expected SET pulse-width can be filtered, but a 90% efficiency is a reasonable assumption given the experimental error in estimating SET pulse-widths and impact of voltage and process variation. The use of 0.01X for the LEAP approach is from experimental results presented in [Lilj-14] where this approach has been shown to be efficient in reducing combinational logic SER as well as latch SER. The three cases are contrasted in Figure 8-1 for the relative improvements in power and logic SER for the three different hardening approaches. With extreme hardening (0.01X) the highest reduction in logic SER is accompanied with highest power and area overheads compared to the baseline circuit with no hardening. On the other hand, modest amount of hardening with filters, for example (H=0.1X) results in substantial logic SER reduction but does come with a small power overhead as well as a modest area penalty. The Shannon approach provides the least reduction in combinational logic SER among the three approaches but does not suffer any power or area overheads compared to the two other approaches. Note that the reduced area overhead in this case from Shannon expansion is because of the use of transmission gates. Use of standard CMOS gates will result in an area overhead.

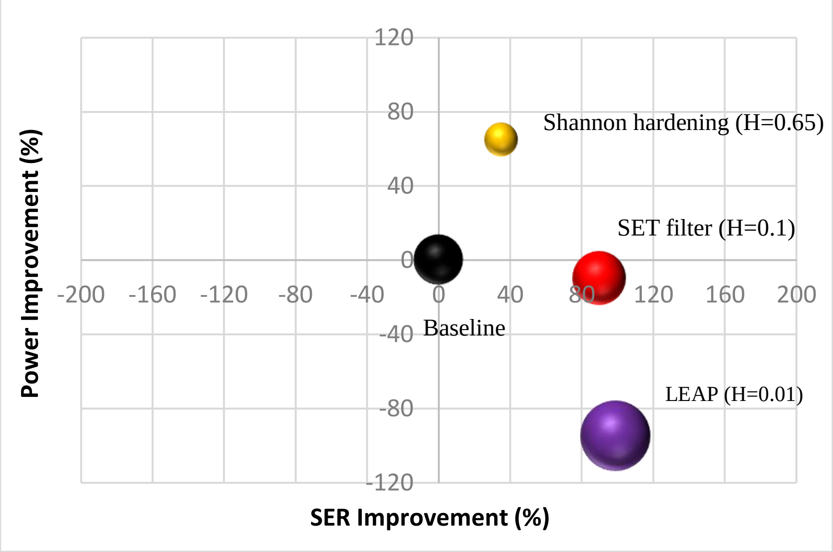


Figure 8-1 Power-SER comparison of different hardening approaches. The size of the bubble represents the area of the circuits.

Thus logic SER mitigation approaches can provide different levels of logic SER improvement. Generally speaking from these three techniques, as logic SER reduction improves, the power overheads may increase. Thus while a design that is hardened may meet the SER budget, the associated power overheads may hurt the power budget. In fact only one of the solutions provides both, lower SER and lower power than the original (as seen in Fig. 8-1). However the important consideration for designers is whether the approach that is adopted meets the specified targets for SER and power. The above methods are fairly generic and can be applied at any design stage but it is important to remember that design tweaks at the end of the design cycle can hurt the power budget if sufficient care is not exercised.

8.1.2 Goal 2: Pipelining for Power Reduction and impact on SER

In chapter 7 we have seen that pipelining is a powerful tool to achieve low power consumption. The performance and area overheads from such an approach are not very significant either. From Figure 7-5, it is seen that an increase in the pipeline depth while lowering the operating voltage results in lower power consumption up to a certain pipeline depth. However, the minimum is quickly reached at N=2 or N=3 for Vdd values of ~0.7-0.65 V. Practically it is reasonable for operating voltages to be lowered to these values in high speed paths but not much lower because of the increased impact of process variation which requires further design guard banding. But pipelining for low power inevitably increases both, the logic as well as latch SER due to lower logic Vdd and increased latch count with deeper pipelines. Thus while low-power pipelining can reduce the power consumption by even as much as 60-70 %, the SER overheads can be substantial. This is shown in Figure 7-5 and Figure 7-11 respectively.

From the two approaches discussed above we see that there is a flaw in the design philosophy and that is the fact that often improving power and SER is never considered jointly. If power optimization and SER mitigation is done in isolation (as is the case for most designs), either of the two (power consumption or SER) is likely to degrade compared to the original circuit. The next section discusses how the two approaches can be synthesized into a cohesive strategy to reduce the power consumption and SER of circuits.

8.1.3 Goal 3: Joint power and SER Minimization

Consider a specific design constraint : 40 % reduction in SER is required along with a 20 % reduction in power to meet the overall SER and power budget. It is known from earlier discussions that conventional hardening techniques introduce power overheads. Similarly power reduction approaches, such as pipelining, introduce power overheads. There are indeed several different hardening approaches and their associated impact on the power consumption. Therefore one approach designers could adopt is to first evaluate the improvement *only* in SER that each hardening approach brings about as illustrated in Figure 8-1. Clearly only two of three hardening approaches meet the required SER target (40% reduction). The Shannon approach can be eliminated at this stage because it does not meet the specified criteria (and it is known that adopting pipelining at a later stage to lower the power consumption will only *increase* the SER). Thus without pipelining the circuit for low-power, if an approach does meet the SER target then it must be eliminated. Following this the circuit can be pipelined in an effort to reduce the power consumption to meet the power budget. For simplicity, if the pipeline depth is limited to N=2. The resultant impact on the SER and power is shown in Figure 8-2.

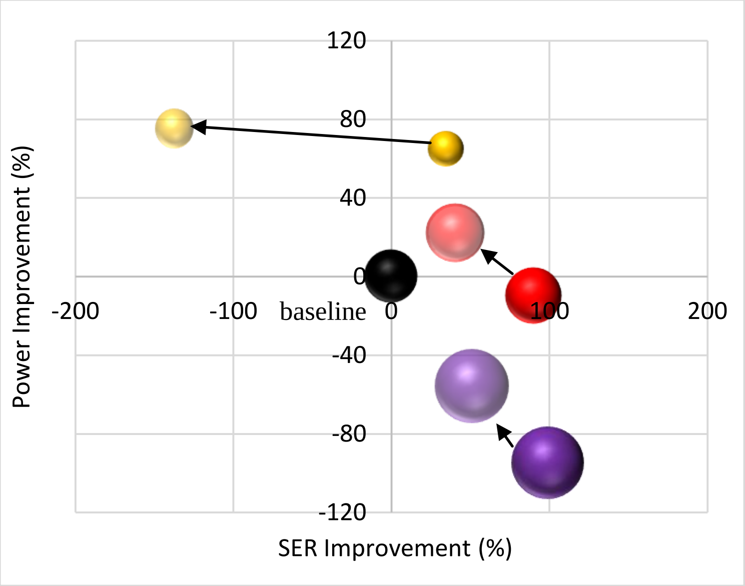


Figure 8-2 Power-SER comparison of different hardening approaches with low-power pipelining N=2. Increasing the pipeline depth decreases the power consumption for all the hardening approaches. However, the SER improvement decreases for all the approaches as well due to the exacerbating effects of reduced voltage. The dark bubbles represent SER-power with only the hardening approach applied. The light bubbles represent hardening followed by two-stage low-power pipelining (N=2). The area increases marginally with this modification in the pipeline depth. The arrows indicate the change in power and SER from hardening the baseline to adding pipelining to the hardened circuits.

In Figure 8-2, the dark bubbles represent hardening applied to the original baseline circuit. The light bubbles represent two-stage pipelining applied to the hardened circuits. Some very interesting results emerge after the hardened circuit has been pipelined to reduce power. Firstly, the power consumption reduces for all the hardening approaches as expected due to a reduction in the logic voltage. This is accompanied by a small increase in area as well due to increased number of latches. But as this is a small portion of the circuit, the increase is modest in each case. It should be noted that the size of the bubbles indicates the area. More importantly though, power reduction is accompanied by a corresponding degradation of the SER in each of the cases. This is indeed due to the effects of reduced logic voltage. The sensitivity of the logic is assumed to be the same in each of the cases. Looking carefully, it is seen that now in only one case (H=0.1, N=2 -> SER reduction 42% power reduction 22%) the target specification of 40% SER reduction and 20 % power reduction is met. Further increase in pipeline depth will of course reduce power but also increase SER. In circumstances where a solution is not found the process can be continued with deeper pipelining. A potential candidate could be H=0.01 and N>=3 to see if the SER improvement and power reduction is better than that offered by H=0.1 and N=2. It is also worthwhile to note that while power and SER are the major parameters being monitored in this discussion, area is also an important concern. The Shannon approach with no pipelining provides the reasonable solution with area reduction as well. This may be an important consideration during the design process as well.

This general discussion highlights a few key takeaways

1. Most hardening approaches result in power overheads and similarly power minimization techniques are usually accompanied with SER overheads.
2. In a design approach where both SER and power are important, a general strategy such as evaluating the best hardening approach followed by low-power approaches, such as pipelining can be a useful technique to eliminate non-feasible solutions when meeting power and SER budgets is essential.

3. A joint SER-power reduction approach is useful only if incorporated into a design flow very early. A separate discussion on when the above approach of joint power and SER optimization is more applicable follows in the general guidelines section below. Alternatively when such an approach is not practical is also included in the general guidelines below.

8.2 Summary

Some of the results in this work have shown that as far as the impact of technology scaling is concerned, combinational logic soft errors as measured per logic gate show a decreasing trend. This is consistent with the fact that as technologies scale, the area available for single-event strikes reduces. At the same time the drive currents do not change drastically with some preference towards modest increases from one generation to the next. This results in smaller pulse-widths. Thus the overall effect of technology scaling is to reduce the soft error rate of combinational logic circuits (per gate) in the terrestrial environment. This has been established through simulations as well as experiments at the 40 nm, 28 nm and 20 nm bulk technology nodes. Based on simple models and experimental results that compare the logic SER of 10 inverters to a latch, it is reasonable to say that the combinational logic soft errors are as important if not more important than latch errors for modern semiconductor circuits. In fact, in the 2+ GHz range, combinational logic soft errors are clearly a much bigger threat compared to latch errors. In harsh environments like space the problem of combinational logic SER is well known. In such environments logic SER is a big threat at even lower frequencies of operation. The trends with technology scaling are very different from those in the terrestrial environment. Experimentally it was observed that with scaling the ratio of logic to latch SER actually increases. The primary reason for this is that the layout can affect the cross-section as well pulse-width. This fact needs to be considered carefully for future generations with the adoption of finFETs etc.

The second and key goal of this work is to explore ways and means to mitigate combinational logic soft errors without any power overheads. In other words design circuits such that they consume less power and have higher combinational logic soft error reliability. The trade-off between power and SER was evaluated at three different levels of abstraction: 1) gate-level (Masters : discussion follows in Appendix I) 2) circuit-level and 3) architectural-level. At the circuit level an attempt was made to reduce the number of actively switching nodes and reduce the number of nodes sensitive to single-event particle strikes at the same time. This reduces the dynamic power consumption as fewer nodes switch and the soft error sensitivity through logical masking of transients from non-essential circuit nodes. Shannon's theorem was employed to achieve reduction in power and combinational logic soft errors.

At the architectural level, low-power pipelining inherently increases the soft error rate. By pipelining the circuits and holding frequency constant, voltage in each pipeline stage can be reduced. Such a scheme offers constant throughput but lower power consumption. However the smaller logic depths and lower voltages lead to an increased likelihood of combinational logic soft errors. This can be countered by carefully hardening the circuits such that a combination of hardening and pipelining then results in lower power and SER than a baseline circuit with no hardening or pipelining. Merely hardening it introduces power overheads while merely adopting low-power pipelining increases the SER.

Analysis in this dissertation shows that combinational logic soft errors are a key problem for designers today. Solving this issue requires different forms of hardening approaches at different level each of which can have area, power and speed penalties. Clever manipulations of circuit designs result in lower power and SER. This is critical in the context of modern IC design that emphasizes low-power and high-reliability operation. In conclusion the key findings are summarized in the form of guidelines for designers to achieve low power and SER especially when protecting combinational logic circuits.

8.3 Guidelines and Recommendations

1. **Technology Scaling Considerations for Logic SER:** In the terrestrial environment, combinational logic soft errors are as important as or even more important than latch errors as far as system level error rates are concerned. In the 2.5+ GHz range, alpha particle logic SER easily exceeds the raw latch SER. Scaling leads to a reduction in the SER per gate and per latch, but designers must keep in mind that the number of devices grows with each technology generation which retains the relevance of the problem at each technology generation.
2. **Circuit-level power-aware techniques to mitigate combinational logic soft errors.** While adopting circuit level techniques to mitigate combinational logic with minimum or no power overhead two key factors must be accounted. Firstly any approach such as Shannon expansion presented in this work, that reduces the number of switching nodes and at the same time reduces the number of nodes sensitive to transients will lead to power and logic SER reduction. In this context the time tested approach of shutting off parts of the circuits that are not required

works equally well for power reduction and SER reduction. Circuit level approaches to minimize power consumption and logic SER can be refined if there is some knowledge of the input vectors and circuit functionality. Using this information, idle sub-circuits in the circuit can be identified so that these can be separated from the main logic block and gated/disabled and the most frequently used sub-circuits can be activated more often. This reduces power consumption and minimizes transients from inactive or idle circuits. The circuit level approaches such as Shannon expansion and its variants discussed in this work are very generic and can be applied to circuits at any stage of the design process. While opting for standard cell design approaches area overheads of as much as 2X can be expected for 10-50% logic SER improvement. Full-custom design where specialized structures like transmission gates etc. are available can result in faster, smaller, lower power circuits with higher logic soft error reliability can be designed using Shannon approach. The design effort in this case is higher because delay estimation for transmission gate structures is difficult.

3. **Architectural-level power-aware techniques to mitigate combinational logic soft errors.** At the architectural level design changes are limited to redesign through pipelining, reordering execution or parallelization of the architecture. Usually, designers adopt pipelining to increase performance. This is the first step in high-speed datapath design and all the focus is on performance and maximizing speed. While dealing with power, the approach in the past has been to move to multi-core operation so that the frequency and pipeline depth does not have to be increased significantly. For such high-speed designs, soft error

considerations often come at the very end of the design process. In such circumstances designers may be better off opting or the low power logic SER mitigation approaches discussed in Chapters 4, 5 and 6. However another form of architectural design change that can be used to reduce the power consumption is pipelining for low power. Low-power pipelining is effective in reducing the power consumption through reduction in logic supply voltage but increases the combinational logic SER. This is because lower voltage results in longer SETs and the combinational logic is partitioned into stages which reduces the logical masking. Low-power pipelining is less applicable to high performance circuits like server processors but more so for ASICs, DSP pipelines or sea-of-gates implementations in large SoCs. Low-power pipelining must ideally be adopted early in the design process to ensure correct timing closure in the presence of multiple Vdd domains that affect the delay. However, if SER is also an issue then some of the power gains from low-power pipelining can be traded for SER mitigation by making the appropriate hardening choices. For example the discussion in this chapter illustrates how a careful choice of hardening technique and pipeline depth a circuit can be designed to have lower SER as well as power than a baseline circuit with no hardening and pipelining.

8.4 Future Directions

This dissertation for the first time has provided experimental trends as far as combinational logic soft errors are concerned. These trends suggest that keeping all things the same, logic SER per gate (as with SRAMs and flip-flops) will reduce with technology scaling, which is the opposite of what was predicted using simulation models

12 years ago [Shiv-02]. With the adoption of FinFETs, which result in smaller sensitive area, some of the assertions made in this work are likely to continue to hold for the next few generations of CMOS transistors.

The second most important contribution of this thesis is to illustrate that low power is possible with higher soft error reliability. Soft error mitigation and power minimization can be jointly achieved by reducing certain key factors that affect both. This can be achieved at various levels of abstraction beginning from the device-level right up to the architectural level. This is a rich, unexplored area of research where plenty of approaches can be synergized to achieve lower power and lower SER. Some of the area of research that could be of great interest in the coming years are listed here:

1. Behavioral level: Circuit descriptions can be behaviorally changed to tailor these towards low power and high reliability targets. For example does a carry-skip adder consume less power than a Manchester carry adder and result in fewer errors?
2. Circuit synthesis: Topological mapping of behavioral descriptions to gates as has been shown in this work can have huge influence on power and SER reduction. Approaches tailored towards low-power, low-area and low-delay can have important effects on the logic SER. Some of this has been explored in part in [Limb-12].
3. Circuit level: Standard cell CMOS design can be potentially replaced with circuit families like transmission gate logic that potentially provide higher reliability and potentially lower power. Formal approaches to design such circuits under reliability constraints is a challenging problem.

4. Clock gating for reliability: Can parts of circuits be clock gated or even power gated to improve reliability?
5. Dynamic Voltage and Frequency Scaling: Can system voltage an frequency be tailored and tuned to meet required reliability targets just as for power targets. For example is it better to run at high speed and complete some tasks then drop down to lower voltage and speed and complete the rest of the task to save power over the duration of the task or is better to run at a fixed speed and power throughout as far as reliability is concerned?
6. Pipelining and Parallel Architectures: Modern CPUs employ multiple cores that are pipelined in different ways. Can pipelining and parallelization be used in tandem to reduce the logic SER and power as has been briefly demonstrated in this work? What about the task completion speeds? Is it better to use one processor at 5 GHz or two processors in parallel at 2.5 GHz and slightly lower voltage? There is some serial component of programs that cannot be spread across two cores to achieve exact sped of 2. Is such cases what is the more reliable option?

The above open questions span the entire reliability-power-performance trade space across different levels of abstraction. As this thesis has shown, making changes to designs at one level of abstraction may or may not have the desired impact at a higher level of abstraction. A cross-layer optimization strategy is critical to achieve low-power and higher soft-error reliability.

Appendix A : Selective Node Hardening for Logic SER Mitigation

In this appendix and the following two appendices, some work pertaining to different aspects of combinational logic hardening and experimental evaluation of logic SER is reported. Some of the results are critical in being able to estimate the logic SER of large circuits efficiently and in the presence of non-ideal effects such as the impact of chi-level voltage drop and its impact on the logic SER. The contents of the appendices have been drawn from the following three papers.

Appendix I : Mahatme, Nihaar N., Indranil Chatterjee, Akash Patki, Daniel B. Limbrick, Bharat L. Bhuva, Ronald D. Schrimpf, and William Robinson. "An efficient technique to select logic nodes for single event transient pulse-width reduction."*Microelectronics Reliability* 53, no. 1 (2013): 114-117.

Appendix II: Impact of voltage on logic SER: Mahatme, N.N.; Gaspard, N.J.; Jagannathan, S.; Loveless, T.D.; Bhuva, B.L.; Robinson, W.H.; Massengill, L.W.; Wen, S.-J.; Wong, R., "Impact of Supply Voltage and Frequency on the Soft Error Rate of Logic Circuits,"*Nuclear Science, IEEE Transactions on* , vol.60, no.6, pp.4200,4206, Dec. 2013.

Appendix III : Fast Estimation of Logic SER and comparison with Latch SER : Mahatme, N.N.; Gaspard, N.J.; Jagannathan, S.; Loveless, T.D.; Abdel-Aziz, H.; Bhuva, B.L.; Massengill, L.W.; Wen, S.; Wong, R., "Estimating the frequency threshold for logic soft errors," *Reliability Physics Symposium (IRPS), 2013 IEEE International* , vol., no., pp.3D.3.1,3D.3.6, 14-18 April 2013.

1. Probabilistic Node Hardening

For older technologies, the hardening of a node, or a circuit path, was achieved by increasing the nodal capacitances. For a given node with capacitance C, the charge stored at the output is given by C * V_{dd}. To introduce a rail-to-rail transient pulse in the circuit, the hit node must collect more charge than what is stored at the output node. If the value of the capacitance is increased (primarily by increasing the input capacitance of the succeeding gate), the charge required to generate an SET pulse also increases, thereby hardening the circuit node [Zhou-04]. This approach worked for older technologies where the value of charge stored at a node was significantly higher than a few pC. If the initial value of capacitance is only a few fC, as is the case for advanced technologies, the increase in capacitance values required to attenuate the transient becomes prohibitively high [Dasg-07]. As a result, instead of increasing nodal capacitances, increasing the restoring current at the struck node is a better approach for advanced technologies.

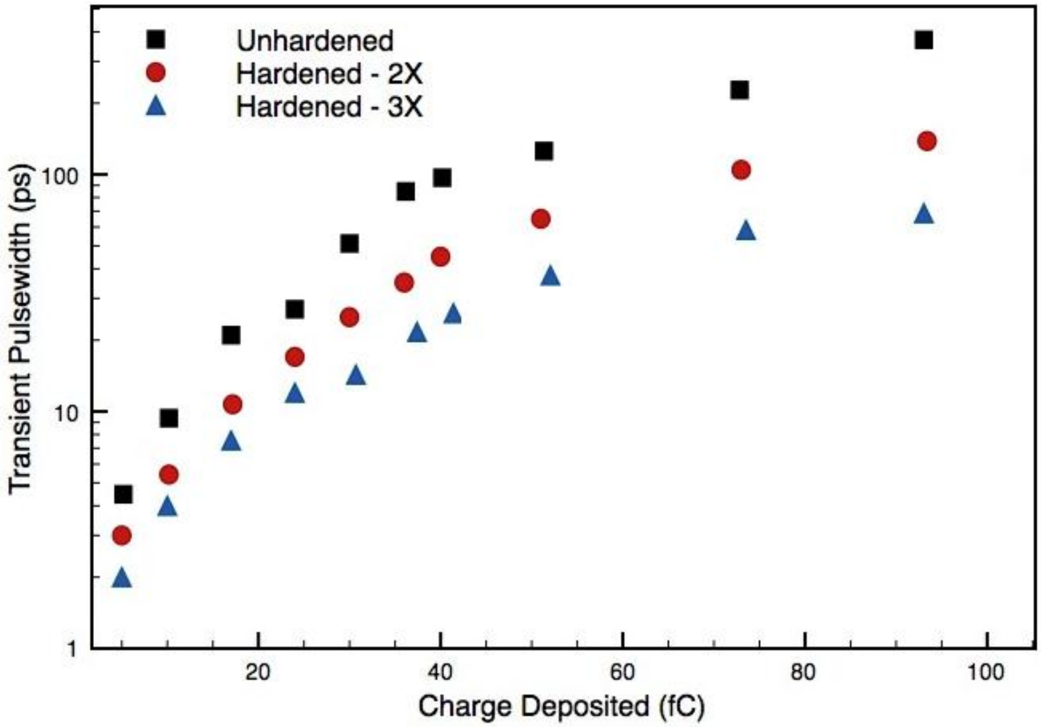


Figure A1-1 Simulated transient pulse-widths versus charge deposited for 1X, 2X and 3X width of a resized pMOS arrays of a NAND gate. The 1X, 2X and 3X widths are designated as unhardened gate, 2X hardened gate and 3X hardened gate.

For combinational logic circuits, the hit node will always return to its original nodal voltage (assuming low frequency operation), resulting in an SET at the hit node. Usually an OFF transistor associated with a node is hit by an energetic ion and ON transistor(s) associated with that node removes the charge collected as a result of the hit. For CMOS technologies, if the hit transistor is an n-MOSFET, then the restoring transistor is a p-MOSFET. The SET pulse width is determined by the amount of charge collected and the current drive of the restoring transistor. The amount of charge collected is usually a technology dependent parameter and designers have very little control of it (except parasitic bipolar transistor size). As a result, restoring transistor size is the only controllable parameter that affects the SET pulse width. Figure A1-1 shows the resultant

SET pulse width as a function of collected charge and restoring transistor size. It is clear that increasing restoring transistor size will significantly decrease the SET pulse width.

The proposed approach identifies the nodes that are most sensitive and/or vulnerable to SE effects. The key idea behind the technique is to identify the nodes at which the probability of transients being generated is high and their propagation probability through the logic chain is also high. Previous approaches identified the most sensitive nodes by looking at only the logic masking effects. However it is important to consider the likelihood of a hit by an ion since SETs are generated when OFF transistors are hit by an ion. If either of the transistor arrays in the CMOS logic (PMOS array or NMOS array) have a greater probability of being turned on, then OFF transistors can generate transients when they are hit. Thus the probability of a transistor being OFF or ON cannot be ignored. The proposed approach takes into consideration all of these factors to determine the node vulnerability. Once the nodes are rank ordered in terms of their vulnerabilities, designer then can select the set of nodes to harden for maximum impact on error rates.

2. Node Vulnerability Estimation

For any given circuit, some of the gate outputs will be in either the HIGH state or the LOW state for a greater percentage of input vectors, assuming equally likely input probabilities at the primary inputs of the circuit. As a result, the probability of producing SETs due to n-hits is greater than that due to p-hits if the gate output stays in the HIGH state for a greater percentage of input. The converse is true for the logic LOW state. Additionally, the SET pulse width for an n-hit or a p-hit is inversely proportional to the current drive of the restoring transistor for the hit node. An increase in the restoring current will lead to a decrease in SET pulse width, assuming all other factors remain the same. Such an approach will reduce the electrical masking and latch-window masking probabilities without significant penalty for the design performance. The main objective is, then to identify the nodes that are most likely to generate an SET that will reach a storage node. The algorithm to prioritize nodes for hardening for the proposed approach is described below.

The probability of signals assuming a logic 1(0) value has been defined as P_{high} (P_{low}) in this chapter. P_{high} can be used to give information about logical masking as a function of nodal probability values. P_{high} (P_{low}) represents the percentage of input vectors for which the n-MOSFETs (p-MOSFETs) connected to the gate node are OFF. For conciseness, P_{high} is used to illustrate the methodology for all following calculations, although the principle works equally well for P_{low}. Moreover, the terms "nodes" and "gate outputs" may be used interchangeably. The gate outputs with $P_{high} > 0.5$ have higher probability of being in the logic 1 state than in the logic 0 state. Gate outputs having relatively high values of P_{high} are therefore more likely to produce SETs due to n-hits. If transients generated at these gate outputs have a high probability of propagating to the output, then those gates are considered sensitive and are targeted for hardening. For such nodes, as the probability that a p-hit will occur is relatively small, it doesn't merit consideration for hardening. As the SET pulse width for n-hits is a direct function of the restoring current drive of the associated pull-up p-MOSFETs, an increase in p-MOSFET size decreases the SET pulse-width at these nodes. Conversely, nodes having low values of P_{high} are more likely to produce SETs due to p-hits and increasing the restoring current drive of the associated n-MOSFETs will reduce the SET pulse width.

The following discussion, using the example circuit shown in Figure A1-2, demonstrates the use of P_{high} to identify the most vulnerable gates in a circuit. The calculation of node signal probabilities is described in [Najm-91, Park-75]. The inputs to the system are assumed to be uncorrelated. For uncorrelated inputs, if $P1$ and $P2$ (representing P_{high}) are input signal probabilities to an AND gate, the output signal probability is given by ($P1 \cdot P2$). For an OR gate the value is ($P1 + P2) – (P1 \cdot P2$). For an inverter, the output signal probability is $(1 – P1)$. To suppress the effects of signal correlations and re-convergent fan-outs, literals in products that are repeated are accounted for only once. For example, in the probability equation of a logic gate, if the term P_i is repeated in a product, it is accounted for only once. For example $P1 \cdot P1 = P1$. And $P_{high} + P_{low} = 1$. Also the product of probabilities of inverted signals is 0,

i.e., $P(i)(1\text{-}P(i)) = 0$.

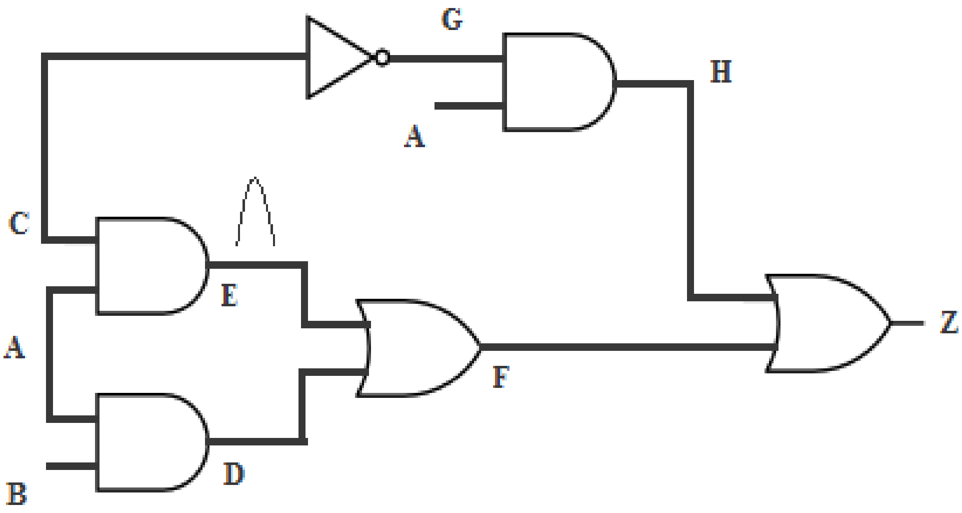


Figure A1-2 Representative circuit for which probability and Logical Masking Metric values have been calculated

For the circuit shown in Figure A1-2, the probability P_{high} for node F is

P(F) = P(A.B) + P(A.C) – P(A.B)P(A.C) **A1-1**

Since the inputs are uncorrelated,

P(A.B) = P(A)·P(B) **A1-2**

and

P(A.C) = P(A)·P(C). **A1-3**

Suppressing P(A) in the third term in (1), we get

P(F) = P(A)P(B) + P(A)P(C) – P(A)P(B)P(C) **A1-4**

and

$$P(Z) = P(A.C') + P(F) - P(A.C')P(F) \qquad \textbf{A1-5}$$

Expanding using the rules above, we get

$$P(Z) = P(A)P(C)' + P(A)P(B) + P(A)P(C) - P(A)P(B)P(C) - P(A)P(B)P(C)' \qquad \textbf{A1-6}$$

The P_{high} values for each node in the circuit are given in Column 2 of Table A1-1. In addition to SET pulse generation, the SET pulse must propagate to an output node of the circuit. If a node signal is blocked from reaching a circuit output for a large percentage of the vectors (strong logic masking), hardening it will not improve SE error rate significantly. Identification of nodes most likely to be struck and the resulting SET pulse most likely to reach a circuit output should be used as a criterion for efficient circuit hardening. For a given set of primary inputs to a circuit, P_{high} values for each node can be used to calculate the probability for a transient to propagate to a circuit output. The probability of a signal propagating from a circuit gate output node to an output of the circuit is defined as the Logical Masking Metric (LMM).

$$Logical\ \ Masking\ \ Metric \quad = \quad \prod_{j=1}^{m}\prod_{k=1}^{l} Pe_k \qquad \textbf{A1-7}$$

where Pe_k is the enabling value probability for input k of each gate j, not lying on the path from input to output. Transients on a given input will appear on the output if the other inputs to the gate are at enabling values. For AND, NAND and XNOR gates this value is 1. For OR, NOR and XOR gates this value is 0. Consider a transient at node E to output Z of the circuit in Figure A1-2. The Logical Masking Metric for E is:

$$LMM\ (E) = (1-P(D))(1-P(H)) \qquad \textbf{A1-8}$$

The LMM for each node in Figure A1-2 is included in Column 4 of Table A1-1. For larger circuits where there are multiple paths from a gate output to the circuit outputs, the path with least masking probability to a single output is considered.

Once the gates having the highest probability of generating transients of each kind are identified, they must be compared based **upon** their propagation probabilities. This is done by taking the product of P_{high} and LLM. The same is done for P_{low} values. LMM values for a given node will remain the same for n-hits and p-hits. LMM values for a given node will remain the same for n-hits and p-hits. The Hardening Metric (HM) thus indicates the gates that produce one kind of transient more than the other and have the highest propagation probability.

$$HM = \begin{cases} P_{high} * LMM & P_{high} \geq 0.5 \\ \left(1 - P_{high}\right) * LMM & P_{high} < 0.5 \end{cases} \qquad \textbf{A1-9}$$

Based on their hardening metric, the gates are arranged in descending order for hardening consideration. It should be noted that increasing the size of a transistor increases the probability of a hit. So if the size of the restoring transistor is increased, the probability for a hit on that transistor also increases. But a high (low) value of P_{high} (P_{low}) for a given node implies that the probability for the restoring transistor to be OFF is low. As a result, any increase in sensitive area for the restoring transistor will have very small effect on the overall error rate.

Table A1-1 Node Signal Probabilities and LMM

Node	P_{high}	P_{low}	LMM	Hardening Metric From Equation 9
A	0.50	0.50	---	---
B	0.50	0.50	---	---
C	0.50	0.50	---	---
D	0.25	0.75	0.56	0.42
E	0.25	0.75	0.56	0.42
F	0.48	0.52	0.75	0.39
G	0.50	0.50	0.26	0.13
H	0.25	0.75	0.52	0.39
Z	0.50	0.50	1	0.50

Based on the above analysis, the signal probabilities have been calculated for the International Symposium on Circuits and Systems (ISCAS) benchmark circuits [Hans-99] using a PERL script operating on a Verilog description of the circuits. Inputs were assumed uncorrelated and were assigned P_{high} = 0.5. This is a reasonable approximation for most logic signals. However the designer can use appropriate probabilities for specific applications for the given circuit by simulating the input load for a random set of vectors. The pseudo code is summarized in Table A1-2

Table A1-2 Pseudocode

```
Start:  Describe circuit in Structural Verilog/VHDL.
  compute $P_{high}$, LMM
for ($P_{high} > 0.5$)
{
HM= Phigh*LMM
else HM= $(1-P_{high})$*LMM
}
Arrange nodes in descending order by HM values.
  Compute circuit area and power
  Re-size selected nodes based on HM
for (delay > delay constraint)
 {
   remove least vulnerable nodes on maximum
   re-compute delay
  }
   re-compute area, power
end
```

Table A1-3 shows the total number of nodes in the circuit and the number of nodes at various levels of P_{high}. It is evident that only a small percentage of gates have probabilities of being either high or low, as indicated by values close to 1 or 0, respectively. For each of the circuits, the top 10, 20, and 30 % of nodes on the HM list were hardened by increasing the restoring transistor by a factor of 2. Based on Figure A1-1, a 2X increase in restoring transistor size results in an average 35% decrease in SET pulse-width for charge deposition spectrum considered. Since circuit SER is directly related to the latching probability of SET pulse-widths, hardening the most sensitive nodes would reduce the SER significantly. Table A1-4 shows the area and power overhead for each circuit for achieving this improvement. The algorithm can be summarized using the flowchart shown in Figure A1-3.

Table A1-3 Circuit Node Signal Probability Distribution

Circuit	Gates	Number of Nodes in circuit with $P_{high} > 0.7$ and $P_{high} < 0.3$					
		>0.9	>0.8	>0.7	<0.3	<0.2	<0.1
c432	160	14	24	48	34	19	7
c499	546	19	68	172	126	70	22
c880	383	14	56	81	107	60	9
c1908	880	27	125	330	228	103	32
c2670	1193	42	117	153	136	84	50
c3540	1669	37	221	325	380	178	21
c5315	2406	54	307	519	395	269	77
c6288	2406	90	365	424	608	331	101
c7552	3512	123	367	675	773	402	88

Flowchart for algorithm implementation

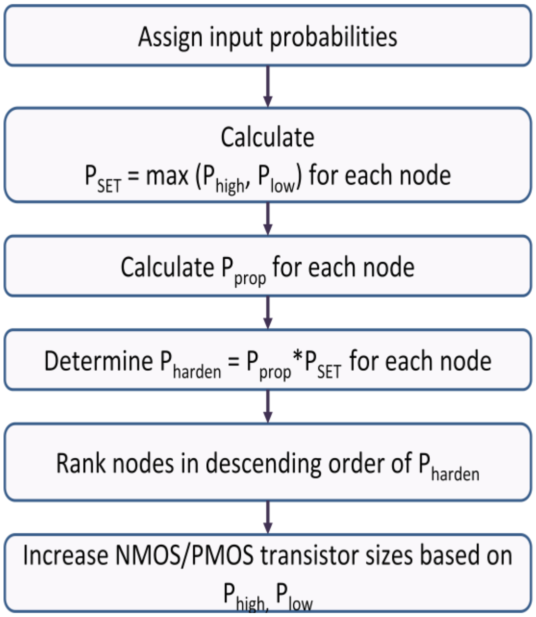

Assign input probabilities
Calculate
P_{SET} = max (P_{high}, P_{low}) for each node
Calculate P_{prop} for each node
Determine P_{harden} = P_{prop}*P_{SET} for each node
Rank nodes in descending order of P_{harden}
Increase NMOS/PMOS transistor sizes based on
$P_{high,}$ P_{low}

Figure A1-3 Algorithm flowchart for node vulnerability estimation and hardening selectively

3. Average Pulse-Width Reduction Using Monte Carlo Simulations

A Monte Carlo simulation was set up to validate the hypothesis that hardening certain nodes selectively for transient pulse-width reduction results in a lower logic SER. The results presented below are for the ISCAS Benchmark c880 8-bit ALU. The circuit was synthesized with minimum sized standard cell libraries built from the IBM CMOS9sf 90 nm PDK. It was then characterized for area, power and delay. Another implementation of the same circuit was synthesized by applying the algorithm and resizing 10% of the candidate gates with the appropriate cells.

Two kinds of Monte Carlo simulations were set-up. These involved random fault injections on circuit nodes with random input vectors. This is classified as Non-Stratified sampling because the sample set is uniformly sampled without weighting the members.

The second involved stratified or weighted sampling to choose the nodes that were resized using the algorithm to be struck more often and then applying random input vectors. This is termed as Stratified sampling.

I. Non Stratified Sampling

In these simulations, random faults were injected at nodes in the circuit using bias dependent piece-wise linear current sources. The piecewise bias-dependent model has been proven to be more accurate compared to the double exponential [Kaup-09]. It also reflects the effects of LET on current shape [Dasg-07]. The resultant voltage transients propagated to the outputs where pulse widths were monitored and histogramed. The results of these simulations are illustrated for ISCAS Benchmark c880 8-bit ALU circuit. 10% of the nodes were hardened based on the algorithm explained earlier. The same procedure was then repeated on the circuit with resized gates was then simulated for the same set of random inputs and faults were injected at the same nodes as in the previous case. The transient pulse-widths following these injections were again monitored and histogramed. The result of the random simulations on the ISCAS Benchmark c880 8-bit ALU circuit with and without resized gates is shown in Figure A1-4. For both the distributions, the 3*sigma values encompass 99% of the area under the curve, hence the distribution can be assumed to be normal.

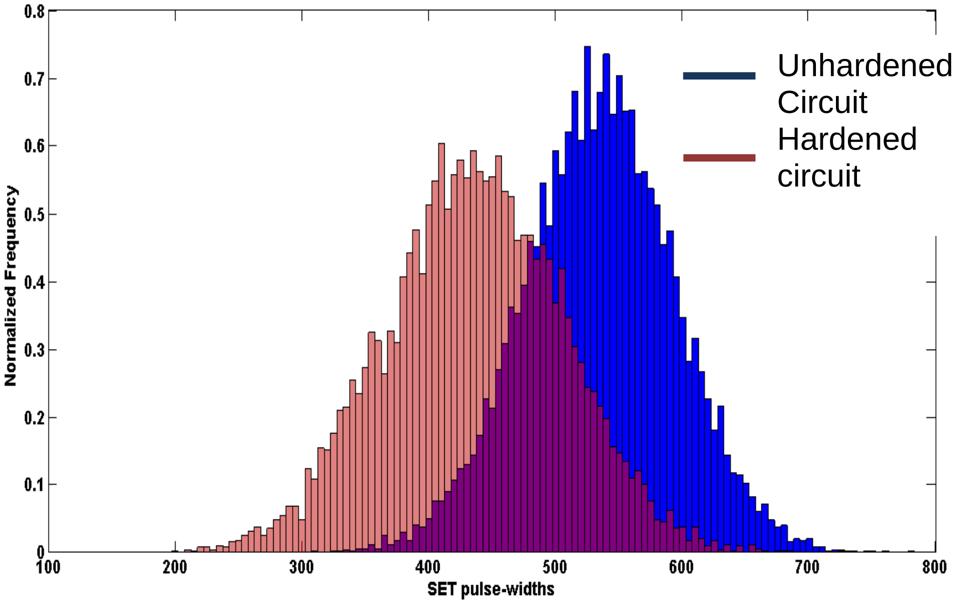


Figure A1-4 Distribution of output SET pulse-widths from random Monte Carlo simulations for an 8-bit ALU before and after resizing.

Assuming the standard normal variate Z for a normal distribution, the mean (for 95% confidence limits) lies between

= -1.96 < Z < 1.96

= -1.96 < X-μ/ σ* < 1.96

The observed mean of the distribution for unhardened circuit in Figure A1-2 is 534 ps and the standard deviation is 70. The total number of simulation runs (or SET pulses monitored) were 10,000. The standard error σ* is therefore σ/√n = 70/√10000 = 0.7. The normalized estimate mean of the standard normal variate lies between

= -1.96 < Z < 1.96

= -1.96 < (X - 534)/0.7 < 1.96

= -1.4 < X-534 <1.4

Therefore the estimated mean is between (**532.6, 535.4**) for the distribution at 95% confidence limits. For the hardened or resized version of the same circuit, the observed mean of the distribution is 436 ps and the standard deviation is 60. The total number of observations were again limited to 10,000. The standard error is therefore σ/sqrt(n) = 60/sqrt (10000) = 0.6. The observed mean is therefore

$= -1.96 < Z < 1.96$

$= -1.96 < (X - 436)/0.6 < 1.96$

$= -1.2 < X - 436 < 1.2$

Therefore the observed mean is between (**434.8, 437.2**) for the distribution at 95 % confidence limits.

Clearly the average pulse-width has reduced. At the cost of hardening only 10% of the nodes a visible reduction in the average pulse-width is observed. However, the case where nodes are sampled based on the probability of them being struck given that their sensitive cross-sections would be different, is also important. This is studied in the section.

II. Stratified Or Weighted Sampling

In the second experiment, stratified Monte-Carlo simulations were carried out. The nodes that were resized had a 2X probability of being chosen compared to the nodes with no resizing. This is so because due to increased sizes their cross-section to radiation particle strikes increases. So the Cumulative Distribution from which random numbers were generated reflected weighted probabilities of nodes being selected for fault injection. The same test vectors were applied to both the simulation sets, i.e., to the golden copy (original unhardened version) and the resized version. However unlike the previous comparison, the same sets of nodes were not selected because of weighted probabilities and different cumulative distributions chosen to generate random numbers. The resultant distribution of SET pulse-widths after this experiment is shown in Figure A1-5. In this case the average pulse-widths reduce by about 20%, but the interesting fact is that the distribution of pulse-widths after resizing the gates is wider. A possible reason could be that, as a result of stratified sampling, the transients at the nodes which are struck more often are longer and thus tend to increase the standard deviation of the distribution. With stratified sampling too, the average pulse-widths reduce by about 25%, which compares favorably with the ideal reduction of about 35% as seen in Figure A1-1 for a range of charge deposition values. Since stratified sampling includes the effects of increased cross-section as a result of resizing, the reduction in pulse-widths should directly translate into reduced latching probabilities.

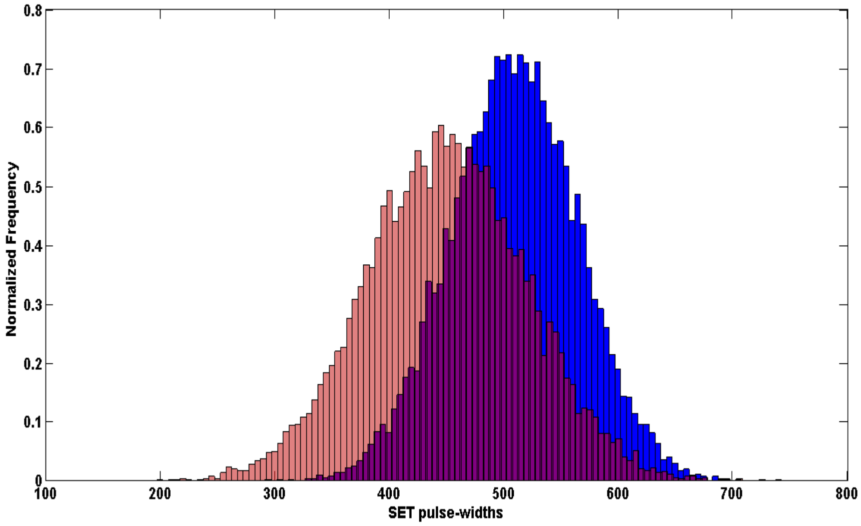


Figure A1-5 Distribution of output SET pulse-widths from non-stratified sampling on unhardened circuit and stratified Monte Carlo simulations for an 8-bit ALU after resizing

4. Circuit Overhead

To determine the performance overheads in terms of area and power the ISCAS benchmark circuits were synthesized using the Oklahoma State University (OSU) 45-nm Process Development Kit (PDK). The area and power overheads were calculated using Synopsis Design Compiler and are shown in Table A1-4. Since CMOS is a ratio-less logic, the effect of resizing nMOS and pMOS transistors independently does not result in a large delay penalty [Amus-07, West-94]. The average overheads resulting from increasing transistor widths is given in Figure A1-6.

Table A1-4 Percentage overheads in terms of area and power for the 2x hardened circuits

Circuit	Percentage overhead due to hardening					
	10% of nodes		20% of nodes		30% of nodes	
	Area	Power	Area	Power	Area	Power
c432	5	3	10	4	17	12
c499	4	2	8	3	22	10
c880	4	5	5	7	13	14
c1908	10	7	14	9	22	11
c2670	4	5	12	9	12	10
c3540	10	9	9	7	16	14
c5315	4	8	9	11	12	8
c6288	5	5	10	7	19	9
c7552	9	6	11	8	27	13

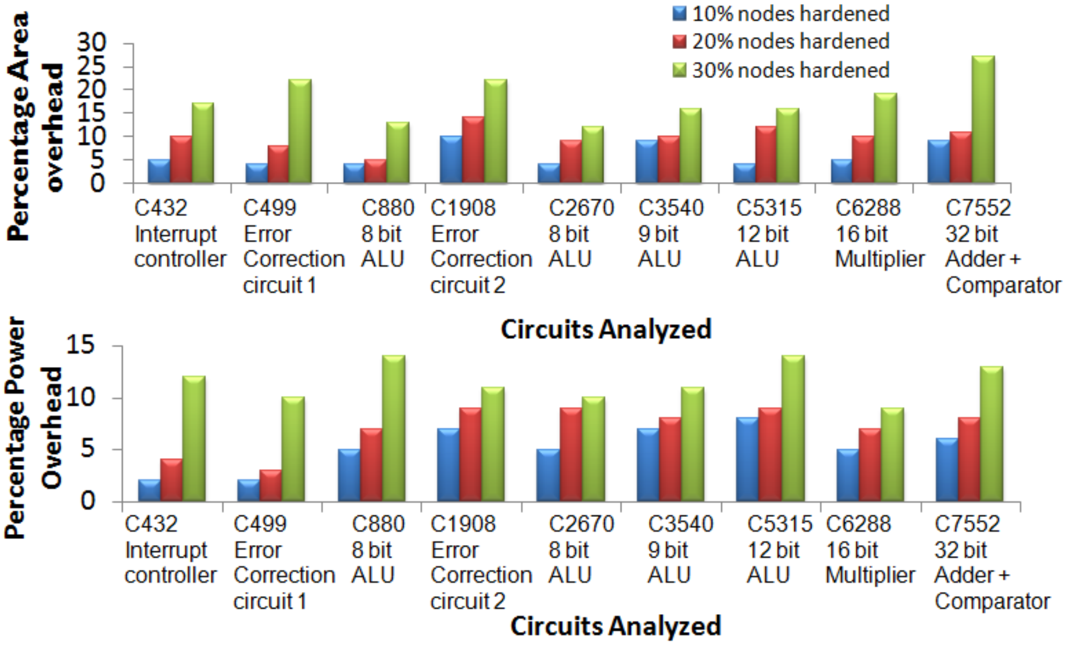


Figure A1-6 Average area and power overheads due to increasing transistor widths.

By accounting for the nodes that predominantly produce transients from either n-hits or p-hits and have a high probability of transients propagating to the output, a computationally efficient algorithm has been proposed to selectively harden a circuit and serve as an alternative to fault injection and simulation studies. Since the circuit SER largely depends on the nodes where transient are generated and their propagation probability, hardening those nodes would lead to significant reduction in the circuit SER. Simulation results for ISCAS benchmark circuits show area overhead to range between 12% to 27% and power overhead to range between 8% to 14% when 30% of total nodes were hardened. The delay overhead was less than 8%. Thus, this technique is most useful when applied to harden circuits with tight area, power or delay constraints.

Judging from the power overheads in this work, hardening even 20 % of the nodes can lead about 20% overhead in the power which is unacceptable as far as modern circuit designs are concerned. In [Zhou-06] the authors adopted the approach to characterize the sensitivity of nodes in the circuits and increase transistor sizes uniformly and therefore reduce SET pulse-widths. Similar approaches to characterize the most sensitive nodes were employed by [Nieu-06] [Almu-08] [Pagl-12]. The key characteristic of all these approaches is that the reduction in logic soft error rate comes at the expense of unavoidable power overheads. Limited data is available on the exact power overhead of these techniques but others have reported the power overheads from hardening selectively.

Appendix B : Impact of Supply Voltage on Logic Soft Error Rate

In this section, the impact of voltage on single event transients and their likelihood of being latched is discussed. The key results suggest that the although voltage increases the SET pulse-width, the flip-flop setup-and-hold time also increases which impacts the probability of latching the SET.

1. Introduction

The objective of modern high-speed circuit designs is to maximize performance. However, increasing emphasis is being placed on minimizing power consumption while maximizing performance [Keat-07], [Venk-05]. Designers therefore routinely employ aggressive power management schemes, such as dynamic voltage and frequency scaling (DVFS) to save power without sacrificing performance. As a result, ICs operate under a variety of different voltage and frequency conditions that can affect the single-event error rate or soft error rate (SER) of circuits. In this work, the impact of voltage and frequency on the flip-flop and logic soft error rate for state-of-the-art, 28-nm, circuits is experimentally characterized. The frequency threshold beyond which logic SER exceeds flip-flop and latch SER (encircled in Figure A2-1) is also identified [Buch-97]. Identifying this threshold or cross-over frequency as a function of supply voltage will help designers to develop effective hardening strategies for logic and/or flip-flops and allow for a much better trade-off between performance, power, and soft error reliability.

It is well known that decreasing the supply voltage results in higher latch SER, while frequency has a much greater impact on logic SER than does supply voltage [Hazu-00].

In this work, experimental results suggest that the alpha particle logic SER is relatively unaffected by variations in the supply voltage. Increasing the supply voltage by even 50 mV above the nominal supply voltage can reduce the frequency threshold at which logic soft errors exceed flip-flop soft errors from 800 MHz to 300 MHz. The experimentally observed results are qualitatively explained using circuit-level simulations. Simulations are also used to compare the voltage dependence of logic SER under alpha particle and heavy-ion irradiations.

This chapter is organized as follows: in Section II, the test circuits and alpha particle experiments are described. The experimental test results for soft-error measurement under different voltage and frequency conditions are discussed in Section III. In Section IV, the impact of voltage on the combinational logic cross section is explained through simulations of ion strikes by alpha particles. Impact of heavy-ion irradiation on the voltage dependence of logic SER is also discussed. In Section V, the implications of operating circuits at different voltage and frequency conditions and the choice of hardening strategies are discussed.

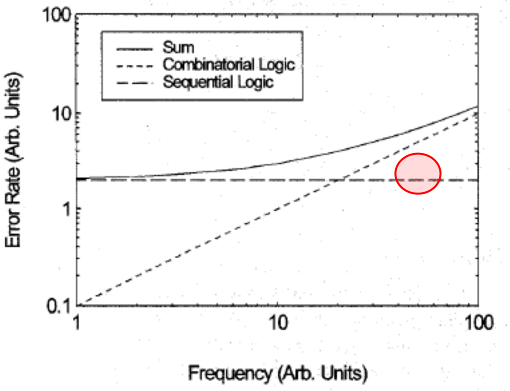


Figure A2-1 Logic errors increase with frequency. [Buch-97] defined a frequency threshold (encircled above) at which logic errors would exceed flip-flop errors. Beyond such a threshold, logic SER exceeds flip-flop SER.

2. Test Circuit Description & Experiments

All the 28-nm circuits were irradiated with alpha particles to investigate the voltage and frequency dependence of combinational logic and flip-flop circuits. The test circuits and the experimental set-up is explained in the following sections.

2.1 Circuit Description

The approach used for this work to measure the logic error cross section is based on the Combinational Circuit for Radiation Effects Self-Test (C-CREST) technique [Ahlb-09]. The basic block diagram for this technique is shown in Figure A2-2. The Circuit-Under-Test (CUT) consists of a shift register design with logic circuits interleaved with flip-flops as shown in Figure A2-3. One flip-flop circuit along with the associated logic circuit comprises a single stage. The C-CREST design consists of 2,056 of such stages to improve the error statistics. The error detection circuit compares the correct data with the output of the CUT. In the absence of errors, the patterns from the output of the CUT and the data source are identical, and no errors are recorded. If errors occur, then a counter records the total number of errors observed. Error detection circuits are protected against single event errors using Triple Modular Redundancy (TMR).

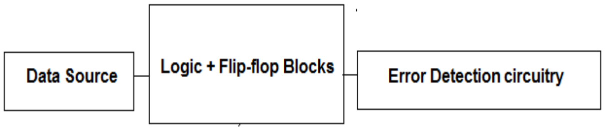


Figure A2-2 Basic structure used to evaluate flip-flop and combinational logic cross sections.

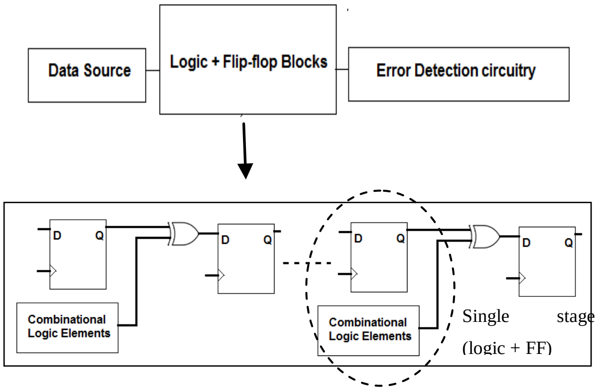


Figure A2-3 The Circuit Under Test consists of flip-flops and logic blocks. Two different such structures were tested, one with inverters in the logic block and other with a 4-bit comparator.

Two variants of the C-CREST design were fabricated. For both the variants, 2,056 stages were used. The flip-flop design used for both the variants was a conventional transmission gate D flip-flop circuit. For the first C-CREST design, the logic circuit consisted of a block of 72 inverter gates (12 chains of six inverters each) OR'ed together using 11 OR gates. Each OR gate consists of 1 NOR gate + 1 inverter. Thus there were 83 NOT gates and 11 NOR gates in total. The second C-CREST design consisted of a four-bit 'greater than or less than' comparator. The four-bit comparator compares two four-bit numbers, A and B. The output of the comparator is a logic '1' if A > B. The four-bit comparator was chosen because the logic depth for this circuit is close to that of modern circuits [ARM-11], [Inte-10].

The drawn length of all the transistors was 30 nm, and the minimum transistor width used in the designs was 100 nm. The threshold voltage of the NMOS (PMOS) devices was about 250 mV (-270 mV). The area of the transistor drain regions was (W x 75) nm^2 where W is the width of the transistor. All the gates were sized to have their rise and fall time equal to that of a minimum sized inverter (W_N = 100 nm and W_P = 250 nm). The dimensions of the NMOS (PMOS) transistors used in the NOT, NAND, and NOR gates were 100 nm (250 nm), 200 nm (250 nm), 100 nm (500 nm) respectively. Table I provides more information about the gate counts of the two logic circuits.

TABLE I.
Number of gates, transistors, and transistor total area for different circuit types.

Circuit type	Inverter	Comparator
Total # of gates	94	46
Type of gates	83 NOT 11 NOR	26 NOT 12 2-input NAND 2 3-input NAND 6 2-input NOR
Total # of transistors	210	136

Test Details

The circuits were irradiated with 5.5 MeV alpha particles from an Americium-241 source with an activity of 10 µCi, at room temperature. The flux was 430 particles/mm^2-s. The alpha source was placed at a distance of 1 mm from the die during testing. The size of the alpha source was 1 cm^2, and the size of the die was 3 mm x 3 mm. To account for both inter-die and experimental variability, measurements were repeated 16 times at each frequency and logic input value. Testing was conducted in accordance with JEDEC specifications [JEDE-06]. The operating voltage was varied from 0.8 V to 0.9 V in 50 mV steps, and the operating frequency of the CUT was varied up to 500 MHz using an on-chip low-noise Phase Locked Loop (PLL) circuit. The PLL was also implemented using TMR.

2.2 Experimentally Measuring Logic Cross section

The logic and flip-flop cross sections were measured separately. To determine the flip-flop cross section alone, the shift register chains were operated at very low frequency (e.g. 10 MHz) and at fixed voltage. At such low frequencies, the logic soft error contribution is orders of magnitude lower than flip-flop soft errors and can be neglected [Buch-97]. The total soft errors observed from the shift register chain normalized by the fluence and the number of flip-flops in the chain yields the flip-flop cross section. Several trials were performed across four different dies at each operating voltage and frequency to minimize effects of experimental as well as die-to-die variations. The error bars represent the standard error of measurement at each data point. Each data point corresponds to 4 measurements each from 4 dies. The experiment was repeated 16 times for each data point, the standard error is σ/√n, where *σ* is the standard deviation of the sample and *n* is the number of times the experiments was repeated (n=16). This procedure was then repeated for different voltage values. Following this procedure, the shift register chains were operated at higher frequencies to record logic as well as flip-flop upsets. The average value of the flip-flop cross section was subtracted from the total cross section at higher frequencies to yield the logic cross section. The variation in the measured flip-flop cross section was very small and thus its variation was not propagated to the logic measurements. While the frequency dependence of flip-flops has been reported earlier, subtracting the low frequency flip-flop cross section from the total cross section to obtain the logic cross section is a reasonable assumption. The frequency dependence of flip-flops arises from the fact that one of the latch stages is always transparent while the other stage holds data. If the slave stage is transparent, the master stage can latch transients from the slave latch portion of the flip-flop [Jaga-12]. In this work, the soft error contribution of the transparent latches has been assumed to be part of the combinational logic interfaced to the flip-flop, especially because the size of the logic is much larger compared to the latch itself. This assumption is also true for conventional circuit designs where substantial computational logic is present between flip-flop stages. Besides, any errors occurring on previous (upstream) latch stages will have a high probability of being temporally masked by the logic between flip-flop stages [Seif-04]. Thus, although the flip-flop cross section was measured only at 10 MHz, it is plotted as being independent of frequency for the whole frequency spectrum, in this work. The expressions used to calculate the flip-flop and logic cross section are as follows:

$$Total\ \ Cross\ Section = \frac{Total Number of\ Errors}{Total Number of\ stages \times Fluence} \qquad (1)$$

$$\begin{gathered} Logic\, Cross\ Section\, per\, Stage\ = \\ (Total\, Cross Section) - (Flip\ Flop\ \ Cross\ Section)_{@10MHz} \qquad (2) \end{gathered}$$

2.3 Experimental Results

In this section, the results of alpha particle irradiation of the 28-nm circuits are reported. Although several input conditions were tested, detailed explanation of results is provided for inverter inputs at logic ‘0’ and the comparator inputs at A = ‘0000’ and B = ‘0000’. Other test conditions showed similar trends as far as alpha particle irradiations were concerned.

2.4 Impact of voltage and frequency on combinational and flip-flop SER

Figure A2-4 shows the alpha particle cross section of the FF and the comparator circuit as a function of frequency with supply voltage of 0.9 V. The input to the comparator was A = ‘0000’ and B = ‘0000’. The flip-flop cross section has been plotted as being independent of frequency based on the reasoning provided in the previous section.

The low frequency (10 MHz) cross section of the flip-flop, shown as a dashed line in Figure A2-4, was 1x10^{-11} cm^2. The logic cross section, on the other hand, increases linearly as a function of frequency. The frequency threshold at which the logic cross section exceeds the flip-flop cross section is approximately 300 MHz. In other words, beyond 300 MHz, the number of logic soft errors would exceed that for flip-flop soft errors. The two key factors that influence the cross-over frequency are the slope of the logic cross section and the flip-flop cross section itself. In order to study the impact of supply voltage on both of these factors, the supply voltage was varied from 0.8 V to 0.9 V in steps of 50 mV.

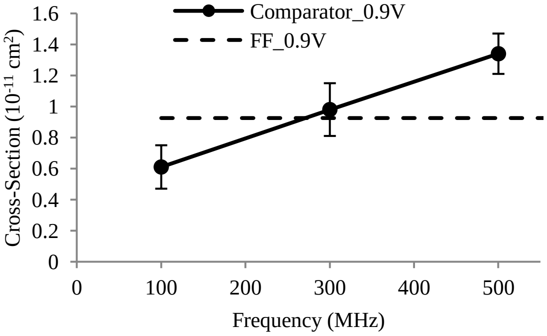


Figure A2-4 Experimental cross section for comparator circuit with comparator inputs A = ‘0000’ and B = ‘0000’ and 0.9 V. The flip-flop cross section is ~1×10^{-11} cm^2. The logic cross section exceeds the flip-flop cross section at 300 MHz.

The impact of supply voltage on the FF and logic cross sections is plotted in Figure A2-5. As the supply voltage is reduced, the flip-flop cross section increases significantly. In fact the cross section increases by as much as 2.5 X when the voltage is varied from 0.9 V to 0.8 V. The reason is that as the supply voltage is reduced, the critical charge decreases, and the cross section is exponentially dependent on the critical charge [Hazu-00]. On the other hand, the slope of the logic cross section is not affected significantly by a change in supply voltage. The cross-over or threshold frequency at which logic soft errors exceed flip-flop soft errors can be estimated by extrapolating the logic cross section.

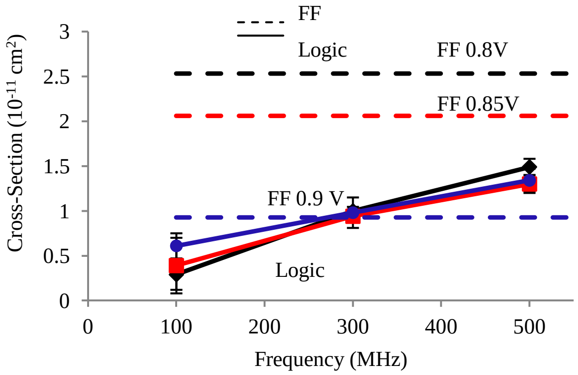


Figure A2-5 Flip-flop cross section increases with decrease in voltage. The logic cross section for input A = B = '0000' is nearly independent of supply voltage variation.

There are several important implications of these results. Firstly, as Figure A2-4 suggests the frequency threshold at which the logic cross section exceeds flip-flop cross section is in the neighborhood of 300 MHz. Clearly, for average sized modern high-speed circuits that are capable of multi-GHz operation, the cross-over frequency of 300 MHz can be easily exceeded by the operating frequency. At such frequencies, hardening flip-flops alone will not result in significant reduction in the total SER. Secondly, the supply voltage impacts the flip-flop's SER more than it affects the logic circuit's SER for alpha particle irradiations, as a result of which, the cross-over frequency increases with a decrease in supply voltage. Figure A2-6 plots the cross-over frequency as a function of voltage. In other words, as the circuit operates at lower voltages, flip-flop soft errors are likely to contribute to a majority of the total chip-level soft errors. On the other hand, to ensure high-speed operation, a higher voltage is required because the delay is inversely proportion to the supply voltage. The higher the supply voltage, the lower is the gate delay and the higher the maximum operating frequency. Thus as the supply voltage is increased, the cross-over frequency decreases, which means that logic soft errors could exceed flip-flop soft errors at much lower frequencies.

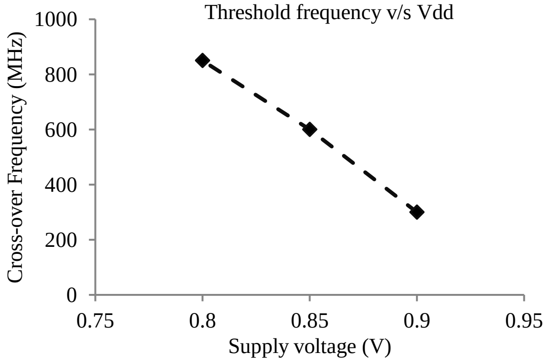


Figure A2-6 Logic cross section was measured up to 500 MHz. The logic cross section extrapolated to 1 GHz shows that the threshold frequency at which logic errors exceed flip-flop errors decreases as voltage is increased. The cross-over frequency is 300 MHz, 600 MHz, and 800 MHz for 0.9, 0.85, and 0.8 V operation.

A. Impact of input conditions on the cross-over frequency

Along with the operating voltage and frequency, the inputs to the logic circuits were also varied. Figure A2-7 shows the range of cross-over frequencies for different inputs conditions for the comparator as well as the inverter circuit. Different input combinations result in different logical masking factors based on the expression used to model the logic cross section [Alex-11], [Nguy-03], [Lide-94]

$$\text{Logic Cross Section}_{\text{estimated}} = A \cdot EM \cdot LM \cdot TM \qquad (3)$$

where *A* is the sensitive drain area, *EM, LM* and *TM* are the electrical, logical and temporal masking factors respectively. Consequently, the sensitive area varies with change in the input conditions. As Equation 3 suggests, the logical masking factor, *LM* influences the cross section and thus the threshold frequency.

As far as the impact of supply voltage is concerned, for all input conditions of the comparator as well as for the inverter circuit, the threshold frequency decreases with increasing voltage. At a particular input voltage, such as 0.8 V in Figure A2-7, the threshold frequencies for the comparator for two different input combinations are 2.2 GHz and 1 GHz. In the case where the threshold frequency is 2.2 GHz, a large number of transients are masked due to logical masking due to inputs applied. As a result the number of gates sensitive to transients is small compared to total number of gates in the circuit. Thus, the condition in which very few errors are recorded is referred to as the maximum logical masking case in Figure A2-7. On the other hand, certain inputs result in very little logical masking and the logic SER exceeds the flip-flop SER at as low as 1 GHz. Thus, the condition in which very few errors are logically masked is referred to as the minimum logical masking case in Figure A2-7. As the voltage increases to 0.9 V, the threshold frequency reduces considerably to the 300-700 MHz range as shown encircled in Figure A2-7. Indeed, the decrease in threshold frequency at higher voltages is a result of a decrease in the flip-flop cross section. However, the implication is that logic SER from even a few gates are able to exceed the flip-flop SER in the 300-700 MHz range.

When the comparator circuit was analyzed for different input combinations, maximum logical masking resulted in six gates being sensitive to transients. Yet, the cross-over frequency is only 700 MHz, suggesting that at higher supply voltages, the effects of logical masking are diminished. Thus at higher supply voltages, because the flip-flop cross section is much lower, logic SER dominates at much lower frequencies than at nominal supply voltage. In the following section, the reason for the relative supply voltage independence for logic SER is explained through simulations. The simple models utilized to explain the supply voltage dependence are also extended to evaluate the supply voltage of logic SER under heavy-ion irradiation.

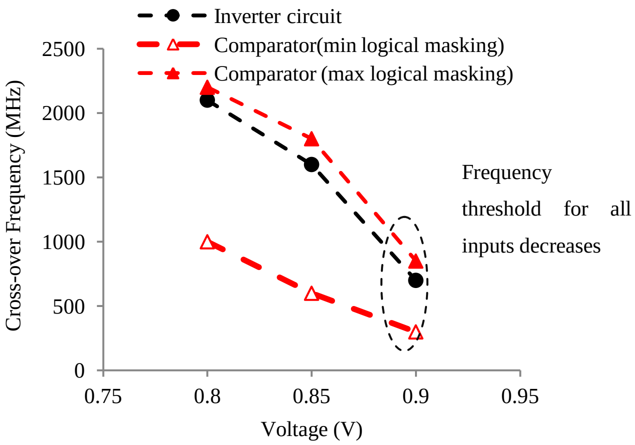


Figure A2-7 The cross-over frequency decreases as the voltage is increased for both comparator and inverter circuits. It is as low as 300-700 MHz at 0.9 V.

2.5 Simulations To Explain The Voltage Dependence of Logic SER

The flip-flop cross section increases exponentially with decreasing voltage [Hazu-00]. As Equation 3 suggests, the logic cross section depends on the sensitive drain area as well as electrical, logical, and temporal masking factors. Electrical masking is a measure of transient attenuation or broadening and is dependent on supply voltage. However, the temporal masking, which mainly depends on the single-event transient (SET) pulse-width and setup-and-hold (SH) time of flip-flop, is directly influenced by a change in the supply voltage. The temporal masking factor or latching probability of transients can be expressed as [Nguy-03], [Shiv-02]

$$Temporal\,Masking\,Factor = \frac{T_{SET} - T_{SH}}{T_{CLK}} \tag{4}$$

where T_{SET} , T_{SH} and T_{CLK} are the SET pulse-width, setup-and hold time, and the clock period. Both the SET pulse-width and the setup-and-hold time are functions of the supply voltage. In this work, first order models have been used to explain the impact of supply voltage on these two parameters that influence the temporal masking factor and hence the logic SER. Figure A2-8 shows the circuit used to evaluate the effects of voltage on single-event transients. Ion strikes on complex gates, such as NAND and NOR gates, could result in transient pulse-widths that are different from inverters but the voltage dependence of the transient pulse-widths is likely to be similar. Hence, in the following sections, analysis of the voltage dependence of inverter transient pulse-widths is used to explain some of the experimental results. Ion strikes on PMOS *MP1* shown in Figure A2-8 were simulated using a bias-dependent current source for different values of charge deposited and operating voltage [Kaup-09]. The restoring current of transistor *MN1* was monitored. For a transient to propagate unattenuated through the logic chain, it must exceed the rise and fall time of the succeeding gate [Mass-08]. For this condition to be satisfied, when a transient occurs at node *V1*, the amplitude of the transient must exceed V_{dd}-V_{tp}. For simplicity, we assume it must exceed V_{dd}-V_t (where V_{tp}=V_{tn}=V_t). The threshold voltages of the NMOS and PMOS transistors used for simulations were similar. This condition ensures that the PMOS transistor *MP2* turns on, and a rail-to-rail excursion occurs at node *V2*. The transient excursion above V_{dd}-V_t also forces restoring NMOS transistor *MN1* into saturation. Thus the condition that must be satisfied for a rail-to-rail transient at *V2*, is that the NMOS transistor *MN1* must go into saturation. The saturation current of *MN1* helps to restore node *V1*. As the transistor quickly moves into the saturation region following an ion strike, the transistor can be assumed to be in the saturation region for the duration of the strike [Dasg-07].

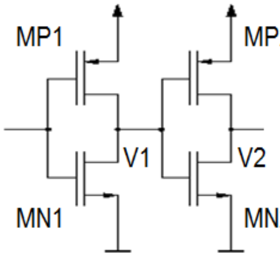


Figure A2-8 Inverter circuit used to study the impact of voltage on the logic cross section. Transistor MP1 was struck. SET pulse-width at node V1 and NMOS restoring drive were monitored for different values of supply voltage and charge deposition.

The transient pulse-width at the node *V1* is inversely proportional to the drive current. Ideally, the transistor drive current is proportional to only the gate voltage while operating in saturation region, thus the transient pulse-width is inversely proportional to the supply voltage as explained by the following Equations.

$$IDsat_{MN1}\,\alpha\,(V_{GS}-V_T)^2 \qquad (5)$$

$$T_{SET}\,\alpha\,\frac{1}{(V_{GS}-V_T)^2}\,\alpha\,\frac{1}{V^2} \qquad (6)$$

SET pulse-widths as a function of the supply voltage for different charge deposition values are shown in Figure A2-9. Indeed, the simulation results follow the *1/V*^{2} relationship explained previously for the different values of charge deposition.

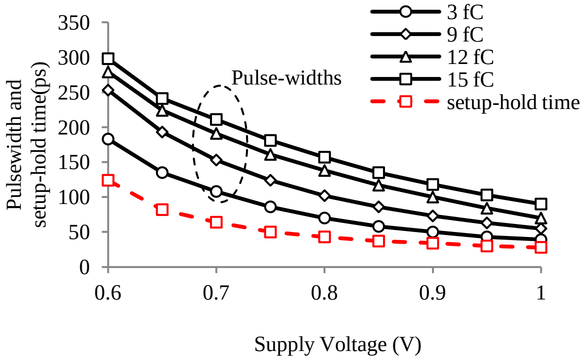


Figure A2-9 Simulation results for SET pulse-widths as a function of different voltages and setup-and-hold time for the flip-flop.

Similarly, the voltage dependence of the setup-and-hold time of the D flip-flop used in the design was evaluated. The impact of supply voltage on the setup-and-hold time is also plotted in Figure A2-9. In this case too, the *1/V*^{2} dependence on supply voltage is evident. However, it is not as strong as the SET pulse-width dependence on supply voltage because the individual transistors do not remain in saturation for as long as it does in the case of SETs. Unlike SETs, where the input voltage is pinned, the nodal voltages in a feedback structure are not pinned. Thus during switching activity, the transistors in a FF spend comparatively less time in saturation than those in logic circuit during a SET.

Least squares fit to an Equation of the form *K/(V-0.25)*^{2} for the data for SET pulse-widths and setup-&-hold time obtained from simulations was performed using MATLAB. The results are shown in Table II. *K* is the factor that includes the *β* value of the transistor drive current. The R^2 values of all the fits exceeded 0.90, indicating the goodness of fit as also justifying the assumption that the NMOS transistor, *MN1,* is mostly in the saturation region for the duration of the SET.

The temporal masking factor depends on both SET pulse-width and setup-&-hold time as suggested by Equation (4). The difference of the SET pulse-width and setup-&-hold time is plotted in Figure A2-10. Both, the SET pulse-width and the setup-&-hold time have an inverse square relationship with supply voltage. Hence their difference also follows an inverse square relationship. However, the slope of the curve decreases as the charge deposited decreases. In other words, for higher values of charge deposition, the SET pulse-width varies at a rate much faster than the setup-&-hold time, as a function of supply voltage. As Table II shows, the slope of the difference of the SET pulse-width and setup-&-hold time increases as the charge deposited increases. For 5.5 MeV alpha particles, the charge collection is small (i.e., in the 5-10 fC range) [Dupo-02]. As a result, the increase in SET pulse-width as a function of decreasing voltage is not as dramatic compared to the setup-&-hold time. In effect, the two terms in Equation (4) compensate each other; as a result, the net impact on the temporal masking factor, and subsequently the logic SER, is negligible.

TABLE II.
Fits to the SET pulse-width data and SH time data and slope for the difference between SET values and setup-&-hold time

Q_{dep}	T_{SET}	Diff = T_{SET} - T_{SH}	T_{SH}	$\left\lvert \frac{dDiff}{dV} \right\rvert$
3 fC	$\frac{22}{(V-0.25)^2}$	$\frac{8}{(V-0.25)^2}$	$\frac{14}{(V-0.25)^2}$	$\frac{16}{(V-0.25)}$
9 fC	$\frac{31}{(V-0.25)^2}$	$\frac{17}{(V-0.25)^2}$	$\frac{14}{(V-0.25)^2}$	$\frac{34}{(V-0.25)}$
12 fC	$\frac{36}{(V-0.25)^2}$	$\frac{22}{(V-0.25)^2}$	$\frac{14}{(V-0.25)^2}$	$\frac{44}{(V-0.25)}$
15 fC	$\frac{40}{(V-0.25)^2}$	$\frac{36}{(V-0.25)^2}$	$\frac{14}{(V-0.25)^2}$	$\frac{72}{(V-0.25)}$
$T_{SH} = \frac{14}{(V-0.25)^2}$				

In contrast, as the charge deposition increases and approaches the charge deposited by heavy ions, the rate of change of SET pulse-width as function of supply voltage is much higher than that for the setup-&-hold time. Therefore, as the charge deposited is increased, the SET pulse-widths increase at a much faster rate with decreasing supply voltage than the setup-&-hold time, leading to strong supply voltage dependence for logic SER. Gadlage *et. al* observed that the logic error rate increases as the supply voltage is lowered [Gadl-07]. However, for alpha particles, the charge deposition is much smaller, resulting in relative supply voltage independence as far as logic error cross section is concerned.

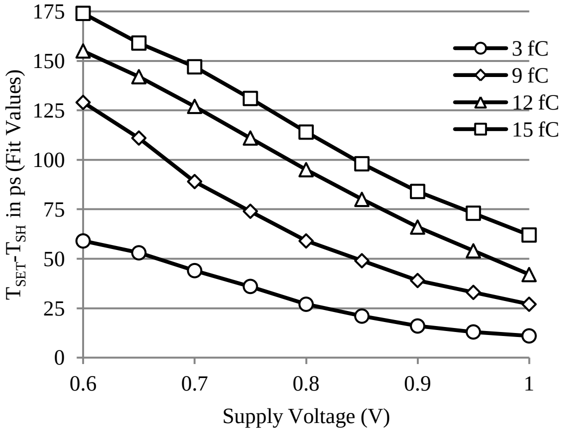


Figure A2-10 Difference of the SET pulse-width value and the SH time for different values of charge deposition. The slope of these curves decreases as the amount of charge deposited decreases.

The other factor that influences the logic cross section is the electrical masking factor. Simulations used to quantify the modulation in the pulse-width as it propagates through 10 uniformly loaded FO4 inverters, suggested that the attenuation was less than 6 ps at the 0.8 V. Hence, electrical masking introduces a small change in the original pulse-width and the impact of electrical masking can be neglected, especially for modern circuits with short-logic depths. Thus, the voltage dependence of the logic cross section is a function of the deposited charge and can be different for different environments. For alpha particles that deposit less charge compared to heavy-ions, the supply voltage dependence of logic SER is weaker than that for flip-flops.

2.6 Conclusion

Logic SER is expected to dominate flip-flop SER for circuits operating in GHz range at the 28-nm technology node. Results presented in this work suggest that as the supply voltage is varied, the frequency at which logic SER could exceed flip-flop SER also varies. The key reason is that for alpha particle exposure, as supply voltage reduces, the critical charge decreases, leading to an increase in flip-flop SER. On the other hand, logic SER shows a comparatively weaker dependence on supply voltage. As a result, at higher supply voltages, the cross-over frequency can be as low as 300 MHz, and logic soft errors from very few gates are enough to exceed the flip-flop soft errors. Also, as the cross-over frequency reduces, the impact of logical masking diminishes. The concept of the cross-over or threshold frequency will help designers to determine both frequency- and voltage-aware mitigation approaches. If the operating frequency is well beyond the cross-over frequency, then designers may be better off employing logic hardening techniques or reliability-aware synthesis [Limb-07]. Moreover, as higher voltages are needed to sustain higher operating frequencies and the cross-over frequency decreases for such operating conditions, it may be most beneficial to harden combinational logic circuits. Conversely, low voltage and low frequency operation ensures that the cross-over frequency is very high, and the total SER is likely to be dominated by latch SER. In such cases flip-flop hardening will bring the most benefit. In contrast to alpha particles, simulations suggest that in the case of heavy-ions, which deposit more charge compared to alpha particles, a much stronger voltage dependence will be observed where the logic cross section also increases significantly as the voltage is decreased.

Appendix C : Frequency Threshold for Combinational Logic Soft Errors

In this section, the frequency threshold at which logic soft errors could exceed flip-flop errors is identified. The frequency threshold is calculated for several benchmark circuits.

1. Introduction

Traditionally, in the terrestrial environment, soft error (SE) concerns have been limited to errors in storage elements, such as SRAMs and latches. However, while the flip-flop (FF) soft error rate (SER) per-bit, is saturating or even decreasing with scaling, higher operating frequencies facilitated by scaling could result in combinational logic soft errors dominating the chip-level SER [Buch-97], [Shiv-02]. This has two important implications. Firstly, efforts to harden flip-flops to achieve lower soft-error rates will be less effective if logic soft errors dominate the total soft error rate. This could affect large ASIC and FPGA designs which implement dense logic structures. Secondly, the amount of control logic for large memory blocks is quite substantial. As memories are operated faster, single-events in the control circuitry will result in huge increases in memory soft error rates. Such single-events in the control logic have been correlated to significant increases in multiple bit upset rates in memory blocks [Nara-10]. Thus, it is necessary to characterize the frequency at which logic soft errors will be a significant contributor to chip-level soft error rates.

The purpose of this study was threefold: 1) compare the 28 nm flip-flop and combinatorial logic SER, especially as a function of frequency; 2) characterize the threshold frequency at which logic soft errors exceed flip-flop soft errors, as a function of number of gates in the circuit and the logic area; 3) propose a simple metric to quantify and determine the threshold frequency for arbitrary circuits. Understanding the first two issues will help in quantifying the effect of logic circuit size (area, gate count) and frequency on the logic soft error rates. This could be used to develop frequency-aware chip-level soft error mitigation schemes. The third objective will provide guidelines to designers about the general range of frequencies at which logic errors exceed flip-flop errors. Alpha-particle exposure results presented in this work suggest that logic soft errors contribute significantly to the total soft error rate for 28-nm technology node and may exceed flip-flop soft errors at frequencies close to 1 GHz. Also, total transistor sensitive area influences the logic SER more than gate count.

2. Test Circuit Description & Experiments

2.2 Circuit Description

The test circuit description and test details are the same as described in earlier chapters. The circuit technique used to measure logic and flip-flop soft errors is similar to the approach based on Combinational Circuit for Radiation Effects Self Test (C-CREST) [Ahlb-09]. Two variants of the basic structure based on a 2056 stage shift register design were built. The two variants differ only in the type of logic circuit used in the CUT. Each CUT consisted of 2056 standard NAND gate based flip-flops, one for each shift register stage. The logic block used in the first C-CREST structure, shown in Figure A3-1, consisted of 72 inverter gates (12 chains of 6 inverters each) OR'ed together using 11 OR gates. The OR gate consisted of 1 NOR gate + 1 inverter. The total gate count was thus 94 (83 inverters, 11 NOR gates). A 4-bit ‘greater than or less than’ comparator, shown in Figure A3-2, was used in the second C-CREST structure because the logic depth in this circuit is similar to that used in modern ASIC designs [ARM-11],[Inte-11],[Gunt-08]. The comparator consisted of 46 complex gates (2- and 3-input NAND, NOR and inverter gates).

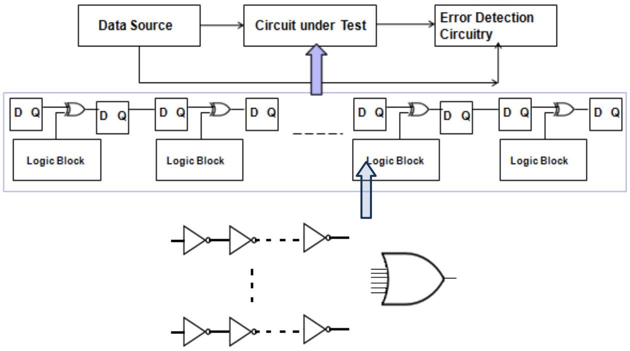


Figure A3-1 Each logic block consists of a group of 12 chains each consisting of 6 inverters each. The OR gate shown in the above figure consisted of 11 separate OR gates and each OR gate is constructed using a NOR gate and an inverter.

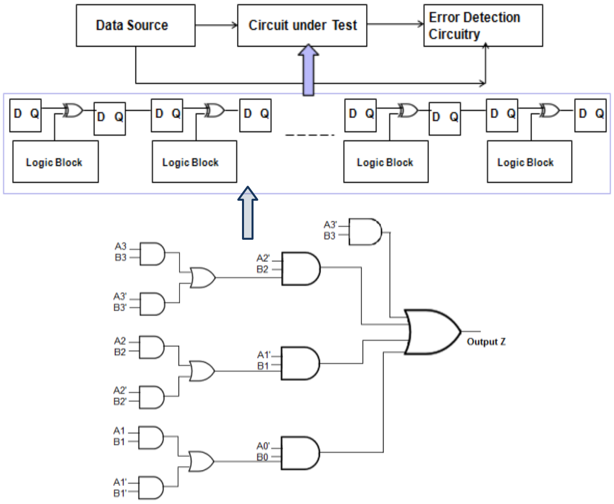


Figure A3-2 Each logic block consists of a group of a 4-bit comparator. The comparator consists of 46 complex gates (2- and 3- input NAND and NOR gates respectively) in addition to inverters.

The length of all transistors used in the designs was 30 nm and the minimum transistor width used in the designs was 100 nm. The gates were sized to achieve rise and fall time for all gates similar to a balanced minimum sized inverter (W_N = 100 nm and W_P = 250 nm). Table I provides more information about the gate counts and total number of transistors for the two logic circuits.

Number of gates, transistors and transistor total area for different circuit types.

Circuit type	Inverter	Comparator
Total # of gates	94	46
Type of gates	83 NOT 11 NOR	26 NOT 12 2-input NAND 2 3-input NAND 6 2-input NOR
Total # of transistors	210	136

2.3 Test Details

The flux of 5.5 MeV alpha particles from an Americium-241 source with an activity of 10 µCi, was 430 particles/mm^2-s at a distance of 1 mm from the alpha source. Testing was carried out with the alpha source at a height of 1 mm from the die. The relative size of the alpha source and the de-capped exposed die were 1 cm^2 and 0.09 cm^2 respectively. Testing was carried out in accordance with JEDEC specifications [JEDE-06]. Experiments were repeated at least 16 times at each data point to reduce experimental and inter-die variability.

2.4 Experimentally Measuring FF and Logic Cross-Section

The logic cross-sections were calculated as follows: The low-frequency (10 MHz) cross-section is assumed to yield the flip-flop cross-section alone. At very low frequencies logic contribution is extremely low and can be neglected [Buch-97]. The frequency dependence of flip-flop errors can be neglected because logic blocks are present between each of the flip-flop stages. Any errors due to single-event transients (SETs) in previous flip-flop stages are likely to be masked by the logic present between flip-flop stages. Besides, the size of the logic is much larger than the flip-flop latch stages. The average flip-flop cross-section resulting from several trials was calculated. Following this, the C-CREST chains were operated at higher frequencies. At high frequencies the total number of errors recorded is due to flip-flop SEUs and logic errors. The logic cross-section was obtained by subtracting the low-frequency flip-flop cross-section from the high-frequency cross-section. This is summarized in the following Equations.

$$Cross-Section\,per\,block = \frac{Total Number of\ Errors}{Total Number of\ stages \cdot Fluence} \qquad \text{A3-1}$$

$$\begin{aligned} &High\ Frequency\ Logic\ Cross-Section\,per\ block \quad = \\ &(Cross-Section\,per\,block) - (Cross-Section\,per\,block)_{@\,10MHz} \end{aligned} \qquad \text{A3-2}$$

2.5 Results

Figure A3-3 shows the alpha particle cross-section of the inverter and comparator circuits per (logic+flip-flop) stage, up to a frequency of 500 MHz. The input applied to the comparator was A = ‘0000’ and B = ‘0000’. The low frequency cross-section of the circuit was 2.1x10^{-11} cm^{2}. This is plotted as a constant value across the frequency spectrum and is assumed to be the cross-section of the FF design. On the other hand the logic cross-section increases with frequency for both, the inverter and the comparator circuits. At 500 MHz, the comparator cross-section is about 0.6 times the flip-flop cross-section, while the inverter is about 0.3 times the flip-flop cross-section. In other words the comparator contributes about 40% of the total soft errors recorded at 500 MHz, while the inverter contributes about 25% of the total soft errors at 500 MHz.

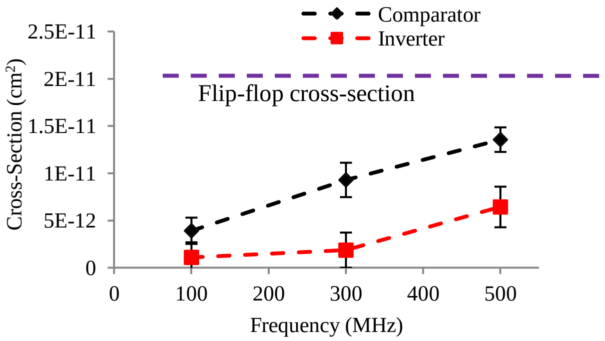


Figure A3-3 Logic cross-sections plotted as a function of frequency shows a clear increase. The low frequency cross-section of the flip-flop is plotted as being constant with frequency. The cross-section at 500 MHz for the comparator (inverter) is 0.6 (0.3) times the flip-flop cross-section.

In order to calculate the frequency threshold at which logic soft errors will exceed flip-flop soft errors, the cross-section of the comparator and inverter are extrapolated as a function of frequency. The resulting plot shown in Figure A3-4 predicts that the soft errors from the comparator circuit for this input would exceed those from the flip-flop at about 700 MHz, while the same for the inverter circuit would occur at about 1.6 GHz. Such frequencies are well within the operating region of modern processors, ASICs and even FPGAs. Both these results clearly illustrate that logic soft errors could form a significant proportion of the total soft errors recorded from complex circuits. Another key aspect about these results is that there isn't a single value of the frequency threshold at which logic soft errors will exceed flip-flop errors. As shown in Figure A3-4, the threshold frequency for different circuits in the same technology can differ by as much as 900 MHz. In addition to this, logical masking may result in different threshold frequencies depending on the inputs applied. This will complicate the estimation of the threshold frequency for different circuits for different input conditions.

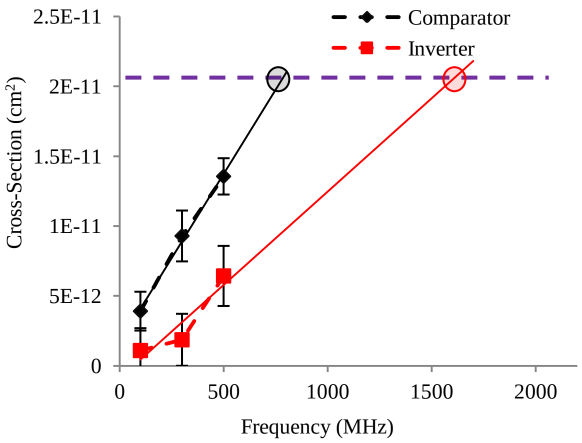


Figure A3-4 The frequency threshold for the comparator input with A = 0000 and B = 0000 is about 700 MHz. The threshold frequency for the inverter on the other and is about 1.6 GHz. A wide range of threshold frequencies is possible depending on the inputs and sensitive area exposed.

In the past, the logic SER has been characterized using inverters. This approach involves calculating or measuring the logic soft error rate of inverters and then using this number to quantify the SER of other logic circuits. However this may underestimate the logic soft error contribution. This is shown in Figure A3-4 where the comparator circuit cross-section is actually higher than that of the inverter circuit, although the comparator circuit has fewer gates than the inverter circuit. In fact, since the comparator circuit has 46 logic gates while the inverter circuit has 83 logic gates, the percentage contribution per logic gate for the comparator circuit is almost 4-5 times that for the inverter circuit as shown in Figure A3-6. Therefore there is a need to develop a simple yet robust technique to better characterize the frequency threshold at which logic soft errors will dominate the total SER of circuits.

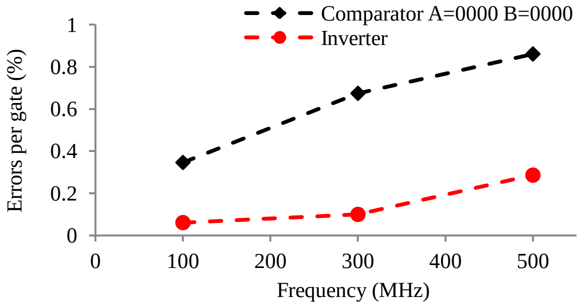


Figure A3-5 The per gate contribution of logic errors to the total SER is higher for the comparator than for the inverter. The comparator has 46 gates while the inverter has 94.

2.6 Characterizing the Frequency Threshold

Since the approach to merely scale the logic soft-error rate according to inverter contribution tends to underestimate the logic error rate, an alternative approach is presented here. The approach involves characterizing the inverter and NAND/NOR contributions separately to account for the differences in transient pulse-widths as well. Intuitive as well as empirically verified models suggest that the logic cross-section can be modeled as shown in Equation 3 [Lide94], [Alex-11], [Wang-08],[Nguy-05].

$$\mathrm{Cross-Section}_{\mathrm{estimated}} = \mathrm{A} \cdot \mathrm{EM} \cdot \mathrm{LM} \cdot \mathrm{TM} \qquad \textbf{A3-3}$$

where *A* is the transistor sensitive area. *EM*, *LM* and *TM* represent logical-, electrical- and temporal masking respectively. The cross-section is a function of the inputs applied, since the total area of the OFF transistors changes with the applied inputs. In other words, OFF transistor area is the sum of all drain areas of the transistors in the circuit, transients from which can propagate to the output. Electrical masking results in SETs being attenuated as they propagate through logic circuits. The temporal masking factor can be expressed as follows [Nguy-03].

$$TM = \frac{t_{WOV}}{T_{clk}} \qquad \textbf{A3-4}$$

where t_{wov} and T_{clk} are the window of vulnerability (WOV) and the clock period respectively. The WOV depends on both the setup-and-hold time of the flip-flop as well as the SET pulse-width distribution. We now make two simplifying assumptions. The first assumes the electrical masking *EM* to be equal to 1. SPICE simulations were used to calculate the electrical masking for a 10 stage inverter chain. Each stage had a uniform Fo4 loading. The maximum pulse-width reduction due to propagation through 10 stages, observed at the output, was only 6 ps. On the other hand experimentally observed SET pulse-widths for similar technologies have been reported in the 250 ps to 1.5 ns range [Hara-08]-[Nara-08]. Thus, the SET pulse-width value is much larger compared to the electrical masking introduced by propagation. In fact, the logic depth of most modern circuits is less than 10 (the maximum logic depth was 9 in this work) [ARM-11]-[Inte-11]. Therefore, in the case of modern circuits with few logic stages, electrical masking can be safely neglected without any loss of accuracy. Secondly a single effective value of the WOV is assumed for simplicity. This assumption is justified using experimental results, subsequently. Thus Equation 3 can be simplified as follows.

$$\text{Cross} - \text{Section}_{\text{estimated}} = \text{A}_{\text{LM}} \cdot \frac{< K >}{\text{T}_{\text{clk}}} \qquad \textbf{A3-5}$$

$$\frac{\text{Cross} - \text{Section}_{\text{estimated}}}{\text{A}_{\text{LM}} \cdot \text{F}_{\text{clk}}} = K \qquad \textbf{A3-6}$$

Here A_{LM} is the total OFF transistor drain area that is sensitive to transients for the particular input. *K* is the effective value of the WOV for each circuit node. Thus the differences in WOV due to the transient distribution at each circuit node are replaced by a single value. This value need not necessarily be a constant. This value was calculated for different inputs of the comparator circuit and was compared to those of the inverter circuit. The experimentally measured cross-section for each input condition was divided by the total OFF transistor drain area and the clock frequency (500 MHz) to obtain *K*. As Figure A3-6 shows, the value of *K* is not significantly different for all the inputs at 500 MHz. The error bars at each point represent the standard error of measurement at each point. The colored band represents the maximum variation in *K* for all the input conditions. As seen in Figure A3-6, the values of *K* for *different inputs* are not significantly different. This justifies the use of a single value of *K* within reasonable bounds. It also shows that the value of *K*, which includes the SET pulse-width, is slightly lower for inverters than it is for NAND/NOR gates. This is in agreement with [Cann-09]-[Atki-11], where the lower restoring drive for NAND/NOR is cited as the reason behind longer SET pulse-widths for NAND and NOR gates compared to inverters. The OFF transistor drain area is a good estimate of the sensitive area for low-LET particles [Dasg-07]. This was also verified using 3D TCAD simulations where calibrated 28 nm models were struck with 1 Mev/cm^2-mg low-LET alpha-like particles. The charge collection region was limited to the drain area and pulse-widths reduced to 0, when the transistor was struck outside the drain region. The values of *K* were calculated at 100 MHz and 300 MHz and were found to be similar to those obtained at 500 MHz. This also confirms that *K* is independent of the frequency. There is no physical reason for the setup-and-hold window or the SET pulse-widths to be related to the frequency of operation.

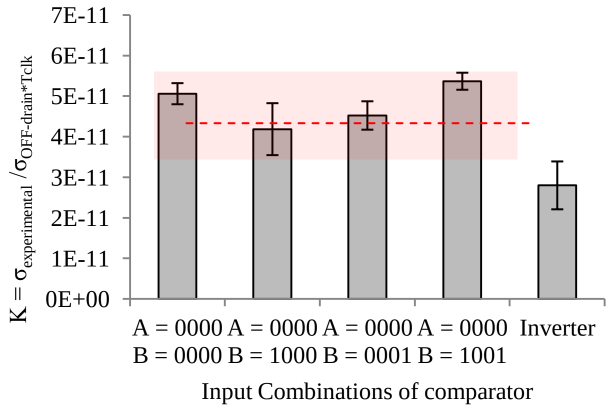


Figure A3-6 The value of K is plotted for different input combinations of the comparator and the inverter. The comparator values are distinctly higher than that of the inverter. Moreover the values of K for the different inputs of the comparator are comparable to each other.

Based on the value of K, the value of the frequency cross-over or the frequency threshold for a variety of circuits can be calculated. The ISCAS-85 Benchmark circuits analyzed in this work are listed in Table II [ISCA-85]. The underlying assumption is that these circuits would be synthesized using a similar process develop kit (PDK) as the one used in this analysis. Indeed the use of a different PDK would mean that some of the factors that affect K would change (WOV and area).

TABLE II.
Number of gates, inputs-outputs for the benchmark circuits analyzed.

Circuit Name	Number	# gates	# I/O
4-bit ALU	74181	61	14/8
16-bit Multiplier	C6288	2406	32/32
8-bit ALU	C880	383	60/26
27 channel interrupt controller	C432	160	36/7
32-bit adder/comparator	C7552	3512	207/108

The average value of K (4.8x10^{-11} for NAND/NOR and 2.7x10^{-11} for inverters) shown in Figure A3-6 was used to calculate the frequency threshold for all these circuits. The average value was defined separately for the NAND/NOR gates and separately for the inverters. Detailed explanation of the technique to calculate the frequency threshold is provided below for the 4-bit ALU (74181). Since the circuit has 8 outputs, the flip-flops cross section was 8*nominal low frequency cross-section observed in the results for the comparator and the inverter circuits. The nominal low-frequency flip-flop cross-section was about 2.1x10^{-11} cm^{2} (observed experimentally, Figure A3-3). Thus the total flip-flop cross-section was about 16.8 x10^{-11} cm^{2}. The total logic OFF-transistor area was calculated as follows. For each input, faults were injected at each node. If faults injected at a particular node appear at the output, then the OFF transistor area at that node (gate) is added to the sensitivity list. For each input, faults were injected at each node to determine the total OFF transistor drain area for that input combination. This was repeated for each input. The pseudo code shown below summarizes this technique.

```
Start:  Describe circuit in Structural Verilog/VHDL.

for (input = 1, input<= max_inputs, input++)
{
        for (m = 1, m<= max_nodes,m++)
        inject fault at node m
        observe output for error
                if (error observed at output)
                {
                1. Calculate OFF drain area based
                    on gate type
                2. Add to sensitivity list
                3. Update sum of area of list
                }
      }
}
End;
```

In order to calculate the cross-over frequency, Equation 6 was used. The estimated cross-section was replaced by the flip-flop cross-section (16.8x10^{-11} cm^2). The sensitive area was obtained from the above analysis, and the appropriate value of *K* was used for NAND/NOR and inverter gates.

Based on this, the threshold frequency for different inputs is plotted as a histogram in Figure A3-7. The threshold frequency for a majority of the input combinations is between 700 MHz and 2.1 GHz. Even after including the effects of logical masking, the logic error could exceed flip-flop errors in the 700 MHz to 2.1 GHz. Clearly, circuits fabricated using modern technology nodes are capable of operating at these frequencies and for most of the inputs applied, the threshold frequency is well within operating range.

Exhaustive analysis was only possible for the ALU 74181 because of the limited size of the circuit. As such, the total number of combinations for fault injection is 2^N xm, where N and m are the number of inputs and nodes in the circuit respectively. Besides, the logical masking factor can be very different for different input combinations. As a result the frequency threshold varies widely for different inputs. In such circumstances it is hard to predict whether logic errors or flip-flop errors would dominate the total SER for the circuit in general. For the much larger circuits, at least 10^6 combinations were evaluated to calculate the different sensitive areas for each combination.

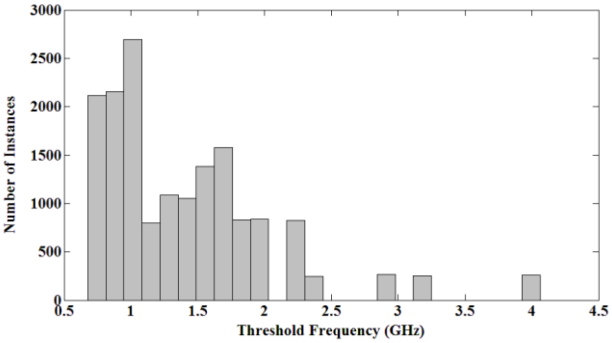


Figure A3-7 Distribution of frequency thresholds for each input of the ALU 74181. The frequency thresholds are distributed between 700 MHz and 4 GHz. Majority of the values lie between 700 MHZ and 2.1 GHz. The average value is 1.3 GHz.

However a useful metric that can be used to quantify the frequency threshold for the whole circuit is introduced below. In the case of the ALU 74181 described above, the frequency threshold was calculated for all the input vectors. The Cumulative Distribution Function (CDF) of the threshold frequencies is plotted in Figure A3-8. The median of the distribution is also shown on the plot. The median in this case represents the frequency threshold that is exceeded by half the input vectors applied. In other words, for 50 % of the inputs applied, the frequency threshold is less than 1.2 GHz. Thus for a designer, it indicates that for 50 % of the possible inputs to the circuit, the frequency threshold is 1.2 GHz. As more input vectors are included, the frequency threshold for the “circuit” increases. At 4 GHz, all input vectors applied would result in logic errors dominating flip-flop errors. Thus depending upon the frequency of operation and the threshold frequency at which logic errors would exceed flip-flop errors, designers can adopt suitable logic hardening strategies.

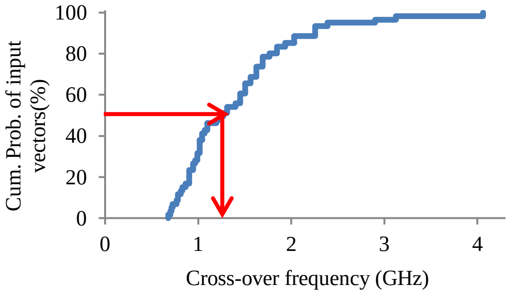


Figure A3-8 CDF of the frequency threshold for all inputs to the ALU 74181. The median value of the frequency threshold is about 1.2 GHz. 50 % of the input vectors would result in a frequency threshold that is less than 1.2 GHz.

The median value of the frequency threshold was similarly calculated for other ISCAS benchmark circuits as shown in Figure A3-9. As Figure A3-9 suggests, the threshold frequency at which logic errors could dominate for most circuits, lies in the 900 MHz - 2 GHz range. One of the circuits; c432, has a frequency threshold that is much higher than the rest. This due to the fact that c432 is a 27 channel interrupt controller. Most parts of the logic circuitry are inactive for a large majority of the inputs. This underscores the point that different circuits can have very different threshold frequencies at which logic errors dominate the total SER.

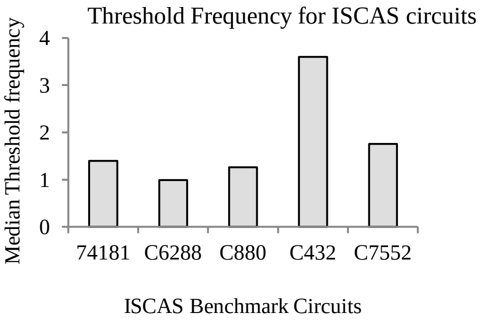


Figure A3-9 Median frequency thresholds for different benchmark circuits. The lowest value is 900 MHz while the highest is about 3.5 GHz. Most of the threshold values lie well within the operating region of modern semiconductor devices and circuits.

2.7 Conclusions

For the terrestrial environment, logic soft errors are likely to dominate the total SER for future technology generations with ICs operating at multi-GHz range of frequencies. Through alpha particle exposure, it is shown that logic soft errors are a strong function of individual logic-gate sensitive area and not of gate count. Logic errors account for as much as 25 - 50% of the total errors recorded for the circuits tested at 500 MHz. The frequency threshold for ISCAS circuits was calculated using simple yet robust approximations of the logic SER. The frequency threshold beyond which logic soft errors will dominate overall soft-error rate is about 900 MHz to 2.1 GHz for most of the ISCAS circuits. The impact of this is that, if logic errors dominate, flip-flop hardening alone will not reduce system level SER. While scaling may result in a modest decrease in logic SER due to smaller active areas, as in the case of flip-flops, higher operating frequencies will mean that the frequency threshold will be easily within the operating range of even complex circuits.

REFERENCES

[Ahlb-08] Ahlbin J., Black J. D, Massengill L. W., Amusan O. A., Balasubramanian A., M. Casey, D. Black, M. McCurdy, R. Reed, B. Bhuva, “C-CREST Technique for Combinational Logic SET Testing,” IEEE Transactions on Nuclear Science, Vol. 55, pp 3347-3351, 2008.

[Ahlb-12] Ahlbin J., Ph.D Dissertation Vanderbilt University, 2012.

[Aker-59] Akers, Jr, B. Sheldon, "On a theory of Boolean functions." Journal of the Society for Industrial & Applied Mathematics Vol 7, pp 487-498, 1959.

[Aldi-94] Alidina, M.; Monteiro, J.; Devadas, S; Ghosh, A.; Papefthymiou, M., "Precomputation-based Sequential Logic Optimization For Low Power," Computer-Aided Design, 1994., IEEE/ACM International Conference on , vol., no., pp.74,81, 6-10 Nov 1994

[Alex-11] D. Alexandrescu and E. Costenaro, “A Practical Approach to Single Event Transients Analysis For Highly Complex Designs”, Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, 2011, pp 155-163.

[Almu-08] Almukhaizim S., Makris Y., “Soft error mitigation through selective addition of functionally redundant wires,” IEEE Trans. Rel., vol. 57, no. 1, pp. 23–31, Mar. 2008.

[Amus-07] Amusan, O.A.; Massengill, L.W.; Bhuva, B.L.; DasGupta, S.; Witulski, A.F.; Ahlbin, J.R.; “Design Techniques to Reduce SET Pulse Widths in Deep-Submicron Combinational Logic”, IEEE Transactions of Nuclear Science, Vol 54, No. 6, pp 2060-2064, Dec 2007.

[Amus-06] Amusan, O., Ph.D Dissertation Vanderbilt University, 2006.

[ARM-11] http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0360e/I1002919.html, ARM 11 processors.

[Asha-91] P. Ashar, S. Devadas, and A. R. Newton. Sequential Logic Synthesis. KluwerAcademic Publishers, Boston, Massachusetts, 1991.

[Atki-11] N. M. Atkinson, J. R. Ahlbin, A. F. Witulski, N. J. Gaspard, W. T. Holman, B. L. Bhuva, E. X. Zhang, L. Chen, and L. W. Massengill "Effect of Transistor Density and Charge Sharing on Single-Event Transients in 90-nm Bulk CMOS", IEEE Transactions on Nuclear Science, Vol 58, No. 6, pp 2578-2584, Dec. 2011.

[Baha-94] R. I. Bahar, H. Cho, G. D. Hachtel, E. Macii, and F. Somenzi, "A Symbolic Method to Reduce Power Consumption of Circuit Containing False Paths". In Proceedings of the Int'l Conference on Computer-Aided Design, pages 368–371, November 1994.

[Baum-05] Baumann R., "Radiation-Induced Soft Errors in Advanced Semiconductor Technologies," IEEE Transactions on Device Materials and Reliability, Vol. 5, pp 305-316, 2005.

[Bind-75] D. Binder, E. C. Smith, and A. B. Holman, "Satellite anomalies from galactic cosmic rays," IEEE Trans. Nucl. Sci., vol. 22, pp. 2675–2680, Dec. 1975.

[Bork-12] S. Borkar, IPDPS Invited Talk, 2012.

[Buch-93] Buchner S., Kang K., "Dependence Of The SEU Window Of Vulnerability Of A Logic Circuit On Magnitude Of Deposited Charge", IEEE Transactions On Nuclear Science, Vol. 40, No. 6, December 1993, pp 1853-1857.

[Buch-97] Buchner S., M. Baze, D. Brown, D. McMorrow, J. Melinge, "Comparison of Error Rates in Combinational and Sequential Logic," IEEE Transactions on Nuclear Science, Vol. 44, pp. 2209-2216, 1997.

[Buch-00] Buchner, S.; Campbell, A.B.; Meehan, T.; Clark, K.A.; McMorrow, D.; Dyer, C.; Sanderson, C.; Comber, C.; Kuboyama, S.; "Investigation of single-ion multiple-bit upsets in memories on board a space experiment", IEEE Transactions on Nuclear Science, Vol 47, pp 705-711, 2000.

[Bund-94] J. Bunda, W. Athas, and D. Fussell. Evaluating power implications of CMOS microprocessor design decisions. In Proceedings of the International Workshop on Low Power

Design, pages 147–152, Napa, CA, Apr. 1994.

[Cann-09] Cannon, E.H., Cabanas-Holmen, M., “Heavy Ion and High Energy Proton-Induced Single Event Transients in 90 nm Inverter, NAND and NOR Gates”, IEEE Trans. Nucl. Sci. Vol 56, pp 3511-3518, 2009.

[Catt-94] F. Catthoor et al. Global communication and memory optimizing transformations for low power signal processing systems. In IEEE workshop on VLSI signal processing, La Jolla, CA, Oct. 1994.

[Cell-05] IBM Cell Processor, Complete references https://www.ibm.com/developerworks/library/pa-cellperf/#resources

[Chan-92] A. Chandrakasan, T. Sheng, and R. W. Brodersen. Low Power CMOS Digital Design. Journal of Solid State Circuits, 27(4):473–484, April 1992.

[Chan-94] Anantha P. Chandrakasan. Low-Power Digital CMOS Design. PhD thesis, University of California at Berkeley, UCB/ERL Memorandum No. M94/65, August 1994.

[Chan-95] A. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R. Brodersen. Optimizing power using transformations. IEEE Transactions on Computer-aided Design, 14(1), January 1995.

[Chat-94] A. Chatterjee and R. Roy. Synthesis of low power linear DSP circuits using activity metrics. In International Conference on VLSI Design, India, Jan. 1994.

[Choi-02] I. Choi, K. Hyoung, S. Lim, S. Hwang, B. Lee, B. Kim, "A Kernel-Based Partitioning Algorithm for Low-Power, Low-Area Overhead Circuit Design Using Don’t-Care Sets." ETRI Journal Vol. 24, No. 6, pp 473-476, 2002.

[Chre-95] W. A. Chren. Low Delay-Power Product CMOS Design Using One-Hot Residue Coding. In Proceedings of the Int’l Symposium on Low Power Design, April 1995.

[Dasg-07] Dasgupta S. “Trends in Single Event Pulse Widths and Pulse Shapes in Deep Submicron CMOS”, MS Thesis, Vanderbilt University, 2007.

[Dasg-07] S. DasGupta, A. F. Witulski, B. L. Bhuva, M. L. Alles, R. A.

Reed, , O. A. Amusan, J. R. Ahlbin, R. D. Schrimpf and L. W. Massengill, "Effect of Well and Substrate Potential Modulation on Single Event Pulse Shape in Deep Submicron CMOS," IEEE Trans. On Nucl. Sci., vol. 54, no. 6, pp. 2407–2412, Dec. 2007.

[Dieh-84] S. E. Diehl-Nagle, “Single event upset rate predictions for complex logic systems,” IEEE Trans. Nucl. Sci., vol. 31, no. 6, pp. 1132–1138, Dec. 1984.

[Dixi-11] A. Dixit, A. Wood, "The impact of new technology on soft error rates," Reliability Physics Symposium (IRPS), 2011 IEEE International, pp.5B.4.1,5B.4.7, 10-14 April 2011.

[Deva-94] S. Devadas, A. Ghosh, and K.Keutzer. Logic Synthesis. McGraw Hill, New York, NY, 1994.

[Dodd-03] Dodd P., Massengill L. W., “Basic Mechanisms and Modeling of Single-Event Upset in Digital Microelectronics”, IEEE Transactions On Nuclear Science, Vol. 50, No. 3, pp 583-602, June 2003.

[Ferl-04] V. Ferlet-Cavrois, G. Vizkelethy, P. Paillet, A. Torres, J. R. Schwank, M. R. Shaneyfelt, J. Baggio, J. du Port de Pontcharra, and L. Tosti, “Charge enhancement effect in NMOS bulk transistors induced by heavy ion irradiation—Comparison with SOI,” IEEE Trans. Nucl. Sci., vol. 51, no. 6, pp. 3255–3262, Dec. 2004.

[Ferl-06] V. Ferlet-Cavrois, P. Paillet, M. Gaillardin, D. Lambert, J. Baggio, J. R. Schwank, G. Vizkelethy, M. R. Shaneyfelt, K. Hirose, E.W. Blackmore, O. Faynot, C. Jahan, and L. Tosti, “Statistical analysis of the charge collected in SOI and bulk devices under heavy ion and proton irradiation-implications for digital SETs,” IEEE Trans. Nucl. Sci., vol. 53, no. 6, pp. 3242–3252, Dec. 2006.

[Frie-85] A. L. Friedman, B.Lawton, K. R. Hotelling, J.C. Pickel, V.H.Strahan and K. Loree, “Single event upset in combinatorial and sequential current mode logic,” IEEE Trans. Nucl. Sci., vol. 32, no. 6, pp. 4216–4218, Dec. 1985.

[Gadl-08] M. J. Gadlage, R. D. Schrimpf, B. Narasimham, J. A. Pellish, K. M. Warren, R. A. Reed, R.A. Weller, B. L. Bhuva, L. W. Massengill, Xiaowei Zhu, "Assessing Alpha Particle-Induced Single Event Transient Vulnerability in a 90-nm CMOS

Technology," Electron Device Letters, IEEE , vol.29, no.6, pp.638,640, June 2008.

[Gais-97] Gaisler, Jiri. "Evaluation of a 32-bit microprocessor with built-in concurrent error-detection." In Fault-Tolerant Computing, 1997. FTCS-27. Digest of Papers., Twenty-Seventh Annual International Symposium on, pp. 42-46. IEEE, 1997.

[Gasi-06] Gasiot G. , Giot D., Roche P., “Alpha-Induced Multiple Cell Upsets in Standard and Radiation Hardened SRAMs Manufactured in a 65 nm CMOS Technology”, IEEE Transactions On Nuclear Science, Vol. 53, No. 6, December 2006.

[Ghos-92] A. Ghosh, S. Devadas, K. Keutzer, and J. White. Estimation of Average Switching Activity in Combinational and Sequential Circuits. In Proceedings of the 29th Design Automation Conference, pages 253–259, June 1992.

[Gill-09] B. Gill, N. Seifert, V. Zia, "Comparison of alpha-particle and neutron-induced combinational and sequential logic error rates at the 32nm technology node," Reliability Physics Symposium, 2009 IEEE International, pp.199-205, 26-30 April 2009.

[Gobe-56] Gobelli G., “Range-Energy Relation for Low-Energy Alpha Particles in Si, Ge, and InSb”, Physics Reviews, Vol. 103, No. 2, pp 275-278, May 1956.

[Good-94] L. Goodby, A Orailoglu, and P. Chau. Microarchitectural synthesis of performance-constrained, low-power VLSI designs. In Proceedings of the International Conference on Computer Design, pages 323–326, Boston, MA, Oct. 1994.

[Guen-81] Guenzer, C. S., A. B. Campbell, and P. Shapiro. "Single event upsets in NMOS microprocessors." Nuclear Science, IEEE Transactions on 28, no. 6 (1981): 3955-3958.

[Gunt-08] S. Gunther and R. Singhal, “Next generation Intel micro-architecture (Nehalem) family: Architectural insights and power management,” Intel Developer Forum, 2008

[Hach-94] G. D. Hachtel, M. Hermida, A. Pardo, M. Poncino, and F. Somenzi. Re-Encoding Sequential Circuits to Reduce Power Dissipation. In Proceedings of the Int’l Conference on Computer-Aided Design, pages 70–73, November 1994.

[Hans-99] Hansen, M.C.; Yalcin, H.; Hayes, J.P.; “Unveiling the ISCAS-85 benchmarks: a case study in reverse engineering”, IEEE Design & Test of Computers, vol. 16, pp. 72-80, 1999.

[Hara-10] R. Harada, Y. Mitsuyama, M. Hashimoto and T. Onoye, “Measurement circuits for acquiring SET pulse-width distribution with sub-1ns inverter delay distribution”, Proceedings of IEEE International Symposium on Quality Electronic Design, 2010, pp 839-844.

[Hard-86] R. Harboe-Sørensen, L. Adams, E. J. Daly, C. Sansoe, D. Mapper, and T. K. Sanderson, “The SEU risk assessment of Z80A, 8086 and 80C86 microprocessors intended for use in a low altitude polar orbit,” IEEE Trans. Nucl. Sci., vol. 33, no. 6, pp. 1626–1631, Dec. 1986.

[Hart-02] A. Hartstein and T. Puzak. The optimum pipeline depth for a microprocessor. In ISCA 29, pages 7–13, May 2002.

[Hass-89] Hass, K. J., R. K. Treece, and A. E. Giddings. "A radiation-hardened 16/32-bit microprocessor." Nuclear Science, IEEE Transactions on 36, no. 6 (1989): 2252-2257.

[Hazu-00] Hazucha P., Christer Svensson, Stephen A. Wender, “Cosmic-Ray Soft Error Rate Characterization of a Standard 0.6-micron CMOS Process”, IEEE Journal Of Solid-State Circuits, Vol. 35, No. 10, October 2000.

[Hazu-03] Hazucha P., T. Kamik, J. Maiz, S. Walstra, B. Bloechel, J. Tschanz, G. Dermer, S. Hareland, P. Armstrong, S. Borkar “Neutron Soft Error Rate Measurements in a 90-nm CMOS Process and Scaling Trends in SRAM from 0.25-pm to 90-nm Generation”, Proceedings of IEDM Technical Digest, pp 21.5.1-21.5.4, 2003.

[Heid-06] Heidel, D.F.; Marshall, P.W.; LaBel, K.A.; Schwank, J.R.; Rodbell, K.P.; Hakey, M.C.; Berg, M.D.; Dodd, P.E.; Friendlich, M.R.; Phan, A.D.; Seidleck, C.M.; Shaneyfelt, M.R.; Xapsos, M.A.; , “Low Energy Proton Single-Event-Upset Test Results on 65 nm SOI SRAM”, IEEE Transactions on nuclear Science, December 2008, pp 3394- 3500.

[Heo-04] Seongmoo Heo; Asanovic, K., "Power-optimal pipelining in deep submicron technology," Low Power Electronics and Design, 2004. ISLPED '04. Proceedings of the 2004

International Symposium on , vol., no., pp.218,223, 9-11 Aug. 2004

[Hris-02] M. Hrishikesh et al. The optimal logic depth per pipeline stage is 6 to 8 FO4 inverter delays. In ISCA 29, pages 14–24, May 2002.

[Hs-81] C. M. Hsieh, P. C. Murley, and R. R. O'Brien, "A Field-Funneling Effect on the Collection of Alpha-Particle-Generated Carriers in Silicon Devices," IEEE Electron Device Letters, vol. 2, no. 4, pp. 103–105, Apr. 1981.

[Iman-94] S. Iman and M. Pedram, "Multi-Level Network Optimization for LowPower". In Proceedings of the Int'l Conference on Computer-Aided Design, pages 371–377, November 1994.

[Inte-10] Intel R 64 and IA-32 Architectures Software Developer's Manual Volume 1: Basic Architecture, Intel Corporation, Jun 2010.

[ISCA-85] ISCAS-85 Benchmark Suite http://web.eecs.umich.edu/~jhayes/iscas.restore/benchmark.html, ISCAS-85 benchmark circuits.

[Jaga-12] S. Jagannathan, T.D. Loveless, B.L. Bhuva, N.J. Gaspard, N. Mahatme, T. Assis, S. J. Wen, R. Wong, L. W. Massengill, "Frequency Dependence of Alpha-Particle Induced Soft Error Rates of Flip-Flops in 40-nm CMOS Technology", IEEE Transactions on Nuclear Science, Vol 59, pp 2796-2802, Dec 2012.

[JEDE-06] JEDEC Standards for alpha particle testing, JESD89, October 2006,
http://www.jedec.org/standards-documents/results/taxonomy%3A2612

[Kany-06] Kanyogoro N., S. Buchner, D. McMorrow, H. Hughes, M. Liu, A. Hurst, C. Carpasso, "New Approach for Single-Event Effects Testing With Heavy Ion and Pulsed-Laser Irradiation: CMOS/SOI SRAM Substrate Removal", IEEE Transactions on Nuclear Science, Vol. 57, No. 6, pp 3414-3418, 2006.

[Karn-02] Karnik, Tanay, Sriram Vangal, V. Veeramachaneni, Peter Hazucha, Vasantha Erraguntla, and Shekhar Borkar. "Selective node engineering for chip-level soft error rate improvement [in cmos]." In VLSI Circuits Digest of Technical Papers, 2002.

Symposium on, pp. 204-205. IEEE, 2002.

[Kaup-09] Kauppila, J.S.; Sternberg, A.L.; Alles, M.L.; Francis, A.M.; Holmes, J.; Amusan, O.A.; Massengill, L.W., "A Bias-Dependent Single-Event Compact Model Implemented Into BSIM4 and a 90 nm CMOS Process Design Kit," Nuclear Science, IEEE Transactions on , vol.56, no.6, pp.3152,3157, Dec. 2009

[Kaup-09] Kauppila J. S., Sternberg A. L., Alles M. L., Francis A. M., Jim Holmes, Oluwole A. Amusan, Massengill L. W., “A Bias-Dependent Single-Event Compact Model Implemented Into BSIM4 and a 90 nm CMOS Process Design Kit” , IEEE Transactions on nuclear Science, December 2009, pp 3152-3157.

[Keut-87] K.Keutzer. DAGON: Technology Mapping and Local Optimization. In Proceedings of the 24th Design Automation Conference, pages 341–347, June 1987.

[King-10] King, Michael Patrick, et al. "The impact of delta-rays on single-event upsets in highly scaled SOI SRAMs." Nuclear Science, IEEE Transactions on 57.6 (2010): 3169-3175.

[Koga-85] R. Koga, W. A. Kolasinski, M. T. Marra, and W. A. Hanna, “Techniques of microprocessor testing and SEU-rate prediction,” IEEE
Trans. Nucl. Sci., vol. 32, no. 6, pp. 4219–4224, Dec. 1985.

[Kunk-86] S. R. Kunkel and J. E. Smith. Optimal pipelining in supercomputers. In Proceedings 13th Symposium on Computer Architecture, pages 404–414, Tokyo, Japan, June 1986.

[Land-93] P. Landman and J. Rabaey. Power estimation for high level synthesis. In Proceedings of the European Design Automation Conference, pages 361–366, Paris, Feb. 1993.

[Lava-95] L. Lavagno, P. C. McGeer, A. Saldanha, A., A. L. Sangiovanni-Vincentelli, “Timed Shannon circuits: a power-efficient design style and synthesis tool”, Proceedings of ACM/IEEE Design Automation Conference pp. 254-260, 1995.

[Lee-95] T. C. Lee, V. Tiwari, S.Malik, andM. Fujita. Power analysis and low-power scheduling techniques for embedded DSP software. Technical Report FLA-CAD-95-01, Fujitsu Labs of America, March 1995.

[Leis-83] C. E. Leiserson, F. M. Rose, and J. B. Saxe. Optimizing Synchronous Circuitry by Retiming. In Proceedings of 3rd CalTech Conference on VLSI, pages 23–36, March 1983.

[Lemo-94] C. Lemonds and S. S. Mahant Shetti. A Low Power 16 by 16 Multiplier Using Transition Reduction Circuitry. In Proceedings of the Int'l Workshop on Low Power Design, pages 139–142,
April 1994.

[Lide-94] Liden P., P. Dahlgren, R. Johansson, J. Karlsson, "On Latching Probability of Particle-Induced Transients in Combinational Networks," Proceedings of International Symposium on Fault-Tolerant Computing, pp. 340-349, 1994.

[Lilj-14] K. Lilja, M. Bounasser, L. Chen, S. J. Wen, S. Baeg, N. Mahatme, B. L. Bhuva, "Logic Cells with Ultra Low SEU and SET rates in 20nm and 28nm Bulk Technologies Using Layout Optimization", to appear in IEEE Transactions on Nuclear Science.

[Limb-13] D. B. Limbrick, PhD Dissertation, Vanderbilt University, 2013

[Lyon-62] Lyons, Robert E., and Wouter Vanderkulk. "The use of triple-modular redundancy to improve computer reliability." IBM Journal of Research and Development 6, no. 2 (1962): 200-209.

[Ma-84] Ma T. and Dressendorfer P. V., "Ionizing radiation effects in MOS devices and circuits", Wiley Interscience, 1984.

[Maha-00] Mahapatra, N.R.; Garimella, S.V.; Takeen, A., "Efficient techniques based on gate triggering for designing static CMOS ICs with very low glitch power dissipation," Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on , vol.2, no., pp.537,540 vol.2, 2000

[Maha-11] N. N. Mahatme, S. Jagannathan, T. D. Loveless, L. W. Massengill, B. L. Bhuva, S.-J.Wen, and R.Wong, "Comparison of combinational and sequential error rates for a deep submicron process," IEEE Trans. Nucl. Sci., vol. 58, no. 6, pp.

2719–2725, Dec. 2011.

[Maha-13] N. N. Mahatme, I. Chatterjee, A. Patki, D. B. Limbrick, B. L. Bhuva, R. D. Schrimpf, W. Robinson, “An efficient technique to select logic nodes for single event transient pulse-width reduction”, Microelectronics Reliability, Volume: 53, Issue: 1, Page (s) 114–117, 2013.

[Maiz-03] J. Maiz, S. Hareland, K. Zhang, P. Armstrong, "Characterization of multi-bit soft error events in advanced SRAMs," Electron Devices Meeting, 2003. IEDM '03 Technical Digest. IEEE International , pp. 21.4.1,21.4.4, 8-10 Dec. 2003.

[Mass-93] Massengill L. W., “SEU modeling and prediction techniques,” in IEEE NSREC Short Course, 1993, pp. III-1–III-93.

[Mass-08] L. W. Massengill and P. W. Tuinenga, “Single-event transient pulse propagation in digital CMOS,” IEEE Trans. Nucl. Sci., vol.55, no.6, pp. 2861-2871, Dec. 2008.

[Mass-90] L. W. Massengill, D. V. Kerns, S. E. Kerns, and M. L. Alles, “Single-event charge enhancement in SOI devices,” IEEE Elec. Dev. Lett., vol. EDL-11, no. 2, pp. 98–99, Feb. 1990.

[Mavi-02] D. G. Mavis and P. H. Eaton, “Soft error rate mitigation techniques for modern microcircuits,” in Proc. IEEE Int. Rel. Phys. Symp., Dallas, TX, 2002, pp. 216–225.

[May-79] May T.,” Soft Errors in VLSI: Present and Future”, IEEE Transactions on Components, Hybrids, and Manufacturing Technology, pp 377-387, 1979

[May-79] May, T.C.; Woods, M.H.; “Alpha-particle-induced soft errors in dynamic memories”, IEEE Transactions on Electron Devices, pp 2-9, 1979.

[May-84] T. C. May, G. L. Scott, E. S.Meieran, P.Winer, and V. R. Rao, “Dynamic fault imaging of VLSI random logic devices,” in Proc. Int. Reliability Physics Symp., 1984, pp. 95–108.

[Mont-93] J. Monteiro, S. Devadas, and A. Ghosh. Retiming Sequential Circuits for Low Power. In Proceedings of the Int’l Conference on Computer-Aided Design, pages 398–402, November 1993.

[Mont-95] J. Monteiro and S. Devadas. Techniques for the Power

Estimation
of Sequential Logic Circuits Under User-Specified Input Sequences and Programs. In Proceedings of the Int'l Symposium on Low Power Design, April 1995.

[Moor-65] Moore G, "Cramming onto integrated circuits", Electronics, Vol 38(8), 1965

[Mukh-05] S. Mukherjee, J. Emer, S. K. Reinhardt, "The soft error problem: an architectural perspective," High-Performance Computer Architecture, 2005. HPCA-11. 11th International Symposium on , pp.243-247, 12-16 Feb. 2005.

[Najm-91] Najm F., "Transition Density a Stochastic Measure of Activity in Digital Circuits", ACM/IEEE Design Automation Conference, pp 644-649, 1991.

[Najm-94] F. Najm. A Survey of Power Estimation Techniques in VLSI Circuits (Invited Paper). IEEE Transactions on VLSI Systems, 2(4):446–455, December 1994.

[Nase-06] R. Naseer and J. Draper, "DF-DICE: A scalable solution for soft error tolerant circuit design," in Proc. Int. Symp. on Circuits and Systems, Island of Kos, May 2006, pp. 3890–3893.

[Naza-11] Nazar, G.L.; Carro, L., "An Area Effective Parity-Based Fault Detection Technique for FPGAs," Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2011 IEEE International Symposium on , vol., no., pp.27,33, 3-5 Oct. 2011.

[Nico-10] M. Nicolaidis, "Design techniques for soft-error mitigation," in Proc. Int. Conf. IC Des. Tech., Grenoble, Jun. 2010, pp. 208–214.

[Nieu-06] A. K. Nieuwland, S. Jasarevic, and G. Jerin, "Combinational logic soft error analysis and protection," in Proc. IEEE Int. On-Line Testing Symp., Lake Como, 2006, pp. 99–104.

[Norm-96] Normand E., "Single-event effects in avionics", IEEE Transactions on Nuclear Science, pp 461-474, 1996.

[Nguy-05] H. T. Nguyen, Y. Yagil, N. Seifert and M. Reitsma, "Chip-level soft error estimation method," IEEE Transactions on Device and Material Reliability, Vol. 5, No. 3, Sep. 2005, pp.

365–381,

[Nguy-03] H. T. Nguyen and Y. Yagil, “A Systematic Approach to SER Estimation and Solutions”, Proceedings of IEEE International Reliability Physics Symposium, 2003, pp 60-70.

[Ong-94] P. W. Ong and R. H. Yan. Power-conscious software design – a framework for modeling software on hardware. In Proceedings of 1994 IEEE Symposium on Low Power Electronics, pages 36–37, San Diego, CA, Oct. 1994.

[Pagl-12] S. N. Pagliarini, L. A. B. Naviner, and J.-F. Naviner, “Selective hardening methodology for combinational logic,” in Proc. Latin American Test Workshop, Quito, Ecuador, Apr. 2012, pp. 1–6.

[Park-75] Parker K. P., McKluskey E., “Probabilistic Treatment of General Combinatorial Networks,” IEEE Transactions on Computers Vol C-24 pp 668-670, 1975.

[Pick-78] J. C. Pickel and J. T. Blantford, “Cosmic ray induced errors in MOS memory cells,” IEEE Trans. Nucl. Sci., vol. 25, no. 6, pp. 1166–1171, Dec. 1978

[Poli-08] Polian, Ilia, Sudhakar M. Reddy, and Bernd Becker. "Scalable calculation of logical masking effects for selective hardening against soft errors." In Symposium on VLSI, 2008. ISVLSI'08. IEEE Computer Society Annual, pp. 257-262. IEEE, 2008.

[Powe-90] S. Powell et al. Estimating power dissipation of VLSI signal processing chips: The PFA technique. VLSI Signal Processing IV, pages 250–259, 1990.

[Pras-94] S. C. Prasad and K. Roy. Circuit Optimization for Minimization
of Power Consumption Under Delay Constraint. In Proceedings of the Int’l Workshop on Low Power Design, pages 15–20, April 1994.

[Ragh-94] A. Raghunathan and N. Jha. Behavioral synthesis for low power. In Proceedings of the International Conference on ComputerDesign, pages 318–322, Boston, MA, Oct. 1994.

[Ragh-95] A.Raghunathan andN. Jha. ILP formulation for lowpower based on minimizing switched capacitance during data path allocation. In Proceedings of the International Symposium on Circuits & Systems, 1995.

[Rama-09] Ramanarayanan, R., Degalahal, V. S., Krishnan, R., Kim, J., Narayanan, V., Xie, Y., Irwin, M. J., and Unlu, K, “Modeling soft errors at the device and logic levels for combinational circuits”, IEEE Transactions on Dependable and Secure Computing, 6(3):202–216. v, 11, 37, 38, 2009.

[Ray-87] Raymond, J. P.; Petersen, E. L.; “Comparison of Neutron, Proton and Gamma Ray Effects in Semiconductor Devices”, IEEE Transactions on Nuclear Science, pp 1621-1628, 1987.

[Reed-96] Reed R. A.; Carts, M.A.; Marshall, P.W.; Marshall, C.J.; Buchner, S.; La Macchia, M.; Mathes, B.; McMorrow, D.; “Single Event Upset cross sections at various data rates” IEEE Transaction on Nuclear Science, 1996, pp 2862-2867.

[Rodb-07] Rodbell, K.P.; Heidel, D.F.; Tang, H.H.K.; Gordon, M.S.; Oldiges, P.; Murray, C.E.; “Low-Energy Proton-Induced Single-Event-Upsets in 65 nm Node, Silicon-on-Insulator, Latches and Memory Cells”, IEEE Transactions on Nuclear Science, December 2007, pp 2474-2479.

[Roy-92] K. Roy and S. Prasad. SYCLOP: Synthesis of CMOS Logic for Low Power Applications. In Proceedings of the Int’l Conference on Computer Design: VLSI in Computers and Processors, pages 464–467, October 1992.

[Savo-91] H. Savoj, R. Brayton, and H. Touati. Extracting Local Don’t-Cares for Network Optimization. In Proceedings of the International Conference on Computer-Aided Design, pages 514–517, November 1991.

[Schm-90] Schmitt, J., R. Creasey, F. Gomez-Molinero, and J. Lafay. "The development of high performance flight computers for the European space programmes." Acta Astronautica 21, no. 6 (1990): 375-383.

[Seif-05] Seifert N., P. Shipley, M.D. Pant, V. Ambrose, B. Gill, “Radiation induced clock jitter and race”, Proceedings of IEEE International Reliability Physics Symposium, pp 215 – 222, 2005.

[Seif-01] Seifert, N.; Xiaowei Zhu; Moyer, D.; Mueller, R.;

Hokinson, R.; Leland, N.; Shade, M.; Massengill, L.; "Frequency dependence of soft error rates for sub-micron CMOS technologies", in Proceedings of IEDM technical Digest, pp 14.4.1-14.4.4, 2001.

[Seif-04] N. Seifert and N. Tam, "Timing Vulnerability Factors of Sequentials", IEEE Transactions on Device Materials and Reliability, Vol. 4, pp 516-522, Sep 2004.

[Sell-68] F. F. Sellers, H. MoYo, L. W. Bearnson. "Analyzing errors with the Boolean difference." IEEE Transactions on Computers, Vol 100, pp 676-683, 1968.

[Shen-92] A. Shen, S. Devadas, A. Ghosh, and K. Keutzer. On Average Power Dissipation and Random Pattern Testability of Combinational Logic Circuits. In Proceedings of the Int'l Conference on Computer-Aided Design, pages 402–407, November 1992.

[Siew-10] Sierawski, Brian D., et al. "Muon-induced single event upsets in deep-submicron technology." Nuclear Science, IEEE Transactions on 57.6 (2010): 3273-3278.

[Sika-13] M. D. Sika, A. Yazdanbakhsh, B. Kiddie, J. R. Ahlbin, M. Bajura, M. Fritze, J. Damoulakis and J. Granacki, "Applying Residue Arithmetic Codes to Combinational Logic to Reduce Single Event Upsets", to appear in Proceedings of Radiation Effects in Components and Systems, 2013.

[Spra-02] E. Sprangle and D. Carmean. Increasing processor performance by implementing deeper pipelines. In ISCA 29, pages 25–36, May 2002.

[Tan-94] C. H. Tan and J. Allen. Minimization of Power in VLSI Circuits Using Transistor Sizing, Input Ordering, and Statistical Power Estimation. In Proceedings of the Int'l Workshop on Low Power Design, pages 75–80, April 1994.

[Tiwa-93] V. Tiwari, P. Ashar, and S. Malik. Technology Mapping for Low Power. In Proceedings of the 30th Design Automation Conference, pages 74–79, June 1993.

[Tiwa-94] V. Tiwari, S.Malik, and A.Wolfe. "Power analysis of embedded software: a first step towards software power minimization". IEEE Transactions on VLSI Systems, 2(4):437–445, Dec. 1994.

[Tiwa-98] V. Tiwari, S. Malik, P. Ashar, "Guarded evaluation: Pushing power management to logic synthesis/design." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol 17, pp 1051-1060, 1998.

[Tsui-93] C-Y. Tsui, M. Pedram, and A. M. Despain. Technology Decomposition and Mapping Targeting Low Power Dissipation. In Proceedings of the Design Automation Conference, pages
68–73, June 1993.

[Wen-10] ShiJie Wen; Wong, R.; Romain, M.; Tam, N.; “Thermal neutron soft error rate for SRAMS in the 90nm–45nm technology range”, IEEE International Physics Symposium, pp 1036-1039, 2010.

[Sato-94] T. Sato, M. Nagamatsu, and H. Tago. Power and performance simulator: ESP and its application for 100MIPS/W class RISC design. In Proceedings of 1994 IEEE Symposium on Low Power
Electronics, pages 46–47, San Diego, CA, Oct. 1994.

[Sava-86] Savaria, Yvon, N. C. Rumin, J. F. Hayes, and V. K. Agarwal. "Soft-error filtering: A solution to the reliability problem of future VLSI digital circuits." Proceedings of the IEEE 74, no. 5 (1986): 669-683.

[Shiv-02] Shivakumar P., Kistler M., “Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic”, Proceedings of the International Conference on Dependable Systems and Networks, pp 389- 398, 2002.

[Shiv-02(2)] P. Shivkumar, “The optimal logic depth per pipeline stage is 6 to 8 FO4 inverter delays”, Proceedings of the 29th annual international symposium on Computer architecture, pp 14-24, 2002.

[Seif-02] N. Seifert, X. Zhu, and L. W. Massengill, “Impact of scaling on soft error rates in commercial microprocessors,” IEEE Trans. Nucl. Sci., vol. 49, no. 6, pp. 3100–3106, Dec. 2002.

[Seif-12] N. Seifert, B. Gill, S. Jahinuzzaman, J. Basile, V. Ambrose, Q. Shi, R. Allmon, and A. Bramnik, “Soft error susceptibilities of 22 nm tri-gate devices,” IEEE Trans. Nucl. Sci., vol. 59, no. 6, Dec. 2012.

[Seif-01] N. Seifert, X. Zhu, D. Moyer, R. Mueller, R. Hokinson, N. Leland, M. Shade, and L. Massengill, “Frequency dependence of soft error rates for sub-micron CMOS technologies,” in Proc. Int. Electron Dev. Meeting, Washington, DC, DC, Dec. 2001, pp. 14.4.1–14.4.4.

[Srin-05] Srinivasan, V., A. L. Sternberg, A. R. Duncan, W. H. Robinson, B. L. Bhuva, and L. W. Massengill. "Single-event mitigation in combinational logic using targeted data path hardening." IEEE transactions on nuclear science 52, no. 6 (2005): 2516-2523.

[Stan-94] M. Stan and W. Burleson. Limited-weight codes for low-power I/O. In Proceedings of the Int’l Workshop on Low Power Design, pages 209–214, April 1994.

[Su-94] C. L. Su, C. Y. Tsui, and A. Despain. Saving power in the control path of embedded processors. In IEEE Design & Test of Computers, pages 24–30, Winter 1994.

[Sven-94] C. Svensson and D. Liu. A power estimation tool and prospects for power savings in CMOS VLSI chips. In Proceedings of the International Workshop on Low Power Design, pages 171–176, Napa, CA, Apr. 1994.

[Wall-62] J. T. Wallmark and S. M. Marcus, “Minimum size and maximum packing density of non-redundant semiconductor devices,” Proc. IRE, pp. 286–298, Mar. 1962.

[Wang-08] F.Wang and V. Agarwal, “Soft error rate determination for nanometer CMOS VLSI logic”, Proceedings of 40th Southeastern Symposium on System Theory, 2008, pp. 324–328.

[West-09] Weste, Harris, Banerjee, “CMOS VLSI Design: A Circuits and Systems Perspective”, Pearson 2009.

[Zhou-04] Q. Zhou and K. Mohanram, “Cost-effective radiation hardening technique for combinational logic,” in Proc. Int. Conf. Comput. Aided Design, Nov. 2004, pp. 100–106.

[Zoel-08] Zoellin, Christian G., H. Wunderlich, Ilia Polian, and Bernd Becker. "Selective hardening in early design steps." In Test Symposium, 2008 13th European, pp. 185-190. IEEE, 2008.