This is to certify that the
dissertation entitled

EFFICIENT TECHNIQUES FOR MODELING AND MITIGATION OF SOFT ERRORS IN NANOMETER-SCALE STATIC CMOS LOGIC CIRCUITS

presented by

Srivathsan Krishnamohan

has been accepted towards fulfillment
of the requirements for the

Ph.D. degree in Electrical and Computer Engineering

Major Professor's Signature

12/16/2005

Date

EFFICIENT TECHNIQUES FOR MODELING AND MITIGATION OF SOFT ERRORS IN NANOMETER-SCALE STATIC CMOS LOGIC CIRCUITS

By

Srivathsan Krishnamohan

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering

2005

ABSTRACT

EFFICIENT TECHNIQUES FOR MODELING AND MITIGATION OF SOFT ERRORS IN NANOMETER-SCALE STATIC CMOS LOGIC CIRCUITS

By

Srivathsan Krishnamohan

Soft errors are changes in logic state resulting from the latching of single-event transients (SETs) caused by high-energy particle strikes or electrical noise. Due to scaling of minimum feature size, supply voltage, and clock frequency, soft error rate (SER) is expected to increase by several orders of magnitude in combinational and sequential logic circuits in the near future. In this dissertation, we address the following three important issues related to logic soft errors: (1) modeling of SETs generated in combinational logic blocks (CLBs), (2) efficient design techniques to reduce the SER of CLBs, and (3) analysis and design of soft-error hardened latches.

Our main contributions in modeling and mitigation of soft errors in logic circuits are as follows. (1) A fast and accurate lookup-table (LUT) based approach to estimate SET width, which is necessary to gauge the effectiveness of time-redundancy based SER mitigation techniques. The LUT provides more than 1000 times speedup over HSPICE simulations and has less than 10% error compared to existing techniques which have 15% or more error, without significantly increasing LUT size. (2) An efficient and systematic error masking (EM) technique that samples selected non-critical primary outputs (POs) of a CLB three times using delay-chain-generated control signals and then majority votes on them within the slack available in a cycle to mask errors. Hence, it incurs no performance overhead, does not perform redundant

computation, and can mask SETs of width less than half the slack available at a PO. The average SER reduction from EM on ISCAS85 circuits is 82.67%. Other significant features of this technique include: (a) efficient triple sampling and majority voting and (b) exploitation of circuit timing dependence upon input vector and using non-uniform slack passing/borrowing in pipelined circuits to further reduce SER. (3) A method that supports error masking plus efficient error detection and recovery (EM+EDR) and is suitable for CLBs with a small fraction of non-critical POs. In this case, EM is applied to POs with sufficient slack and EDR to critical or near-critical POs. EM+EDR can tolerate SETs with width up to half the clock period and provides an average SER reduction of 93.78% on ISCAS85 circuits. When a soft error occurs, a very low-likelihood event for an application run, and is detected, EM+EDR recovers from it within a single clock cycle. (4) Design of an efficient and robust delay chain to produce phase-shifted clock signals for use in our EM/EM+EDR techniques. (5) Finally, a comprehensive analysis of a number of existing soft-error hardened latch designs using a variety of metrics and some new designs, the best of which is vulnerable to only single-event multiple upsets, and which has a delay overhead of 12% and consumes only 70% power compared to a standard latch.

Our SER mitigation work represents a significant advancement over previous approaches which, in contrast, rely on introducing explicit hardware or time redundancy or on redundant computation, often both. Consequently, our methods provide substantial energy and performance/hardware advantages while significantly reducing logic SER.

Acknowledgements

It has been an enriching experience working with my advisor over the course of my Ph.D. and also interacting with him during the courses he has taught me. I deeply appreciate his patience with me and being very liberal with his time while I was looking around for a dissertation problem, as I used to discuss with him in every meeting a new topic that I had got excited about. His positive outlook, view that every problem is solvable, and attention to detail are things that I have admired, and have tried to imbibe some of these qualities in myself. His insistence on concise and precise writing, whose importance though initially lost on me, has led me to much learning of this skill.

I would like to thank Professors Anthony Wojcik, Michael Shanblatt, Peixin Zhong, and Shantanu Chakrabartty for consenting to be on my Ph.D. committee. I'm also grateful to them for being very co-operative with the scheduling of my proposal and dissertation defense and for reviewing my dissertation at short notices. Their valuable technical suggestions during my proposal defense, and their advice on the importance of good writing helped me to improve this final dissertation. Also, the courses taught by Professor Wojcik on computer architecture and Professor Chakrabartty on low-power mixed signal design helped broaden my knowledge in these areas.

I would like to thank Dr. Rajeev Murgai for providing me an opportunity to do my internship at Fujitsu Labs, which has been a valuable experience and has helped me to improve the dissertation research tremendously. My interaction with Dr. Rajeev Murgai taught me the rigors of industrial research and also showed that *"Most of us*

can easily do two things at once; what's all but impossible is to do one thing at once"[1]. He has been more than a technical mentor to me, and my interaction with him has helped me to develop many positive qualities (least of which is my fore hand smash in ping-pong!). Also, I would like to thank William Walker for his guidance and funding during my internship, Subodh Reddy for helping me to improve my coding skills and making my stay in and outside of Fujitsu labs enjoyable. I'm grateful to Dr. Dipesh Patel, Stuart Biles, and Dr. Daryl Bradley for offering me the opportunity to investigate my ideas with them in ARM R&D during Summer 2005, and for their invaluable inputs. Also, I appreciate Dr. Dipesh Patel's kindness in offering to extend my internship at their Sunnyvale office.

Also thanks are due to the other members of my lab Krishnan, Gandhi, Sandeep, for making my stay at MSU memorable and for all the fun we had over the last two years. It was a pleasure working on my Ph.D. along side these guys and the productive working atmosphere provided by them made my stay in the ACAC lab enjoyable.

My experience at STMicroelectronics was an enriching one which gave me a deeper insight into the field of chip design and one which developed me professionally. My colleagues Murthy, Paolo, Wreeju, Arvind, Vijay and others in CMG-MCD made my ST years memorable. My colleague Vijay has been a great friend since my STMicro days and I'm deeply grateful for his kindness in providing all sorts of help and for his enjoyable company whenever I have visited Bay area. I am also grateful to Shanker and my other undergraduate friends from BITS Pilani in Intouch, for their great

[1]Mignon McLaughlin, The Second Neurotic's Notebook, 1966

company and support during my BITS years and through my graduate study. Many thanks also to Dr. Sridhar from SUNY Buffalo, whose words of encouragement and advice have helped me in many of my endeavors.

My brother Srikanth has been a beacon through my life. The confidence he instilled in me early on and his constant motivation through the Ph.D. have been a big driving force for my timely completion of the dissertation. Also my parents have helped me reach me this far by instilling in me right from the beginning that *"The virtue of all achievement is victory over oneself"*, and by supporting me in whatever endeavor I have taken up. This dissertation is dedicated to my parents and brother for their support and affection all these years. None of this would have been possible without the guidance and inspiration from my brother Srikanth and my parents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Designers strive to deliver ever-higher performance systems cost-effectively by leveraging technology scaling to meet end-user application needs. However, with unprecedented levels of device integration ($\sim 10^9$ transistors/chip) and scaling of minimum feature size ($\sim$ 10s of nm), clock frequency ($\sim$ 10s of GHz), and supply voltage ($V_{DD} < 1$ V), the transient fault or soft-error rate (SER) of logic circuits is becoming a dominant reliability challenge even in commodity processors. *Soft errors* are changes in logic state resulting from the latching of *single-event transients* (transient voltage fluctuations at a logic node or SETs) caused by electrical noise or external radiation. Unlike *hard errors* (arising from, say, electromigration, hot carrier effects, or dielectric breakdown), they do not result in permanent damage of components. In this dissertation, we are concerned with static CMOS circuit soft errors. Although most of our discussion and mitigation techniques apply to soft errors due to either source (i.e., electrical noise or external radiation), our focus is on radiation-induced errors, particularly, those resulting from high-energy neutron strikes. This is because, as ex-

plained in Sec. 1.1, high-energy neutron strikes represent the most important source of soft errors and their effects are well modeled, allowing us to accurately analyze the effectiveness of our soft error mitigation techniques. Background on radiation-induced soft errors and a brief discussion of our contributions relative to previous work follows next.

1.1 Radiation-Induced Soft Errors: Causes, SET Generation, and SER Calculation

Soft errors are caused by electrical noise (e.g., due to crosstalk and IR or Ldi/dt supply noise), electromagnetic interference, and external radiation, with the latter being the most important source and our main focus. The operating environment of a semiconductor chip contains background radiation from cosmic rays, low energy thermal neutrons, and radioactive traces present in chip packaging material. These radiations comprise electrons, protons, neutrons, pions, muons, alpha, and other particles. The following two types of effects have been observed from these radiations.

- **Total dose effects (TDEs)**: These result from the interaction of ionizing radiation with device materials. TDEs can cause changes in transistor threshold voltage and decrease the mobility of carriers in the channel, and hence the transconductance and gain of a transistor. Gain degradation results in increased propagation delay of a gate. TDEs cause changes in the transistor and circuit characteristics over a long period of time, which do not lead to failure in commodity chips. Hence, they are not considered in this dissertation.

- **Single event effects (SEEs)**: These result from the interaction of a high-energy particle passing through a device. SEEs can cause single event upsets (SEUs) or single event latchups (SELs). SEUs are reversible bit-flips in a latch or memory element that change the logic state of a circuit. The change in the logic state of a circuit due to SEUs are called soft errors in contrast to hard errors which are irreversible. Soft errors are reversible since their effects can be removed by resetting or rewriting the memory elements. SEL occurs when the injected charge activates the parasitic PNPN structure that exists in bulk CMOS transistors. SEL is not a threat to SOI devices and can be prevented in bulk CMOS by using thin epitaxial layers or guard rings [2].

Various ionizing particles generate electron-hole pairs through different mechanisms. An alpha particle ionizes the atoms in a chips's substrate through electromagnetic force between itself and the valence electrons. However, high-energy cosmic-ray neutrons and protons collide with nuclei within silicon substrate and generate secondary particles capable of ionizing silicon atoms. A neutron has to encounter, on an average, 10^{10} atoms before it hits a nucleus. Based on the density of silicon (which is 2.3 gm/cm^3) and the absorption length for neutrons, the average distance a neutron has to travel in silicon substrate before it hits a nucleus is on the order of tens of centimeters. This explains the fact that neutrons, which have enough energy to pass through shields that are hundreds of centimeter thick, do not easily cause logic upsets at multiple nodes in a circuit block.

The secondary particles generated by neutrons traverse and ionize the silicon substrate, in the process creating a track of excess electron-hole pairs, with average ionization energy (i.e., the average energy needed of the particle to ionize a silicon atom) being 3.6 eV/electron-hole pair in silicon. Some of these electron-hole pairs can eventually reach a reverse-biased *pn* junction of a sensitive node, such as a gate output as shown in Figure 1.1. When that happens, the majority carriers are reflected from the depletion region of the *pn* junction while the minority carriers are swept across the junction through the drift mechanism. This causes a net current across the depletion region that can either charge or discharge a logic node. For example, electrons created near the drain node of an NMOS transistor connected to V_{DD} pull the drain node to GND. In the case of static nodes, the electrons recombine with holes shortly thereafter, and the drain node is charged back to V_{DD}. For static nodes, this creates a transient glitchy pulse called *single-event transient (SET)*, whose amplitude and duration depend upon the strength of the driving gate and the output capacitance that it drives. Recently, *Sun* and *Intel* announced that chip packages with alpha-particle emitting lead have been minimized. Therefore, high-energy cosmic rays and the neutrons present in them are expected to become the primary source of soft errors in the future. The methodology and schemes we propose in this dissertation are applicable to soft errors caused by sources other than neutrons too. But, we evaluate most of our proposed schemes based on radiation-induced errors, particularly, those resulting from high-energy neutron strikes, since SET generation mechanisms due to neutron strikes have been studied for a long time and are well-understood. As a result, sufficient information for calculating SERs of CMOS process technologies are available

for neutron strikes.

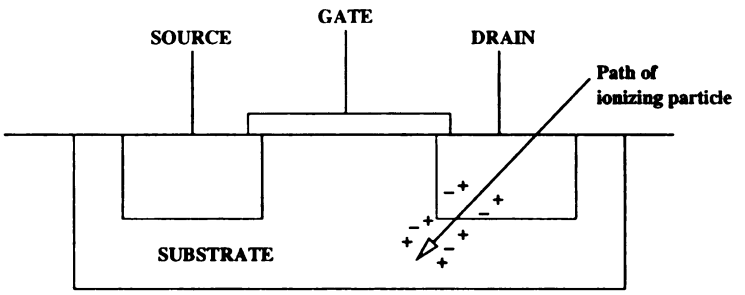

Figure 1.1. Figure shows an ionizing particle passing through a silicon substrate generating holes and electrons. SET is generated when the holes and electrons collect around a *pn* junction of a drain node as shown.

The basic or raw soft-error rate (SER) of CMOS circuits due to cosmic ray neutrons can be calculated using the following equation from [3]:

$$SER(Q_{crit}) = K \times F \times A \times e^{(\frac{-Q_{crit}}{Q_s})}, \tag{1.1}$$

where K is a technology-independent constant, F is the neutron flux, A is the sensitive device area, Q_{crit} is the critical charge, and Q_s the charge collection slope for the technology, which is strongly dependent on doping and supply voltage. The *critical charge* Q_{crit} is defined as the minimum charge required to cause a logic upset in memory elements, and in logic circuits, it is the minimum charge which generates SETs that can change the value stored in a latch or flip-flop. The sensitive device area is equal to the sum of node areas where charge collection can lead to SET generation in logic gates, or logic upset in latches and memory elements.

1.2 SET Propagation, Masking, and Latching in Logic Circuits

Soft errors can occur in both sequential circuits (latches and flip-flops) and combinational logic blocks (CLBs). A particle strike causing a single event upset (SEU) at a latch, when the latch is in hold mode, results in an error. However, SEU at a gate leads to generation of a single-event transient (SET). SETs are transient voltage fluctuations, such as a 1→0 or 0→1 logic flip, occurring at a gate output. Such SETs, apart from being generated by radiation, can also occur due to electrical noise such as crosstalk and IR and Ldi/dt supply noise. Soft errors in CLBs result from SETs changing the value stored in memory elements, such as latches, flip-flops, or register files. For a SET originating in CLB to cause a soft-error, it must propagate to a primary output (PO) gate and be finally captured by an output latch. However, a soft error will not occur if the SET is either: (1) *logically masked*: some other input of a gate in the SET propagation path determines its output instead of the SET; (2) *timing window masked*: the SET does not arrive around the closing clock edge and hence is not captured by the output latch; (3) *electrically masked:* the amplitude of the SET is not sufficient to cause a state change at one of the propagating gates or at the output latch.

In a static CMOS circuit, since the SET can get latched only if it arrives around the clock closing edge, soft errors in combinational circuits are synchronous. Synchronous soft errors can be further classified into the following two types [4].

1. Delay faults: These are caused by transient pulses with width smaller than

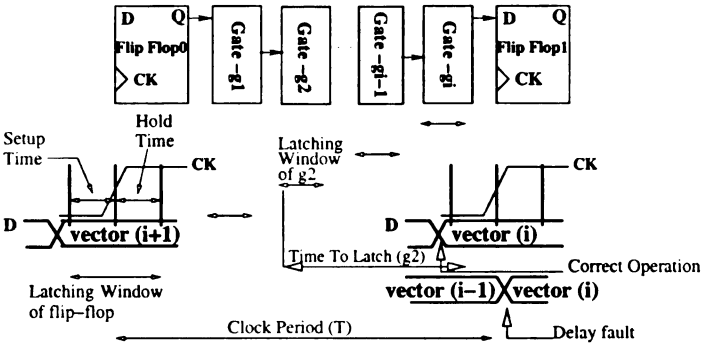

Figure 1.2. Delay fault and latching window of gates in a logic circuit. The figure shows both correct and incorrect operation of a logic circuit. During correct operation, the data corresponding to vector i transitions a setup time before the clock closing edge. Incorrect operation results due to a delay fault, which pushes the data D of flip-flop1 to transition after the clock closing edge. Also, the figure shows latching window of gates in a pipeline stage. Latching window of gates located farther from the PO occurs earlier in the clock cycle time.

the latching window time (sum of latch setup and hold times). Such transients cause an error by delaying the arrival of the PO signal at the output latch. This leads the PO to transition after the setup time as shown in Figure 1.2.

2. Functional faults: These occur when the transient pulse is wider than and overlaps the latching window of the output latch. The value stored in the output latch gets flipped, which changes the logic state.

An SET has two properties associated with it that determine whether it gets latched at the primary output:

1. Spatial: whether the SET originates on a critical or non-critical path of the CLB.

2. Temporal: whether the SET originates at a gate output before or after the output has settled.

These two properties strongly influence the likelihood of an SET not getting masked due to the latching window effect. Each gate has a latching window as shown in Figure 1.2, during which time the error pulse has to pass through the gate in order not to get latching window masked. The latching window of a gate extends approximately from $t_{\text{clk,edge}} - t_{\text{setup}} - t_{\text{TTL}}$ to $t_{\text{clk,edge}} + t_{\text{hold}} - t_{\text{TTL}}$, where $t_{\text{clk,edge}}$ is the time of clock closing edge, t_{setup} and t_{hold} are the setup and hold times of a latch, respectively, and t_{TTL} is the *time to latch* or the propagation delay from the gate output to a latch. Any SET with width w passing through the gate has to completely overlap this latching window to cause a logic flip at the output latch. Latching window of a gate can begin at or after the time the actual output of the gate passes through it. For gates on a critical path, the latching window overlaps with the time when actual gate output settles, while for gates on non-critical paths, latching window occurs after actual gate output settles. An SET generated at any gate before its output settles would get latching window masked, unless its pulse width extends to overlap the complete latching window of the gate.

The SER of a system is usually measured in failures in time (FITs) or mean time between failures (MTBF). *FIT* is defined as the number of failures in one billion hours of operation, while *MTBF* is the mean time between two successive failures. For example, an MTBF of 1000 years equals a FIT rate of 114 ($10^9/(24 \times 365 \times 1000)$). A fault-tolerant system with infinite MTBF corresponds to zero FIT. FIT is

more commonly used by VLSI designers because it is additive (i.e., the FIT rate of a system is obtained by adding the FIT rates of individual components), unlike MTBF.

1.3 SER Scaling Trends for Combinational Logic, Latches, and Memories

The scaling of SER for memories, latches, and combinational logic differ with advancing process generations; minimum physical gate length is used to demarcate different process generations. As can be seen from Eqn. 1.1, SER is: (i) linearly proportional to the sensitive device area A and (ii) exponentially dependent on the ratio Q_{crit}/Q_s. When Q_{crit}/Q_s is close to one, SER scaling is dominated by the sensitive device area. The sensitive device area decreases quadratically with shrinking feature size and, based on scaling trends, is 50% smaller compared to that for the immediately previous process generation. The critical charge depends only on the charge stored ($Q_{stored} = C \times V_{DD}$) at a dynamic node, and on both Q_{stored} and the charge dissipation capability at a static node. Critical charges are decreasing nonlinearly with each process generation due to diminishing Q_{stored}. This can be seen from the *International Technology Roadmap for Semiconductors (ITRS)* prediction of supply voltage and clock frequency scaling shown in Figure 1.3. The supply voltages for high-performance and low-power applications are expected to drop to 0.7 V and 0.5 V, respectively, at the 7 nm technology node, while the clock frequency scales to 55 GHz. This would lead to significant reduction in Q_{crit}, which implies particles of lower energy, with higher flux, can cause SEUs. In addition, it has been experimentally verified that SER of logic circuits increases linearly with clock frequency [5].

Thus, decreasing V_{DD} and increasing clock frequency lead to higher SERs in future technologies.

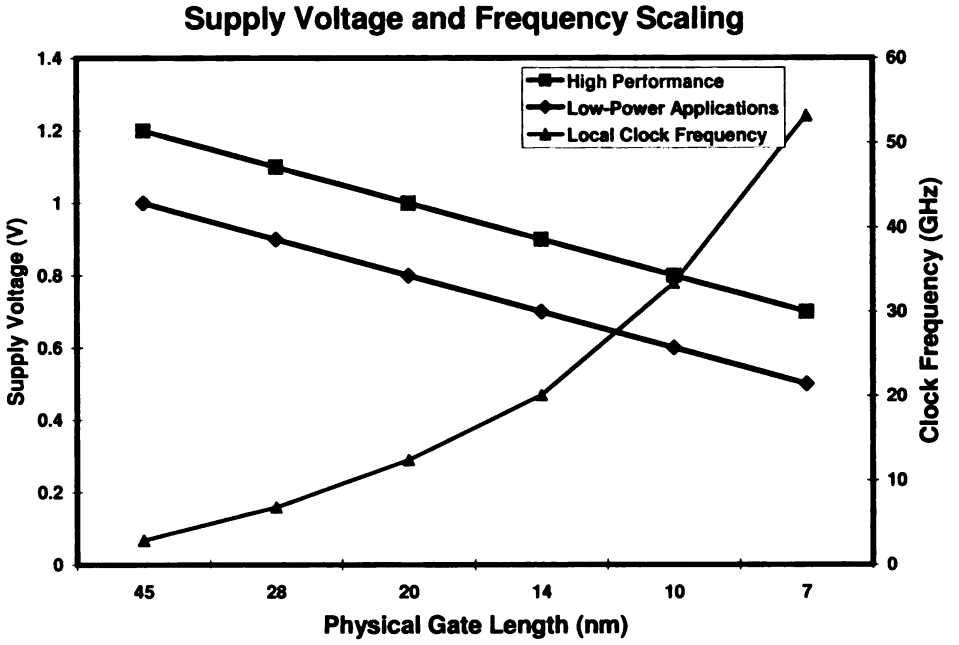

Figure 1.3. Long-term estimates from ITRS for the supply voltage, and clock frequency of DRAMs, microprocessors, and ASICs used in high performance and low-power applications [1].

The charge collection efficiency Q_s scales approximately linearly with device size in a log-log scale [3]. The value of Q_{crit} and Q_s for both logic and memories for different technology nodes, reproduced from [6], is given in Table 1.1. The first five rows give the critical charge for combinational logic, latches, and SRAMs in femto Coulombs (fC). The last row gives the value of charge collection slope in fC for different process generations. The value of Q_{crit} for latches and SRAM cells becomes smaller than Q_s at 100 nm and 180 nm technology nodes, respectively, and hence as feature size and the device area scale down, the SER of memory elements reduces. However, in logic gates, Q_{crit} is more than Q_s, due to which a small decrease in critical charge increases

SER by orders of magnitude.

Circuit Element	**600 nm**	**350 nm**	**250 nm**	**180 nm**	**130 nm**	**100 nm**	**70 nm**	**50 nm**
Q_{crit} of Logic: 16 FO4s	N/A	676	489	250	116	61.3	24.0	10.40
Q_{crit} of Logic: 4 FO4s	4160	509	336	131	63.9	35.2	16.0	7.02
Q_{crit} of Logic: 0 FO4s	1130	386	265	99.3	48.8	27.3	13.2	5.57
Q_{crit} of Latches	360	120	82.4	31.9	15.0	7.96	3.73	1.66
Q_{crit} of SRAM	146	48.8	33.7	12.9	6.31	3.43	1.52	0.67
Q_s	52.3	34.6	26.8	17.2	12.2	9.53	7.19	5.54

Table 1.1. Q_{crit} and Q_s in fC of combinational logic, latches, and SRAMs for different technology nodes.

In current technologies, soft-error contribution of latches and SRAMs far exceeds the soft-error contribution of CLBs. Mitra and others estimate the contribution of combinational logic, latches, and unprotected SRAM for a commercial state-of-the art processor to be 11%, 49%, and 40%, respectively [7]. But as technology scales and clock frequencies increase, SET from a CLB has a higher likelihood of latching because of diminishing Q_{crit} and latching-window probability. This is expected to make SER contribution of logic much more than that of memories. Shivkumar and others have shown that SER per chip of combinational logic circuits will increase nine orders of magnitude when minimum feature size scales from 600 nm to 50 nm, becoming comparable to SER per chip of unprotected memory elements [6]. In addition to logic SER scaling, the SER per latch or SRAM bit is expected to stay the same or decrease in future technology generations [8, 9]. Although the critical charge per latch or SRAM bit decrease, smaller cross section area of these devices reduces the probability of SEU occurring. However, due to the increasing number of latches and SRAM nodes per chip, in the future, the contribution of memory elements to total

chip SER is estimated to increase, but at a rate lower than that of combinational logic circuits [10].

The advances made in semiconductor process technology have reduced manufacturing related failures in chips. The trends observed in permanent fault rates for microprocessors, SRAMs, and DRAMs are shown in Figure 1.4 [11]. This plot shows that the FIT rate due to permanent faults has clearly decreased over the period 1990-2001. In the case of 256 KB SRAM, the FIT rate has fallen by six times reaching 50 in 2001 from an initial value of 300 in 1990. This rapid decline in permanent fault FIT rates and technology scaling are expected to make soft errors very critical in sub-100-nm designs. In fact, soft-errors, if unmitigated, are expected to become the primary source of failures for sub-90 nm chips, causing a failure rate of 50,000 FITs, exceeding that of all other reliability mechanisms combined [12].

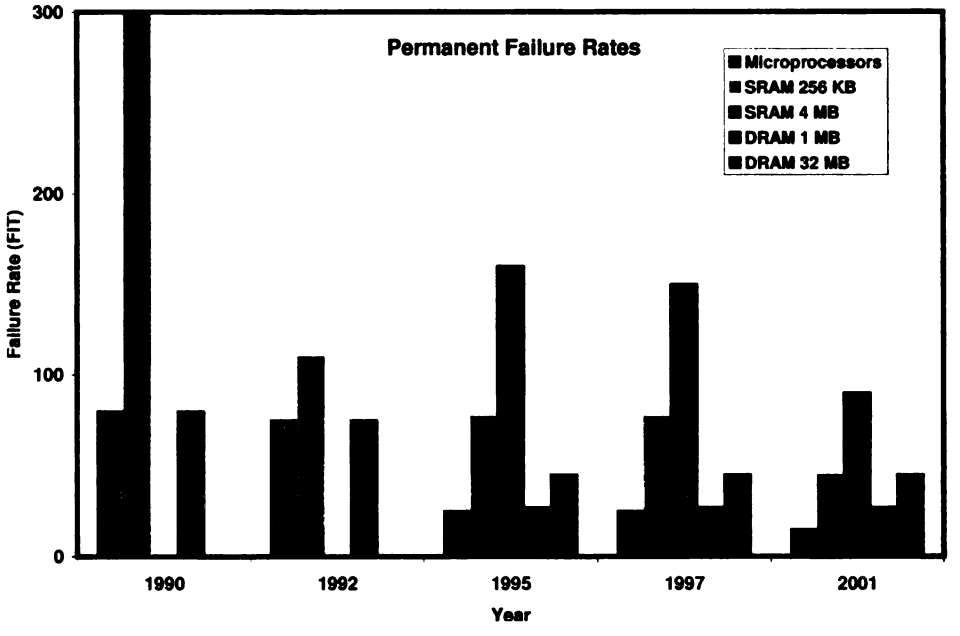

Figure 1.4. Permanent fault FIT rates for microprocessors, SRAMs, and DRAMs [11].

The impact of soft errors at the system level has been reported in diverse applications, ranging from satellites to sea-level computer systems, a few of which are summarized next. A detailed historical review of experiments done by *IBM* on radiation-induced soft fails and failures observed in memories due to soft errors, in the time period 1978-1994, is reported in [13]. This paper reports all the major memory suppliers of the time, such as *Intel*, *IBM*, and *Hitachi*, observing soft failures in their memories due to both alpha particles and cosmic rays. As early as 1978, evidence of sea-level soft fails on 16 Kb DRAMs, from alpha particles present in the memory packaging materials, was provided by May and Woods of *Intel* [14]. Similarly, in 1980, *Hitachi* announced that some of their bipolar RAMs failed under alpha-particle bombardment [13]. In addition, in 1981, *IBM* discovered reliability problems with their 16 Kb DRAM memory chips. *Sun Microsystems* also observed cosmic ray strikes on unprotected cache memories causing random crashes at major customer sites in its Enterprise server line [15].

In addition to the memory errors reported above, *Fujitsu* announced that they have protected 80% of the 200,000 latches in their fifth generation SPARC64 processor fabricated in 130 nm SOI CMOS [16]. Other logic errors in processors have been reported by *iROC Technologies* when they performed radiation testing on an 8-bit logic core processor called ROC-CR11 [7]. The processor was manufactured for French space agency *CNES* in 180-nm technology. During radiation testing, they observed eleven logic errors in the processor datapath.

The increase in logic SER not only affects high-performance microprocessors (HPMs), such as Pentiums, Opterons, SPARCs, and Power PCs used in worksta-

tions and enterprise servers, with core frequencies greater than 1 GHz and with more than 100 million transistors on-chip, but also embedded processors (such as those from *ARM*, *MIPS Technologies*, and *Tensilica*) used in consumer, automotive and networking applications [17]. The push for tackling soft errors in HPMs is due both to application factors and the large number of chip transistors. In the case of embedded processors, application factors, increase in the number of transistors per chip, and the number of chips used in a system (such as a mobile phone) are expected to play an important role in scaling the SER of these systems. For example, when a semiconductor vendor ships a million units of a product with 100 components in each product, the total FIT over the entire shipment would be hundreds of errors every few hours. The number of product recalls due to soft errors would lead to significant loss in the vendor's revenues and reputation.

1.4 Our Contributions

In this dissertation, we address the following three important problems posed by the dramatic increase in logic SER: (1) modeling of SETs generated in CLBs, (2) efficient design techniques to reduce the SER of CLBs, and (3) analysis and design of soft-error hardened latches. These issues and how we have addressed them are briefly outlined and compared to previous work next.

1.4.1 Logic Circuit SER Estimation

Tackling SER along with various conflicting nanometer objectives of power, performance, and area increases design cost, which has been identified as an important

factor with the potential to limit the semiconductor roadmap. Calculation of logic SER is essential to devise efficient SER mitigation techniques and can be used to optimize the different conflicting nanometer objectives, such as power, performance, and reliability. It can also be used to isolate and apply the design techniques to the most vulnerable circuitry.

The SER of SRAM caches can be calculated by determining the Q_{crit} of each cell from SPICE simulation and using it in Eqn. 1.1. However, estimating the critical charge of a combinational logic gate requires the width of the SET generated at a gate output. We describe a fast and accurate lookup-table (LUT) based methodology to calculate both SET width due to particle strikes and the SER reduction that can be obtained with time-redundancy based mitigation techniques. Previous techniques for SET width calculation use complex expressions or large LUTs and have greater than 15% error for inputs not close to pre-characterized points. We study the sensitivity of an SET to various gate and circuit characteristics and determine the parameters to be used, their spacing, and their lower and upper-bounds for constructing the LUT. The proposed LUT uses non-uniform spacing and surface-based interpolation between its indices to obtain the SET width generated at a gate and primary output. It provides more than 1000 times speedup over HSPICE simulations and has less than 10% error compared to existing techniques which have 15% or more error.

1.4.2 Efficient Soft-Error Mitigation Techniques for Combinational Logic

For memories, due to their regular array structures, efficient soft-error detection and correction techniques have been developed. Commonly used techniques in memories are error correcting codes or parity codes for detection. Using these techniques to protect combinational logic circuits requires high cost due to their irregular structure. Other techniques for logic soft-error protection, such as triple modular redundancy (TMR) and RE-computing with triplication and voting, rely on explicit spatial or temporal redundancy. However, most of these techniques suffer from significant shortcomings such as: (1) they are meant primarily for error detection only; (2) they are applicable only to specific classes of circuits such as arithmetic units; and/or (3) they incur high power, performance, and area overheads. This necessitates an efficient design approach that would make logic circuits used in commodity as well as other applications soft-error resilient without adversely affecting other design considerations such as power and performance.

We propose an efficient and systematic error masking (EM) technique that can be applied to combinational logic circuits which have a significant fraction of non-critical primary outputs (POs) with sufficient slack. This error masking technique prevents an SET pulse of width less than approximately half of the slack available in the propagation path from latching and turning into a soft error, without any performance overhead. Previous techniques incur a performance overhead of $2W$ for masking an SET pulse of width W. We perform error masking only at PO flip-flops

with sufficient slack, which ensures that the delay increase caused by the addition of majority voter and control transistors to the flip-flops does not affect the timing of the circuit. Additionally, our technique uses a single delay chain to produce phase-shifted signals and sample POs of a CLB. The results obtained on ISCAS85 benchmark circuits show an average SER reduction of 82.67% from the original unprotected circuit.

For CLBs with a small fraction of non-critical POs, we proposed a method that supports error masking plus efficient error detection and recovery (EM+EDR). In this case, EM is applied to POs with sufficient slack and EDR to critical or near-critical POs. EM+EDR can tolerate SETs with width up to half the clock period and provides an average SER reduction of 93.78% on ISCAS85 circuits. When a soft error occurs, a very low-likelihood event for an application run, and is detected, EM+EDR recovers from it within a single clock cycle.

1.4.3 Robust Delay Chain Construction

An important component of our EM and EM+EDR techniques is a delay chain used to generate phase-shifted clock signals for sampling POs. Delay chains are also used in a variety of other applications too. With technology scaling, sub-90 nm process technologies are introducing increasing variations in designs. This process variation leads to delay uncertainty. Therefore, delay chains need to be constructed with robust delay elements. We analyze three different families of delay elements in terms of their robustness to process variation, and then determine the appropriate delay

element for delay chain construction. The three different delay element families are: (1) transmission gate based, (2) cascaded inverter based, and (3) voltage-controlled ones. We compare the delay element's effectiveness in terms of yield, which is defined as the number of circuits within the specified delay range. The delay variations are obtained through HSPICE Monte Carlo simulations and the delay sensitivity to different process and environmental variations are studied using simulation results. A design methodology used to construct a delay chain that produces control signals phase shifted from the system clock by every 200 ps is presented. Finally, construction of a buffer chain with least delay to distribute the phase shifted clock signals based on the logical effort method is explained. This work will help designers to construct robust delay elements and chains.

1.4.4 Hardening of Latches for Soft Errors

Many different latch designs to prevent soft errors due to particle strikes on the latch nodes have been proposed. We analyze and compare these designs based on some existing and new metrics. This work will help designers to select latches for applications where soft errors are an important design metric. We also propose new latch designs, the best of which is vulnerable only to SEMUs with a delay overhead of 12% and power consumption of 70% compared to a standard transmission gate latch. The proposed latches can also be customized in accordance with application requirements for power consumption, performance, and soft-error resilience. In addition, some of the proposed latch designs can also be used to protect CLBs.

1.5 Dissertation Outline

The remainder of the dissertation is organized as follows. Chapter 2 presents the LUT-based model for determining the width of SET generated at a gate output. The sensitivity of SET width to different circuit and striking particle's parameters are studied and then appropriate LUT indexes are determined. The lower and upper bound for each LUT parameter, and the number and interval between LUT indices are determined based on accuracy requirements. This chapter also explains interpolation performed within the LUT and the accuracy of the SET width estimated. The LUT-based model is further extended in chapter 3 and used to calculate the SER reduction obtained using the time-redundancy techniques presented in the next two chapters.

In the next three chapters, we present our work on soft-error mitigation of combinational logic circuits. In Chapter 3, we first review existing methods to mitigate soft errors in combinational logic circuits and their drawbacks. Then our error masking technique is explained and the SET width tolerated is determined. Finally, the methodology used to calculate the SER of the original and the error-masked circuits and results for SER reduction are presented. Next, Chapter 4 describes how the error masking technique can be improved by combining it with error detection and recovery in critical and near-critical paths. The pulse width tolerated using the EDR technique and the SER reduction obtained are calculated. Later, steps to improve the SER reduction obtained from the proposed error masking technique by increasing the slack available in the CLBs are presented: first, by exploiting critical path delay dependence upon the input vector and second, through slack redistribution in

pipelined circuits. Finally, Chapter 5 discusses construction of the delay chain used to generate phase-shifted clock signals for PO sampling in our EM and EM+EDR techniques. Robustness to process variation and power consumption of different delay elements are studied and guidelines for selecting appropriate delay elements for delay chain construction are given.

Chapter 6 analyses existing soft-error hardened latches and studies the trade-offs involved in using these latches to protect CLBs. Some new latch designs that are only SEMU vulnerable, and which can be customized based on application requirements are also presented.

Finally, Chapter 7 summarizes the important contributions of this dissertation and discusses directions for future research.

CHAPTER 2

Modeling and Analysis of Soft Errors in Logic Circuits

Soft error rate estimation of logic circuits requires accurate and efficient estimation of electrical, logical, and temporal masking effects. To evaluate the electrical masking of a path, the amplitude and duration (AD) of an SET generated at a gate output, due to a particle strike, needs to be calculated. Here, amplitude refers to peak voltage, and duration is the width of SET measured at $V_{DD}/2$. The width and amplitude of an SET, for a specific charge collected, depends on gate drive strength, output load capacitance, supply voltage, and shape of the current waveform. For an SET with amplitude greater than $V_{DD}/2$, the duration of the pulse determines the SET width during propagation. Hence, for all SETs which reach a voltage greater than $V_{DD}/2$, we approximate the amplitude to be V_{DD}. Further, this approximation yields the minimum charge that can cause a soft error. In this chapter, a methodology to calculate the SET width as a function of the charge collected at a gate output is

proposed. In Chapter 3, we extend this methodology, to take into account electrical masking, for calculating the SET width at the output of a path. The extension helps to calculate the worst-case width of the SET reaching a PO. Once the SET width at a gate output is known, other masking effects can be calculated as follows. (1) Commercial noise simulators such as Pacific [18] can be used to characterize electrical masking and get the nominal SET width at a PO. (2) The logical masking probability can be calculated through gate-level simulations of the circuit as described in [19]. (3) The temporal masking probability can be calculated using analytical expressions presented in Chapter 3. As all these tasks are well understood, here we just focus on calculation of SET width at a gate output.

Both lookup table (LUT) based approaches and closed form expressions for the shape of the SET pulse have been presented recently. A closed form expression for the output voltage of a gate $V_{out}(t)$ due to charge collection was presented in [20].

$$V_{out}(t) = \frac{Q}{T \times C_L} \times e^{(-t/\tau)}\left(\frac{e^{(t/\tau)} \cdot e^{(-t/T)}\Big|_0^t}{1/\tau - 1/T}\right), \tag{2.1}$$

Where T is the time constant of charge collection, Q is charge collected, C_L is the load capacitance, $\tau = f(Q, C_L,$ gate size) is a time constant of the gate, and f is a function obtained by doing linear regression on a table of τ values indexed using Q, C_L, and gate size. This leads to a complex expression for calculating $V_{out}(t)$, and the error for points not close to the table index have been reported to be more than 15% [20]. A uniform LUT was used for determining the width of the SET at a gate output in [21]. The LUT was constructed for different gate types, fan-ins, sizes, channel lengths,

supply voltages, threshold voltages, load capacitances, and for a particular charge collected at the gate output. Our experiments on the sensitivity of SET width show that it varies non-linearly with increasing gate sizes and load capacitances. As the SET width varies non-linearly for some parameters, the distance between the LUT indices needs to be non-uniform (and hence the interpolation for points not close to LUT index needs to be non-linear), which has not been considered in [21]. Moreover, constructing a single LUT for many different parameters results in a big LUT size and leads to a significant loss of accuracy during LUT interpolation. In addition, constructing a LUT for power supply variation leads to a large number of points in the LUT, which is handled through regression in our methodology.

The sensitivity of SET width to gate drive strength, output capacitance, and the charge collected around a gate's output is first studied through HSPICE simulations. Based on the sensitivity of SET width to different parameters, we determine the LUT indices and the interval between them. Additionally, the lower and upper bound for each LUT parameter and, the number and interval between each LUT index are determined based on accuracy requirements. The accuracy of the LUT and the time required for interpolation within the LUT are measured, and compared with HSPICE simulation.

2.1 Simulation Setup

The simulation setup for studying the sensitivity is shown in Figure 2.1. C_L is the total load capacitance at a gate output, and is equal to the sum of fanout gate capacitance and lumped wire capacitance driven by the gate. The current source

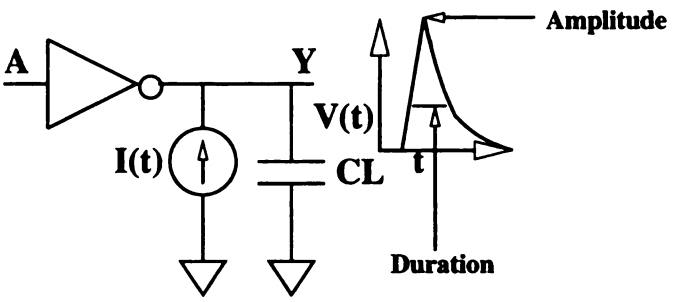

Figure 2.1. Circuit setup used to measure the sensitivity of SET width to various parameters.

connected to the gate output models the current flowing across the P-N junction due to charge collection. The direction of the current source for a 0→1 output flip (charge collection around a PMOS drain) is as shown in Figure 2.1, while it is reversed for a 1→0 flip (charge collection around a NMOS drain). The current source is modeled by a single time constant and is given by equation 2.2 [22]:

$$I(t) = \frac{2Q}{T\sqrt{\pi}}\sqrt{\frac{t}{T}}e^{(\frac{-t}{T})}, \tag{2.2}$$

where Q is the charge collected at a gate output and T is the charge collection time constant. The time constant T depends on the technology and if the drain is P or N-type. It is a measure of how fast the electrons recombine in the drain node. The current has sharp rise time, which models the drift mechanism through which the minority carriers are swept across the P-N junction, and it reaches its maximum value at $T/2$. The fall time is more gradual, due to the diffusion of carriers across the P-N junction and is determined by the exponential term with time constant T.

The value of T for different CMOS technologies was determined through device

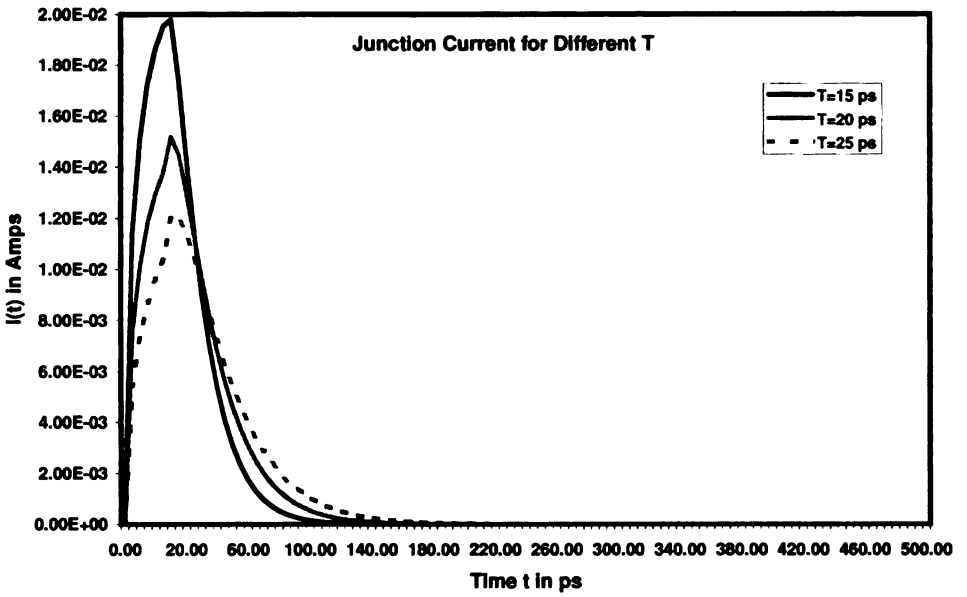

Figure 2.2. Junction current waveform for different time constants.

simulations and tabulated in [3]. We determine the value of the time constant for TSMC 180 nm, which is the technology used in our experiments, by scaling the time constant values given in [3]. The values used for the P and N-type drain time constants are 45.2 and 46.4 ps, respectively. The shape of the current waveform for different values of T is shown in Figure 2.2. As can be seen from Figure 2.2, for smaller T peak value of I(t) is higher, and current decreases rapidly leading to a smaller current pulse width. Higher peak value of I(t) results in lower Q_{crit}, and the smaller current pulse width leads to faster gate output recovery and hence smaller SET width. If the exact value of T for a specific technology is not known, then the LUT can be characterized for different T values. Later, the LUT can be indexed with the time constant for which Q_{crit} is being calculated.

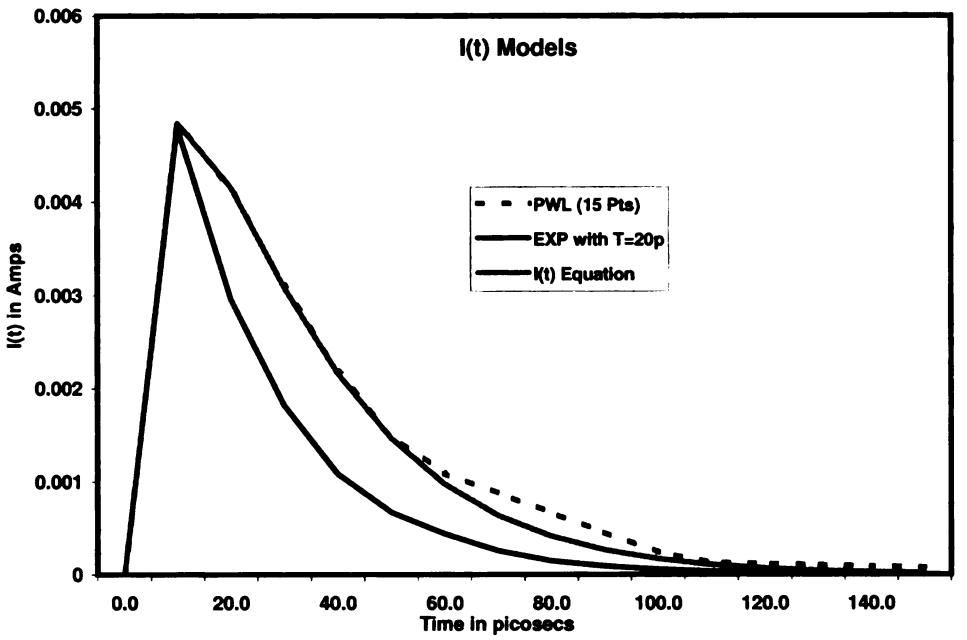

Figure 2.3. Current waveform constructed using values from Eq. 2.2, PWL model with 15 points, and exponential HSPICE model for Q=200 fC.

2.1.1 HSPICE Modeling of I(t)

The current waveform in Eq. 2.2 can be modeled as a piece-wise linear (PWL) or exponential current source in HSPICE. I(t) values for a PWL model with 15 points, and an exponential model from HSPICE were compared to the values computed from Eq. 2.2. The experiments were repeated for two different charges Q=100 and Q=200 fC. The current waveforms for Q=200 fC using the PWL, exponential model, and values from Eq. 2.2 are shown in Figure 2.3.

The percentage error for both the PWL and the exponential model was calculated as follows:

$$Error(\%) = \frac{|I(t) - I(t)_{PWL\ or\ EXP.}|}{I(t)} \times 100. \tag{2.3}$$

The maximum error for PWL and exponential models between t=0 and 50 ps is given in Table 2.1. Error is only measured between t=0 and 50 ps, as the value of current, after t=50 ps, falls to less than 70% of peak value of I(t). The maximum error for the PWL model with 15 points was found to be just 1.7% as compared to 117% for the exponential model. Hence, the PWL model is used for modeling the current across P-N junction, due to charge collection, in HSPICE.

	Q=200 fC	**Q=295 fC**
PWL (%)	1.34	1.7
Exponential (%)	100	117

Table 2.1. Percentage error for both PWL and exponential current sources.

2.2 Sensitivity of SET Width

In this section, the sensitivity of SET width to different gate characteristics is studied. The indices of the LUT are determined based on the sensitivity studies.

2.2.1 Gate Inputs

The critical charge required to cause a 1→0 or a 0→1 flip depends on the input vector applied to the gate. Figure 2.4 shows the normalized Q_{crit} for different input vectors in a three input NAND and NOR gate, respectively. As can be seen from the Figure 2.4, the Q_{crit} considering only the 1 → 0 flip at the output of NAND3 gate varies by six times. This is due to two reasons: (1) The strength of the pull-up or pull-down network that is ON, which determines how fast the deposited charge is dissipated by the gate. (2) If the top transistor in a stack is conducting. When the top transistor is ON, the effective output capacitance is equal to sum of actual output

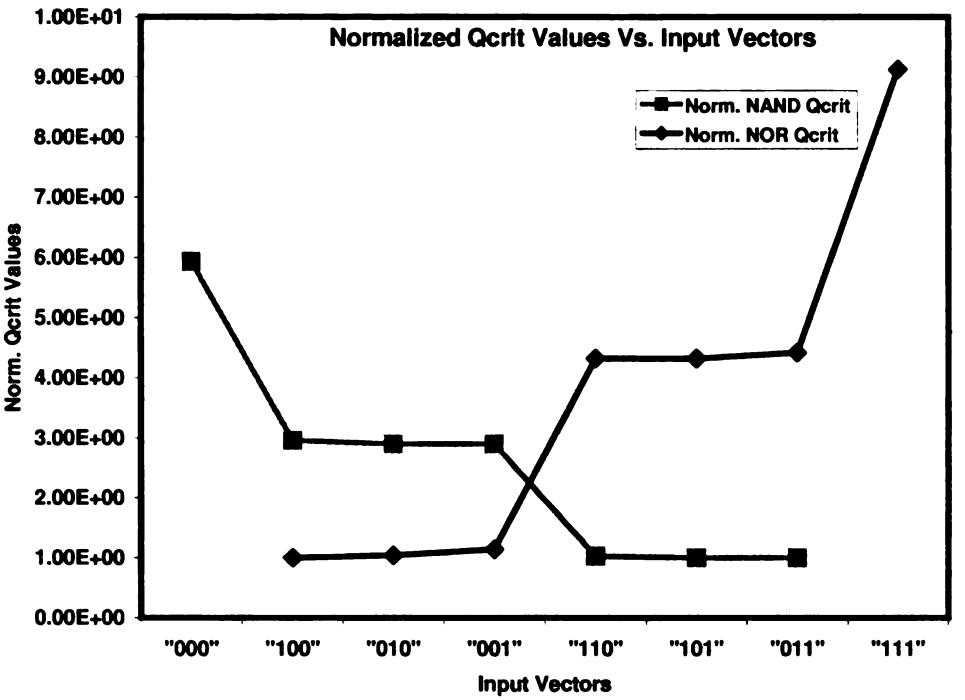

Figure 2.4. Q_{crit} of a NAND3 gate for a 1→0 flip, and a NOR3 gate for a 0→1 flip, normalized with respect to the minimum Q_{crit} among the input vectors considered.

node capacitance and the capacitances at the internal nodes. The larger capacitance increases the charge required to generate an SET, which means the SER of the input vectors - "101", "110", "100" - are much lower compared to other inputs.

2.2.2 Output Load Capacitance

The initial charge stored at a gate output node is equal to the load capacitance (C_L) times the supply voltage (V_{DD}). The effect of the load capacitance on SET width is shown in Figure 2.5. Increasing C_L, increases SET width for a particular charge Q. However, C_L also determines if Q results in an SET with amplitude greater than $V_{DD}/2$. For example, this can be seen from the plot for Q=105 fC, where SET width falls to zero when C_L increases from 30→40 fF. This shows that the effect of C_L on

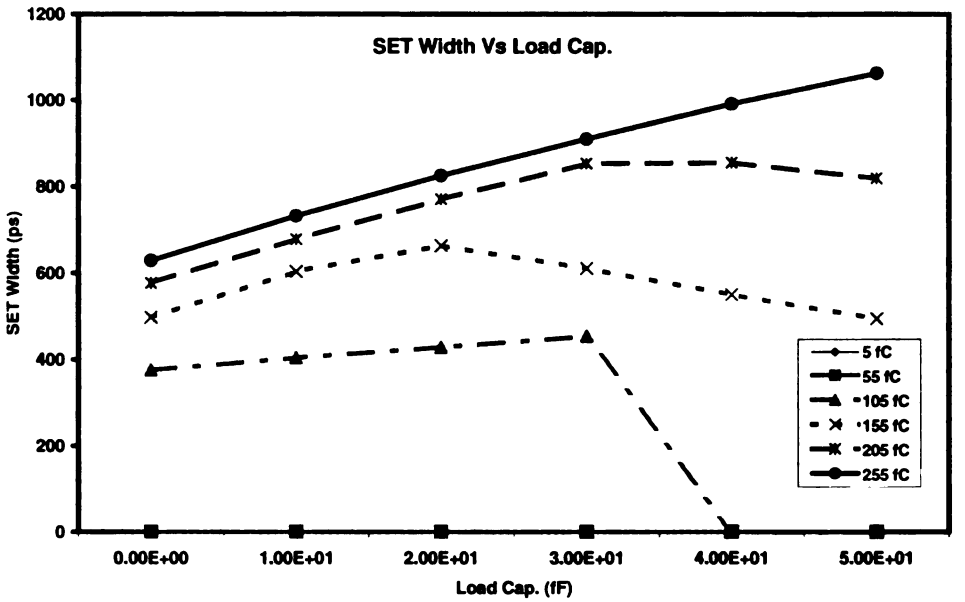

Figure 2.5. The width of SET measured at an inverter output for different charges, when the output load capacitance is varied from 0 to 50 fC.

SET width to be highly non-linear.

2.2.3 Charge Collected

The generation of SET depends on the charge collected around the P-N junction of a drain node. The minimum or threshold charge (Q_{th}), defined as charge required to produce an SET whose amplitude exceeds $V_{\text{DD}}/2$, depends on the gate drive strength and load capacitance. Once the charge collected exceeds Q_{th}, the SET width increases non-linearly with increasing Q. This can be seen from Figure 2.6, which shows the SET width as a function of increasing Q. For example, in the case of INVX1, SET width increases from zero to 600 ps when Q increases from 0→295 fC.

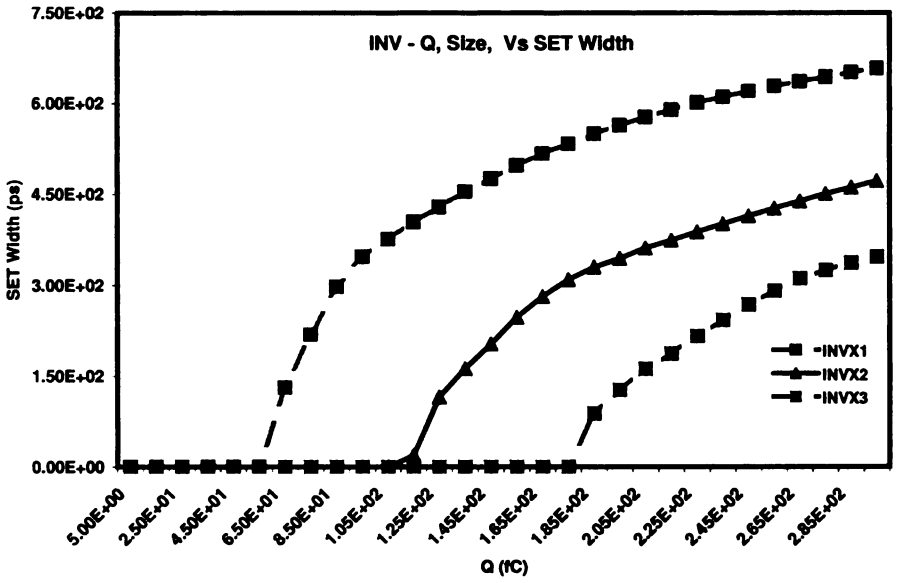

Figure 2.6. The width of SET measured for different drive strengths of an inverter when charge Q varies form 0→300 fC.

2.2.4 Gate Size

The gate size or drive strength determines the Q_{th} required to generate an SET of width W. The drive strength determines how quickly the charge collected around the P-N junction is dissipated. Higher drive strength gates have higher output capacitances, which also increase Q_{th}. The effect of different drive strengths on the SET width can be seen from Figure 2.6, which plots the SET width for increasing Q at the output of an inverter. For example, Q_{th} increases from 60 to 180 fC when the drive strength of an inverter increases from 1X→3X. In addition, the maximum SET width reduces from approximately 600→300 ps when the inverter strength increases from 1X→3X. The drive strength has a bigger and non-linear impact on the scaling of Q_{th} and SET width than C_L and Q, due to which it is not used as an LUT index.

2.3 Lookup Table

A two-dimensional LUT was constructed with output load capacitance and the critical charge as indices, for different gates. We first determine the lower and upper bound for the parameters in the LUT, and then find the number of points to be used for less than 10% error in SET width looked up. The lower and upper bound of C_L is determined by the minimum and maximum load values driven by the gates in the design. In the case of synthesized designs, this value can be determined from the standard-cell library models for delays. For our experiments, we bound the load capacitance of a gate to $0 \rightarrow 4 \times C_g$, where C_g is the gate input capacitance. The maximum charge collected around a P-N junction is determined as follows. The magnitude of the charge collected around a P-N junction depends on the energy of the particle passing through the silicon substrate, as well as the path length over which the charge is collected. The energy of ionizing particles is measured by the metric linear energy transfer (LET), which is the energy per unit mass per unit area transferred from an ionizing particle to the material through which it passes, expressed in MeV-cm^2/mg. It has been estimated that 1 MeV-cm^2/mg of neutrons deposit 10.8 fC/μm of charge [23]. The flux of particles decreases exponentially with increasing LET, and there are far fewer particles with LET $>$ 15 MeV-cm^2/mg [24]. Therefore, the maximum LET of ionizing particles considered is 15 MeV-cm^2/mg. The length over which the charge is collected depends on the technology and is approximated to be 2 microns for TSMC 180 nm technology [23]. Therefore, the maximum charge collected is approximated to be 300 fC (= 15 MeV-cm^2/mg * 10 fC/μm * 2 μm), and

the minimum charge is 0 fC.

Once the lower and upper bound for the parameters were determined, the initial LUT was created by choosing six and thirty points for Q and C_L, respectively. The actual points were uniformly spaced apart within the lower and upper bound mentioned above. For each gate, the total number of points in the LUT were 180 and hence an equal number of HSPICE simulations were done. In each HSPICE run, the SET width for a particular Q and C_L was recorded. Surface described by three co-ordinates X, Y, and Z corresponding to Q, C_L, and SET width, respectively, is used to do interpolation when the LUT is indexed with Q and C_L not in the LUT. The SET width is obtained by solving equation 2.4.

$$Z = \alpha + \beta X + \gamma Y + \delta XY \tag{2.4}$$

The value of the co-efficients were obtained by solving four such equations using gaussian elimination. The four equations correspond to four neighboring points shown in Figure 2.7.

2.4 Accuracy of the LUT Model

The accuracy of the LUT model was tested using C_L and Q located in the middle of the LUT indexes as shown in Figure 2.7. The SET width from the LUT table was compared to that of HSPICE simulations. The percentage error in the LUT interpolation was calculated as follows:

$$Error(\%) = (\frac{|(LUT\ value - Spice\ result)|}{LUT\ value}) * 100 \tag{2.5}$$

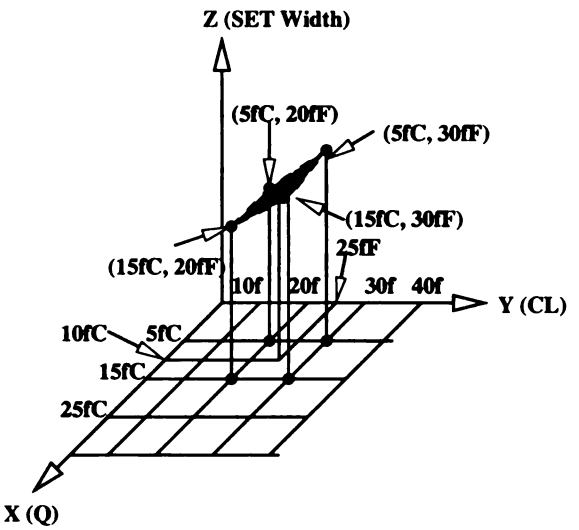

Figure 2.7. Surface described by three co-ordinates X, Y, and Z corresponding to Q, C_L, and SET width. Q and C_L values in the middle of LUT indices are used to test the accuracy of the LUT. The figure shows the neighboring points and the surface formed by them, when the LUT is indexed using Q = 10 fC and C_L = 25 fF.

The percentage error for Q values closer to Q_{th}, where the maximum error occurs, are shown in Table 2.2.

C_L (fF)	Q (fC)	Max. Error (%)	
		Uniform	Non-Uniform
5	80	2.31	1.05
15	90	7.05	4.3
25	100	17.02	9.18
35	110	10.14	6.45
45	120	15.37	8.6

Table 2.2. Percentage error for interpolation from a LUT with both uniform and non-uniform interval between indices.

The width of an SET changed significantly for values of Q near Q_{th}, which led to the

LUT interpolation producing errors greater than desired 10%. The accuracy can be improved by increasing the number of points in the LUT. For example, increasing the number of points for Q to forty would improve accuracy, but with a huge increase in LUT size. To reduce the error and to keep the complexity of the LUT within bounds, it was decided to use non-uniform spacing between charges, instead of increasing the number of points and maintaining an uniform interval between charges. The spacing between charges from 60→120 fC was reduced to 5 fC, while maintaining 10 fC spacing for other charges. The percentage error for the new LUT is also given in Table 2.2, in the column titled non-uniform spacing. The time taken to interpolate and lookup the SET width was found to be 12 ms for 1000 points, while it takes 3 hours for the same in HSPICE simulations. The model offers a speed-up of > 1000x compared to HSPICE simulations.

2.5 Regression for Supply Voltage Variation

Power supply variation also causes the SET width to change. The SET width for a 10% variation in V_{DD} is shown in Figure 2.8. As can be seen, the SET width varies linearly over ±10% range. We tried linear regression over this small range and found that it gave a good fit. The R^2 value was found to be 0.99 for both Q=125 fC and 175 fC. The straight line equation shown in Figure 2.8 can be used to scale the characterized SET width while operating voltage varies by 10%. This avoids costly HSPICE simulations required for re-characterizing the LUT for variations in supply voltages.

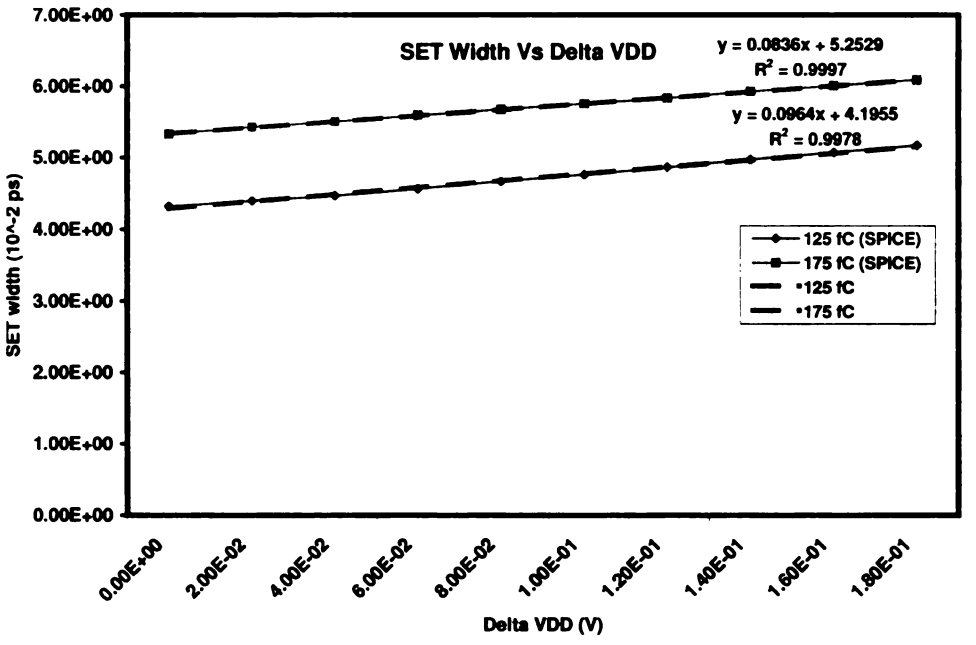

Figure 2.8. Variation of SET width for supply voltage perturbation of $\pm 10\%$ from the nominal value of 1.8 V.

2.6 Conclusion

In this chapter, we first analysed the sensitivity of SET width to gate drive strength, output load and charge collected at the gate output. A LUT, which is indexed using output load and charge Q, was proposed to calculate the SET width. The variation in the SET width for increasing Q is non-linear. Hence, spacing between Q values was varied to improve the accuracy of LUT. The error from the LUT interpolation was found to be less than 10%. A regression based model for scaling SET width due to V_{DD} variation was also presented. Further in Chapter 3, this LUT is extended to interpolate for SET width occurring in different points of a path.

CHAPTER 3

Error Masking for SER Mitigation

3.1 Introduction

In this chapter, we present an efficient error-masking design technique for static CMOS combinational circuits that exploits the inherent temporal redundancy (timing slack) of logic signals to increase their soft error reliability [25]. It has a number of features that make it attractive compared to existing approaches: (1) It modifies only the flip-flops of a combinational logic block (CLB) for sampling PO values and thus has lower area and power overheads. (2) Further helping lower these overheads is the use of a common delay line for an entire CLB or even multiple CLBs for producing control signals used in the technique. (3) In CLBs that have sufficient slack at a significant fraction of the PO gates, which is quite common, SER can be reduced markedly without any performance overhead. (4) The proposed design technique also masks soft errors in both the CLB and the master stage of the flip-flop.

The remainder of the chapter is organized as follows. Techniques that have been

proposed to handle logic soft errors are discussed in Section 3.2. Section 3.3 first characterizes the slack available in a path for error masking, then explains our error masking technique along with the circuits used to achieve this. Section 3.4 explains the logical construction of the delay chain used in the error masking technique. Section 3.5 describes the simulation setup and presents results obtained with ISCAS85 circuits.

3.2 Related Work

In this section, we discuss some of the earlier techniques proposed for logic soft error correction using self checking designs, architectural and circuit level techniques. Their drawbacks are explained and we motivate the need for new techniques.

3.2.1 Self-Checking Designs

Online or concurrent error detection (CED) can be achieved by using self checking circuits [26, 27], or by exploiting temporal redundancy of signals [28]. CED schemes use an output characteristic predictor, whose output is then compared (using a checker) with actual circuit output to detect an error. The output characteristic predictor is implemented in hardware using extra circuits, and recomputation is done in case of an error to recover the correct value. Self checking circuits are more efficient for arithmetic units, and may require high hardware cost for arbitrary logic functions. Also, online error detection and retry may affect performance (throughput) and cannot be used in real-time systems to overcome transient faults due to electrical noise or external radiation.

3.2.2 Architectural Techniques

At the architectural level, executing the same instructions in parallel using two cores or datapath has been used to detect soft errors in the core logic. This requires twice the logic as single-core processors and is extremely power and area hungry. Recently, microprocessor vendors have introduced dual-core designs to reduce the power consumed by their high-frequency processors. However, utilizing these dual cores for error detection, by executing the same instructions in parallel on both the cores would reduce the processor throughput. Additionally, it will have a big performance impact, as compared to single core designs, due to reduced clock frequencies of the dual-core designs. Therefore, dual core solutions targeted for SER reduction are prohibitively expensive for commodity applications. Hence, lower overhead variants such as redundant execution using spare elements (REESE) have been proposed [29]. The REESE approach involves placing each instruction that completes execution into a queue along with the results of the instruction. An additional stage in the pipeline then schedules execution of these duplicate instructions by mixing them in with regular instructions. The results from the duplicate instructions are compared with the original results, and any differences indicate that a fault has occurred. Queuing of the instructions and repeated execution increases the complexity of the system and requires significant amount of extra logic area. Moreover, the extra pipeline stage would increase instruction latency, and any soft errors in the instruction fetch or decode stages would not be detected. Time redundancy based architectural approaches also have significant performance, power overheads and design time cost [10].

3.2.3 Gate and Circuit-Level Techniques

Traditional techniques to provide soft error tolerance rely on triple modular redundancy (TMR), in which the original circuit is triplicated and a majority voter is used to determine the final output. However, this technique involves high overhead (> 200%) in terms of area and cost, which limits its usage to reliability-critical applications. Various ideas for soft error tolerance based on time redundancy were presented in [30]. The time domain majority voter presented in [30] has a performance overhead, since the sampling is started after the longest path in the circuit settles. Another technique called partial error masking, corrects errors with lower overhead than traditional TMR techniques by utilizing the difference in soft error vulnerabilities of gates [31]. But it masks soft errors only in CLBs and has higher overhead compared to the technique presented in section 3.3

Upsizing of gates to reduce the SET amplitude to less than $V_{DD}/2$ was proposed in [32]. Gates which have the lowest logical masking probability are selected to achieve cost-effective trade-offs between overhead and soft error failure rate reduction. The results presented show a performance overhead of 12.2% for 90% SER reduction in 180 nm technology. A combination of skewed logic and output latches which respond only to a 0→1 or 1→0 transition was used to tackle logic soft errors in [33]. The gates in the CLB were skewed by upsizing the NMOS transistors in the gate, such that the generation of 0-1-0 SET was reduced. The 1-0-1 SETs were handled by using a dual-sampling flip-flop, that changed its output only for a 0-1 transition at the input. The results presented show a performance overhead of 420 ps and power

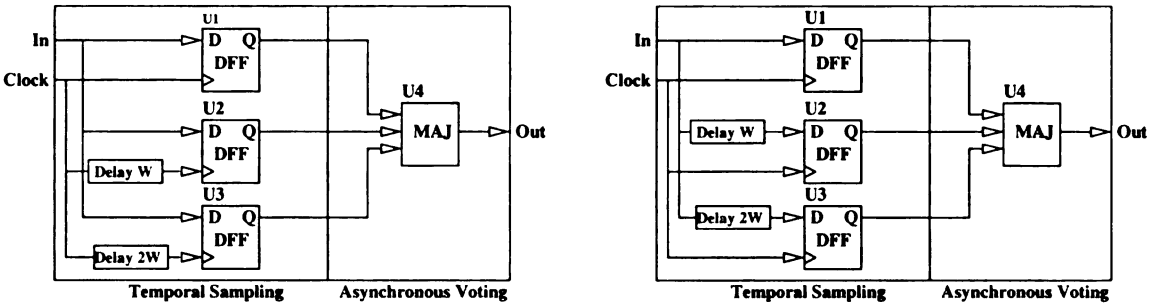

Figure 3.1. (a) Temporal sampling latch with internal clock generation delays. (b) Equivalent temporal sampling latch using internal data delays [23].

overhead of 160 μW even for a simple inverter chain.

Temporal Sampling Latch Designs: Prior efforts have also focused on latch design for mitigating soft errors in CLBs [34, 2, 23]. The latch design in [34] requires resistor insertion to slow down the input stage, which incurs both performance and area penalty. The techniques presented in [2, 23] use temporal redundancy to sample the circuit output at multiple delayed time instances, which is used to mask a pulse of width w. Some practical implementations of this technique described in [23] are shown in Figure 3.1.

Figure 3.1 shows a temporal sampling latch with internal clock generation and data delays. The three flip-flops U1, U2 and U3 sample the PO of a circuit at T, T+w and T+2w, respectively. Majority voting done by U4 on the three samples masks any SET of width w from changing the Out signal, which reduces the SER of the CLB. In Figure 3.1(a), the clock signal is delayed internally using delay chains to generate multiple data sampling edges. Equivalently, data is delayed internally to arrive at multiple times with respect to a single clock edge in Figure 3.1(b). The main drawbacks of the proposed latch designs are:

- **Performance penalty:** In order to tolerate an SET of width w the flip-flop setup time is increased by $2w$, which increases the clock period time and hence reduces clock frequency. For example, tolerating an SET width of 200 ps using the above temporal sampling latch, introduces a performance penalty of 8% for a 200 MHz clock frequency, which jumps to 40% at 1GHz.

- **Area overhead:** Usage of separate delay lines inside each flip-flop increases the area and power overhead incurred for delayed clock or data signal generation.

The techniques we discuss in this chapter, for error masking, have zero performance overhead. In addition, they use a delay line that is common to one or more combinational logic blocks (CLBs), as opposed to a delay line within each latch as done in [2].

3.3 Time Redundancy Based Error Masking

We first analyze the soft error vulnerability of a CLB in the original circuit, and then in the next paragraph, explain our technique conceptually and analyze how it exploits timing slack to reduce SER. All time instants in the following discussion are specified in terms of elapsed time after a cycle begins. Let T denote the cycle time. When an SET pulse is generated at the output of a static CMOS gate in a combinational circuit due to a high-energy particle strike, it may propagate through a path u and be captured by an output flip-flop (FF), causing a soft error. At $t_3 = T - t_{\text{setup}}$, u's output (primary output) is sampled by an output FF, where t_{setup} is the setup time of the FF. Consider an SET pulse of width w that can begin at any time during a cycle with equal probability. The probability $P(w)$ that this pulse, will latch at an

output FF and cause a soft error (i.e., it will overlap the sampling instant t_3) can be determined to be $P(w) = \frac{w}{T}$.[1]

Since the effect of an SET is only temporary, it is possible to prevent a soft error by exploiting timing slack available in the path u as follows. Let t_1 denote the worst-case propagation delay from the primary inputs to the output of u. The slack for u is then $t_s = t_3 - t_1$, i.e., in the absence of an SET, u's output will be stable at its correct value in the time interval $[t_1, t_3]$. If in addition to t_3, we sample u's output (in the connected flip-flop) at t_1 and t_2 too, where $t_1 < t_2 < t_3$, and we then perform majority voting among the three sampled values, we will be able to obtain the correct value of u's output whenever an SET pulse does not overlap more than one sampling instant. Let $t_{s12} = t_2 - t_1$ and $t_{s23} = t_3 - t_2$, and let $t_{s12} \leq t_{s23}$ without loss of generality. The probability $P(w)$ that an SET pulse of width w, after reaching u's output, will cause a soft error (i.e., it will overlap at least two sampling instants) can be verified to be as follows: (1) $P(w) = 0$ when $w < t_{s12}$; (2) $P(w) = \frac{w - t_{s12}}{T}$ when $t_{s12} \leq w < t_{s23}$; (3) $P(w) = \frac{2w - t_s}{T}$ when $t_{s23} \leq w < t_s$; and (4) $P(w) = \frac{\min(w,T)}{T}$ when $w \geq t_s$. The transient pulse and its overlap with different sampling points to cause a soft error is shown in Figure 3.2. Thus, in the first three cases our technique improves soft error tolerance and has the same tolerance as the original circuit in the last case. In the first case, soft errors are always prevented. To maximize the pulse width that is guaranteed to be tolerated, we choose $t_2 = \frac{t_1 + t_3}{2}$ or $t_{s12} = t_{s23}$, so that SET pulses of width less than half of slack at u are guaranteed to be tolerated.

[1]More precisely, a soft error will be caused if the SET pulse overlaps the setup and hold time interval of the output FF.

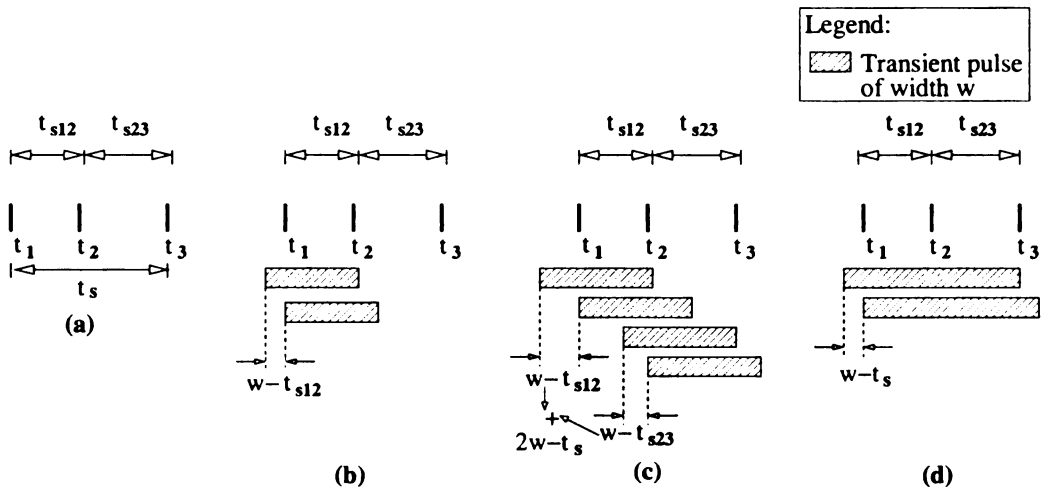

Figure 3.2. Figures (b), (c) and (d) show different transient pulse widths and their starting and ending times when they overlap two sampling points to cause soft error. (a) Effective slack available in a path and the time when the FF samples: t_1, t_2 and t_3. Probability of the SET latching for three different widths: (b) Transient pulse width is greater than t_{s12} and covers both t_1 and t_2. (c) Transient pulse width is greater than t_{s12} and t_{s23}, hence can overlap both t_1 and t_2 or t_2 and t_3. (d) Transient pulse width is greater than t_s and completely covers the slack time t_s.

We now move onto implementation issues. First, we discuss circuits for sampling a path's output values and to do majority voting. Then we describe the logical construction of a delay chain, which is used to generate the sampling control signals for the FF. In the above discussion, we exploited the complete slack from t_1 till t_3 to reduce SER. However, due to implementation reasons (as explained below), the actual slack available in a path for error masking within a clock cycle is given by:

$$S_{max} = T - (t_{pd,worst} + t_{D-C1} + t_{D-CK} + t_{CK-Q}). \tag{3.1}$$

In Eq. 3.1, $t_{\text{pd,worst}}$ is the worst case propagation delay in a path, while $t_{\text{D-C1}}$ and $t_{\text{D-CK}}$ are the setup time requirement for the first and third sample (D1 & D3,

respectively) in the sampling latch, t_{CK-Q} is the clock to flip-flop output delay which includes the majority voter delay. The setup time t_{D-C1} is defined as the D-to-C1 offset that causes a a wrong value to be latched at D_1, while setup time t_{D-CK} is defined as the minimum D-to-clock offset that causes the clock-to-output (Q) delay to be 5% higher than its nominal value. The effective transient pulse width that can be tolerated is then $S_{max}/2$. The actual sampling is done at time instants t'_1, t'_2 and t_3 (the last sampling time remains unchanged), such that $t_1 \leq t'_1 < t'_2 < t_3$. We define $t'_s = t_3 - t'_1$, $t'_{s12} = t'_2 - t'_1$ and $t'_{s23} = t_3 - t'_2$, and let $t'_{s12} \leq t'_{s23}$ without loss of generality.

3.3.1 Output Sampling and Majority Voting

The sampling is performed by adding two sets of n and p control transistors (corresponding to t'_1 and t'_2) to a FF as shown in Figure 3.3(a). At sampling time, control signals C_1 and C_2 ($\overline{C}$) go high (low), which disconnects output node F from V_{DD} and VSS, thus preventing any further transitions and completing the sampling. A majority voter embedded into the slave stage of the FF determines the final output value (see Figure 3.3(a)). To reduce the susceptibility of node D1 and D2 to particle strikes after sampling (when it is essentially a dynamic node), cross-coupled inverters (shown in Figure 3.3(b)) are added to make it static. Explicit switched-capacitors can also be added to harden the cross-coupled inverters against soft errors [35]. The capacitor addition should be done based on SER requirements and power and area overheads incurred.

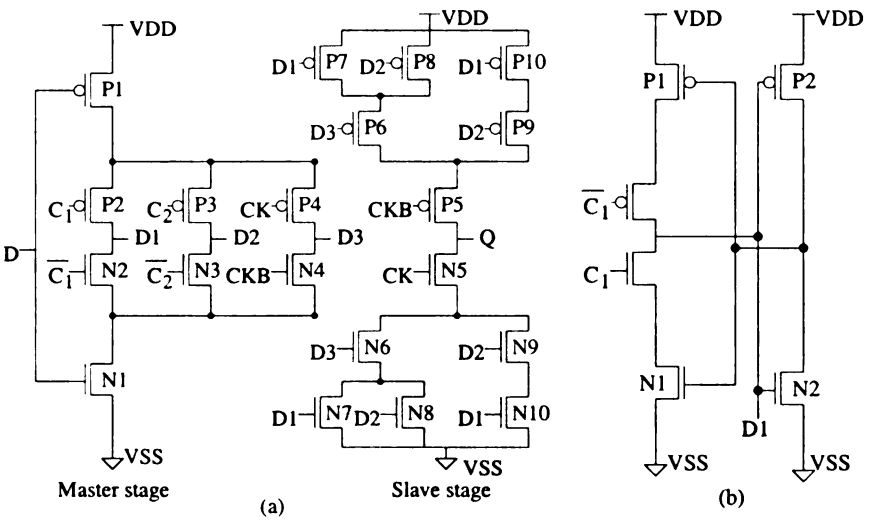

Figure 3.3. (a) A modified C^2MOS flip-flop to sample and latch signal values at different time instances within a clock cycle. The slave stage contains a majority voter to vote among the different sampled values. (b) Cross-coupled inverters added to D1 and D2, to keep it static after C_1 and C_2 become high.

An SET pulse generated in the CLB and reaching the modified FF will be tolerated as per our analysis in Section 3.3. An SET pulse generated only at D1, or D2, or D3 of the modified FF due to a particle strike (an SEU) can always be tolerated because of majority voting. However, a single-event multiple upset (SEMU), i.e., a single particle strike causing transient pulses to be generated at multiple data nodes, can be a problem as it can cause a wrong value to appear at the majority voter output. Since it is hard to characterize, through simulation, the charge required for a SEMU, we do not include soft error contribution of FFs to calculate original and final reduced SER (i.e., we present quantitative SER reduction results only for CLB). However, the data nodes D1, D2 and D3 in the modified FF can be spaced apart in the layout, by placing the cross coupled inverters and the layout of any explicit switched capacitances present between the data nodes. This would further reduce the

chances of a SEMU occurring in the FF itself.

There are two cases when SETs are not masked by the error masking technique. SETs generated at the output of the majority voter gate are not masked, while transient pulses in critical or short slack paths are not masked. Errors occurring at the output of a majority voter gate affect next stage in the pipeline, which is corrected by using our technique in the subsequent pipeline stage. In case of reconvergent paths, where a transient pulse propagates through both paths, a single logical flip originating before reconvergent paths begin can affect more than one sampling point. An error can occur if the delay difference between the reconverging paths makes the same transient pulse overlap two sampling points. To protect the sampling points t'_{s12} and t'_{s23} should be made greater than the delay difference between reconverging paths plus the overlapping error pulse width, or delay difference between reconverging paths can be reduced by increasing the delay of faster path. Techniques to prevent soft errors in critical paths are described in chapter 4.

3.4 Delay Chain

The control signals C and $\overline{C}$ are generated using the circuit shown in Figure 3.4. The generation of the controls signals are explained using the NMOS control signal $\overline{C}$. The circuit used for generation of $\overline{C}$ depends on when it falls low. $\overline{C}$ is generated by delaying CK when it transitions low after $\frac{T}{2}$, while $\overline{C}$ is generated by ANDing CK and delayed $\overline{CK}$ when it goes low before $\frac{T}{2}$. C is generated by inverting $\overline{C}$ in both cases. Particle strikes in the control signal generation circuit can also cause soft errors, as a result of glitchy transitions generated in control signals. The occurrence

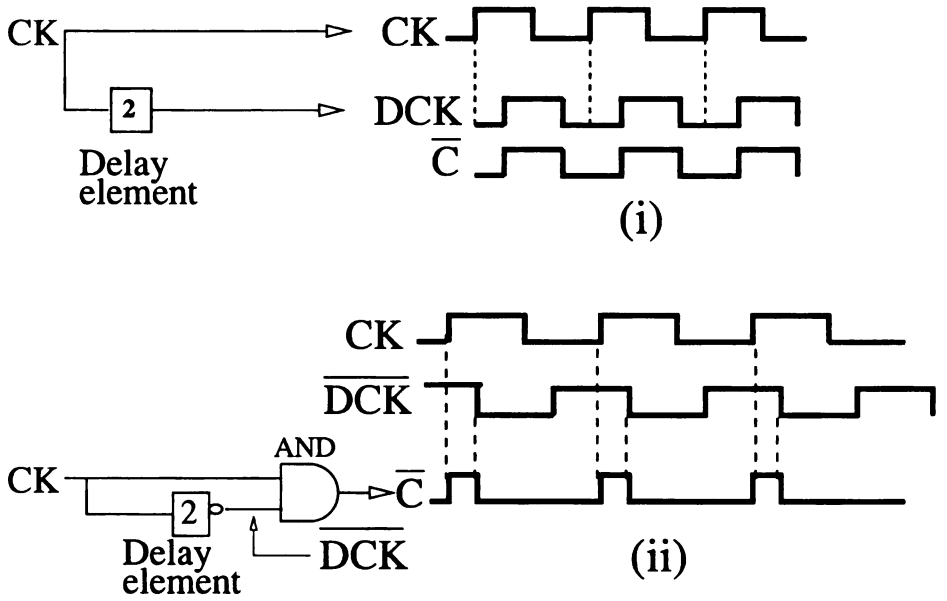

Figure 3.4. **Generation of control signals C and $\overline{C}$.** (i) $\overline{C_1}$ and $\overline{C_2}$ are generated by delaying CK if they go low after $\frac{T}{2}$. DCK shown is used as $\overline{C_1}$ or $\overline{C_2}$. (ii) $\overline{C_1}$ and $\overline{C_2}$ are generated by ANDing CK and delayed $\overline{CK}$ when they go low before $\frac{T}{2}$. C is generated by inverting $\overline{C}$ in both cases.

of such soft errors is determined by the sampling time t'_1, t'_2 and t_3 for a FF. Since sampling time t_3 always occurs at T-t_{setup}, we only consider the occurrence of t'_1 and t'_2 with respect to $\frac{T}{2}$ (CK is symmetric and for simplicity $t_3 = T$ is used here). We do not consider particle strikes on the CK signal itself due to high load on CK signal.

1. $t'_1 < \frac{T}{2}$ and $t'_2 < \frac{T}{2}$: $0 \rightarrow 1$ logic flip occurring in the delay chain before t'_1 and extending till t'_2 will make both $\overline{C_1}$ and $\overline{C_2}$ low before t'_1. $\overline{C_2}$ remains low till t'_2 which causes a wrong value to be latched in both D1 and D2. The corresponding waveforms are shown in Figure 3.5(a).

2. $t'_1 < \frac{T}{2}$ and $t'_2 > \frac{T}{2}$: In this case t'_{2d}, the time by which CK signal has to be shifted to produce control signal $\overline{C_2}$ is:

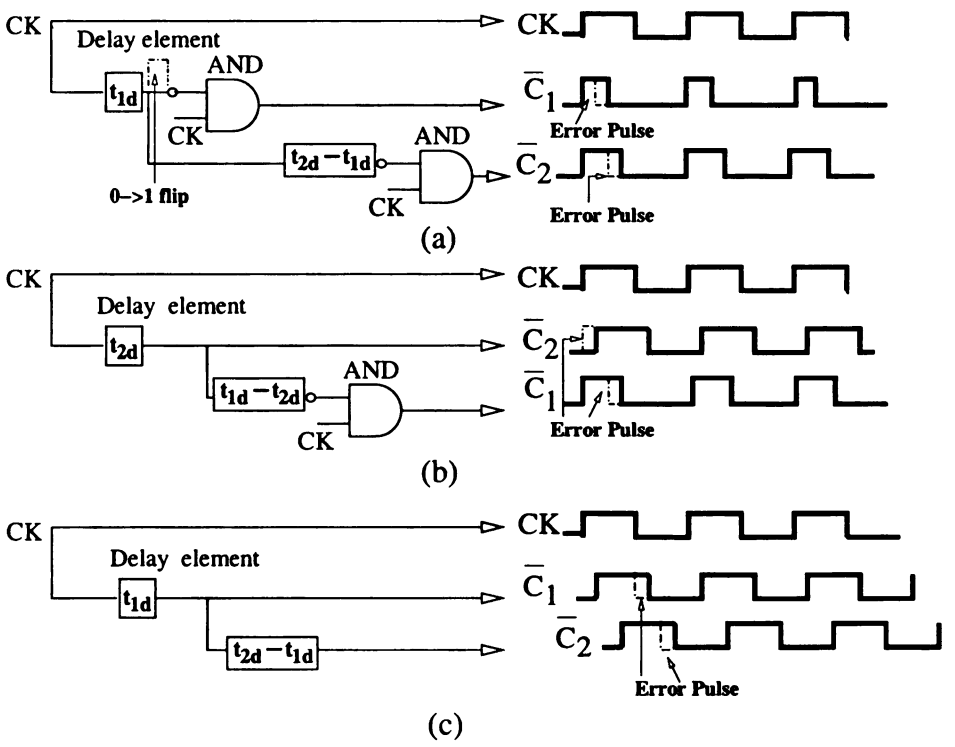

Figure 3.5. (a) $t'_1 < \frac{T}{2}$ and $t'_2 < \frac{T}{2}$. Zero to one logic flip affects both $\overline{C_1}$ and $\overline{C_2}$. (b) $t'_1 < \frac{T}{2}$ and $t'_2 > \frac{T}{2}$. Zero to one logic flip affects only $\overline{C_1}$. (c) $t'_1 > \frac{T}{2}$ and $t'_2 > \frac{T}{2}$: Both $\overline{C_1}$ and $\overline{C_2}$ are affected.

$$t'_2 = t'_1 + (\frac{T - t'_1}{2}) = \frac{t'_1}{2} + \frac{T}{2}$$
$$t'_{2d} = t'_2 - \frac{T}{2} = \frac{t'_1}{2},$$

which is smaller than $t'_{1d} = t'_1$. The corresponding waveforms for $\overline{C_1}$ and $\overline{C_2}$ are shown in Figure 3.5(b). A 0→1 logic flip occurring in $\overline{C_2}$ as shown by the dotted line would cause $\overline{C_1}$ to go low earlier than t'_1, which may cause a wrong value at D1 in the gate shown in Figure 3.3(b). However, as $\overline{C_2}$ and hence D2 are not affected, the majority value still remains correct. Hence, a 0 → 1 logic

flip occurring in $\overline{C_2}$ does not cause a soft error. A 1→0 logic flip occurring in signal $\overline{C_2}$ before t'_1, could cause an error in D2 if the error pulse width extends till t'_2. Since $\overline{C_1}$ only changes to one, D1 is not affected by this 1→0 error in $\overline{C_2}$, which gives a correct value at the majority voter output.

3. $t'_1 > \frac{T}{2}$ and $t'_2 > \frac{T}{2}$: The corresponding waveforms $\overline{C_1}$ and $\overline{C_2}$ are shown in Figure 3.5(c). A 1→0 logic flip occurring in $\overline{C_1}$ before t'_1 and extending till t'_2 can cut-off both NMOS transistors controlled by $\overline{C_1}$ and $\overline{C_2}$, which can cause wrong values to be latched in both D1 and D2.

To avoid soft errors due to particle strikes on delay chains, two different delay chains are used to generate $\overline{C_1}$ and $\overline{C_2}$ in circuits which have flip-flops with triggering times satisfying conditions one and three above. Due to discrete nature of delays produced by the delay elements sampling cannot happen exactly at the ideal t'_1 and t'_2 times, which are equal to worst case output settling time of the path and $\frac{t'_1+t_3}{2}$, respectively. This requires us to determine the nearest sampling time which can be used to reduce SER. The number of discrete control signals C and $\overline{C}$ to be generated can be reduced by *clustering* and using common control signals for flip-flops whose sampling time occur close together. This reduces the area overhead by using fewer delay elements to generate control signals and fewer wires to route. Due to clustering of control signals sampling may be done at new time instants t''_1, t''_2 and t_3 (the last sampling time remains unchanged), such that $t'_1 \leq t''_1 < t''_2 < t_3$. We define $t''_s = t_3 - t''_1$, $t''_{s12} = t''_2 - t''_1$ and $t''_{s23} = t_3 - t''_2$. The new sampling time intervals t''_{s12} and t''_{s23} may reduce the effective error pulse width that can be tolerated. Therefore, the sampling times t''_1

and t_2'' have to be selected such that the decrease in the SER reduction is minimized. The construction of the discrete delay chain and the methodology used for clustering of control signals are explained in Chapter 5.

3.5 Simulation Results

The results for SER reduction on applying our technique to ISCAS85 circuits are presented below.

3.5.1 Extension of LUT to Calculate SET Width at Primary Output

The LUT described in Chapter 2 is extended to calculate the best case SET width that reaches the primary output. The simulation setup shown in Figure 3.6 is used to create a three-dimensional LUT. The gate for which LUT is being constructed drives a fixed number of inverters, so that an SET generated at the gate output has maximum width when it reaches the path end. This allows us to estimate the worst-case SER reduction while using the error masking technique. The LUT is constructed by measuring the SET width at inverter outputs which are driven by the gate. Therefore, the LUT is now constructed using three different parameters viz., gate output load, charge collected, and the level of gate from the PO. A total of ten inverters in a path are used to construct the LUT. Changes in the width of an SET generated more than ten levels away from the PO were found to be negligible. Hence, the LUT was found to provide sufficient accuracy for estimating the width of an SET generated more than ten levels away from the PO. The LUT for each

gate now contains a total of 1800 points (6×30×10). Interpolation, as described in Chapter 2, is done for output load and charge Q not located in the LUT. The accuracy for interpolation was tested using directed points which are located in the middle of existing LUT indexes. The maximum interpolation error was found to be less than 10%, which is well within the acceptable limits. The percentage error is similar to that of a two-dimensional LUT described in the previous chapter. This is because, the level of the gate used to index the LUT is same as one of the pre-characterized points, while interpolation is only done along the output load and the charge collected.

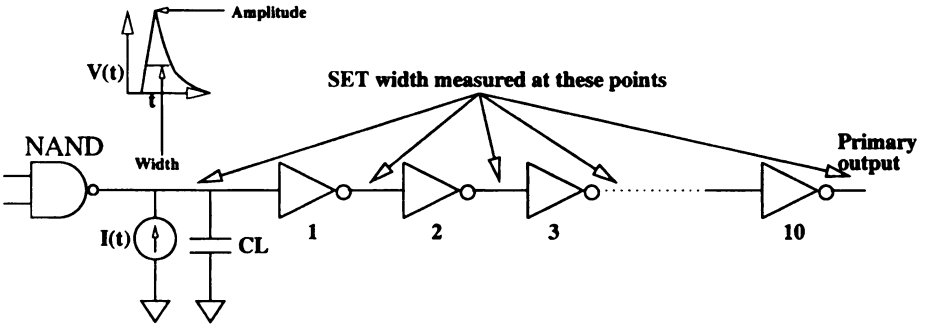

Figure 3.6. A total of ten inverters are connected between the NAND gate and the PO. The SET width is measured at the output of the NAND gate and the inverters.

3.5.2 Critical Charge and Transient Pulse Width Calculation

A LUT which contains the width of SET reaching PO gate was constructed for different gate types, their distance from the PO gate, load capacitance and charge collected. All the simulations were done using TSMC 180 nm transistor models with V_{DD} =1.8V. Thus, by varying the output capacitance and the charge collected around the P-N junction (up to a maximum of 300 fC), we construct lookup tables for the charge versus the transient pulse width. The LUTs were constructed for NAND, NOR and

inverters. The lookup tables are then searched using binary search technique, to get the Q corresponding to the transient pulse width required.

3.5.3 SER Calculation of Complete Circuit

ISCAS85 circuits were synthesized in 0.18 micron technology using the standard cell library described in [36]. Only inverters, two input NAND and NOR gates were used during synthesis. The SER of the original and the error masked circuit are given by the following equations.

$$\begin{aligned} TSER_{orig} &= \sum_{i=1}^{n} SEC(g_{i,w_{orig}}) \\ TSER_{red} &= \sum_{i=1}^{n} SEC(g_{i,w_{t'_s/2}}) \\ SEC(g_i) &= \sum_{\forall j} \Big(\sum_{k=1}^{m} (SER(Q_{L_k}) - SER(Q_{R_k})) \times P_{latch}(w_{Q_{L_k}}) \Big) \times P_j, \end{aligned} \tag{3.2}$$

where $SEC(g_{i,w_{orig}})$ and $SEC(g_{i,w_{t'_s/2}})$ are the soft error contributions of gate g_i when the transient pulse width required to cause an error are w_{orig} and $w_{t'_s/2}$. SER(Q), which is the basic soft-error rate of a gate, is calculated using Eq. 1.1. To calculate basic SER, neutron flux F=0.00565 neutrons·$cm^{-2}s^{-1}$ is used, sensitive device area A is equal to the sum of drain node areas connected to the output node, Q is the charge required to produce an SET pulse of required width and is estimated from the LUT, Q_s is the charge collection efficiency of the device in fC, K is a technology independent constant equal to 2.2×10^{-5}. The charge collection efficiency for 1→0

and 0→1 logic flips are 20.5 fC and 17.2 fC, respectively. For the sake of calculating SER reduction, we consider only 1→0 flips which have higher SER due to higher Q_s. SER(Q) gives the soft-error rate for charges equal to and greater than Q. The soft error contribution of each gate g_i is calculated starting from Q_{crit} up to a charge of 300 fC, which is the maximum charge that can be collected by a P-N junction in 180 nm. In order to calculate the SER of a gate for charges between Q_{crit} and 300 fC, we divide the charge values into m equal intervals of 5 fC. The soft error contribution of each interval is calculated by subtracting SER corresponding to right endpoint from the left [6]. The soft error contribution of each interval is weighted by the latching window probability of a transient pulse produced by charge Q_{L_k}, corresponding to the left endpoint in the interval. The SEC of each gate is calculated by summing the SER with respect to all flip-flops in its fanout cone and weighted by the probability P_j of the path to flip-flop j being functionally sensitized. As ISCAS85 circuits do not have specific input patterns to test them, the logical masking probability P_j is generated as a random number.

3.5.4 SER Reduction Using Error Masking

We first estimate the slack S_{max} available at each flip-flop for sampling the PO values. We constructed the original and modified C2MOS flip-flop (shown in Figure 3.3((a)) and simulated the circuits using TSMC 180 nm micron models. The setup and hold time for the flip-flop was measured by connecting them to a F04 load. The increase in t_{ck-Q} delay for the multi-sampling flip-flop when data transitions closer to ck is

shown in Figure 2.1. The setup time for the flip-flop is calculated from this plot. The value for t_{D-ck}, t_{D-C1} and t_{ck-Q} in the modified design were found to be 125, 115 and 75 ps, respectively. A delay chain capable of generating phase shifted clock signals every 200 ps was constructed and hence the sampling time t_1 and t_2 were determined from the control signal availability. The width of transient pulse that can be tolerated in the modified circuit t'_w is then calculated as $\min(t_3 - t_2, t_2 - t_1)$. The charge required to cause a transient pulse of width t'_w and the latching window width t_w=100 ps are then retrieved from the LUT. If $t'_w \leq t_w$, then no error masking is applied. The results obtained for ISCAS85 circuits due to error masking (EM) are given in Table 3.1.

Circuit	**Circuit Features**			N_{trig}	**SER Redn. %**
	Gates	PIs	POs	EM	
c432	210	36	7	3	55.66
c1908	1005	33	25	16	66.74
c2670	1498	233	128	44	99.58
c3540	2176	50	22	20	95.8
c7552	4785	207	108	58	81.14
c5315	3712	178	106	102	97.12
Avg.	2231	122.8	70.8	43.2	82.7

Table 3.1. SER reduction for ISCAS85 circuits due to error masking.

3.6 Conclusion

In this chapter, we presented an error masking technique for SER reduction in CLBs. The error masking technique samples and votes on the primary outputs within the slack available in a clock cycle time. This results in zero performance overhead.

Efficient flip-flop designs to do triple sampling and majority voting were presented. The error masking technique leads to average SER reduction of 82.7% in ISCAS85 circuits.

CHAPTER 4

Combining Error Masking and Error Detection Plus Recovery

4.1 Introduction

In this chapter, we describe techniques for combining error masking with error detection and recovery (EDR) to cope with soft errors in combinational and sequential circuits [37]. If the error masking technique is used alone, it prevents an SET pulse of width less than approximately half of the slack available in the propagation path from latching and causing a soft error. If the error masking technique is used in combination with EDR, SET of width approximately half the clock cycle time can be tolerated. The EDR technique has a single cycle penalty for recovering from an error latched into the pipeline. The EDR technique can be used in circuits with no slack to provide complete error protection for most applications. The SET is also masked without additional delay in an area- and energy-efficient manner, which makes this

technique attractive for commodity as well as reliability-critical applications. The area and power overhead can be traded-off with soft-error rate (SER) reduction based on application requirements. Techniques to improve slack and hence the SER reduction of the error masking technique such as: (1) exploiting circuit delay dependence on input vectors; and (2) redistributing slack in pipelined circuits, are also presented.

The remainder of the chapter is organized as follows. Section 4.2 presents existing error detection and correction techniques. Section 4.3 explains the EDR technique, and then characterizes the paths where error masking and EDR can be applied. Section 4.4 presents ways to increase the effectiveness of the error masking technique by utilizing the input value characteristics of a circuit, and by redistributing the slack available in a latch-based pipeline circuit. Section 4.5 presents results obtained with ISCAS85 circuits for all the techniques described in this chapter.

4.2 Related Work

In error detection and correction, detection is done by sampling the circuit output at two different time instances and then XORing the sampled values. Once an error has been detected, the correct output is recovered through recomputation. Efficient techniques to do error detection of soft errors due to particle strikes and delay faults were presented in [28, 38]. In both techniques an extra latch (called shadow latch in Razor [38]) is used to sample the circuit output. The first sample is stored in the main pipeline flip-flop at the rising edge of the clock (i.e. after a time T has elapsed from the beginning of the clock cycle), while the second sample is stored half a clock cycle later (at time 3T/2) in the shadow latch. This means any transient pulse with

width less than $\frac{T}{2}$ is detected as an error. Razor recovers from the error by restoring the value stored in the shadow latch into the main flip-flop, while the work presented in [28] suggests re-doing the computation to get the correct value. Implementing recomputation requires storing the current state and executing the program from an instruction not affected by the soft error. Also, recomputation requires many clock cycles, very high area and energy overhead and is difficult to implement in modern super-scalar processors due to complex circuitry required. The present version of Razor works very efficiently for delay faults, but there are certain limitations when it is used for handling soft errors due to particle strikes. In the case of Razor, if a particle strike had altered the value stored in the shadow latch, then restoring this value would result in wrong circuit output. As particle strikes are uniformly distributed in time, there is equal probability of a particle strike affecting either the main flip-flop or the shadow latch. Moreover, a particle strike in the combinational logic circuit can also change the value stored in the shadow latch. Hence, restoring the value from the shadow latch does not reduce the probability of soft error occurrence due to a particle strike on the latch or in the combinational logic circuit.

4.2.1 Error Masking

Error masking refers to error correction on-line. Efficient error masking techniques utilize both the spatial and temporal redundancy in a circuit. In the previous chapter, we presented an error masking technique (EM), which samples the circuit output three times within a single clock cycle and does a majority voting on the sampled

values. The error masking technique presented attempts to trade-off the SER reduction obtained with the performance and area overhead. The sampling and majority voting were done within the slack available in a circuit, which results in zero performance overhead. The area overhead was minimized by using a common delay chain to generate the phase shifted clock signals for sampling. The technique presented in Chapter 3 cannot be applied to circuits without slack, and for circuits with few non-critical paths the ratio of SER reduction to overhead would be very small (overhead is greater).

4.3 Techniques to Combine Error Masking and Error Detection Plus Recovery

In certain applications or circuits with balanced paths, using error masking in paths with slack alone cannot provide required soft error protection. The technique presented in Chapter 3 requires performance overhead in paths with insufficient slack, which is not acceptable for transient fault protection in timing-critical applications. One way to improve the soft error protection provided by error masking is to do only error detection in short-slack paths by sampling only twice within the slack available, or sampling after the clock closing edge plus contamination delay of the path. This can detect errors twice and much more wider than the nominal pulse width masked by error masking schemes. Thus, the error detection in critical paths can be combined with error masking in non-critical paths to provide SER reduction. However, as explained in Section 4.2, the cost of applying such error detection and retry techniques are very high. To overcome these drawbacks, we present a novel technique exploiting

both error detection and error masking on a single path to provide sufficient soft error protection for all circuits.

4.3.1 Error Detection and Recovery on a Single Path

We first explain the technique for doing EDR on a single path, then explain where EDR needs to be used, as opposed to error masking alone, analyze the overhead required and then present techniques to reduce the overhead. In the EDR technique to do error correction, we sample the path output or primary output (PO) three times and do a majority voting among the sampled values. As error detection is also done, sampling is extended till the end of the next clock cycle. Once an error is detected, the pipeline is stalled and the correct value from the error correction circuitry injected into the pipeline, in the next clock cycle. All time instants in the following discussion regarding the sampling time t_1, t_2, and t_3 are specified in terms of the elapsed time after a cycle begins. To better understand the discussion that follows, the reader is referred to Figure 4.1 which presents the latch used for sampling. The PO is sampled at time t_1, t_2, and t_3 to produce D_1, D_2, and D_3. Let T denote the cycle time. To tolerate the maximum transient pulse width, the time interval $t_3 - t_1$ must be maximum, with $t_2 - t_1 = t_3 - t_2$. The maximum slack (S_{max}) available for sampling in a path (where EDR is used) is given by:

$$S_{max} = 2 \cdot T - (t_{pd,worst} + t_{D-CK} + t_{D-C2} + t_{C2-fb}). \tag{4.1}$$

In Eq. 4.1, $t_{pd,worst}$ is the worst case propagation delay in the path, while t_{D-CK}

and t_{D-C2} are the setup time requirement for the first and third sample (D1 & D3, respectively) in the sampling latch, t_{C2-fb} is the delay from signal C2 going high to the output of multiplexor in the feedback path settling, which includes the majority voter delay. The setup time t_{D-CK} is defined as the D-to-CK offset that causes a wrong value to be latched at D_1, while setup time t_{D-C2} is defined as the minimum D-to-C2 offset that causes the D3 settling delay to be 5% higher than its nominal value.

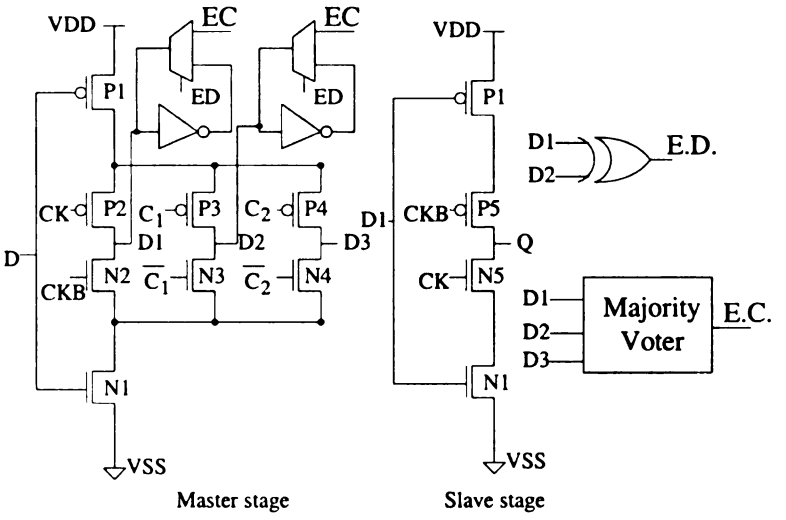

Figure 4.1. Flip-flop used for EDR in a path. XOR is used for error detection and a majority voter generates the correct output which is then fed-back in case of an error.

In the ideal case, the first sampling can be done immediately after the worst case output settling time, and $t_3 = 2 \cdot T - (t_{D-C2} + t_{C2-fb})$, while the second sampling is done at the middle of the time interval between t_3 and t_1. One of the sampled values needs to be passed onto the next pipeline stage, hence one of the sampling times is fixed at time T. D_1 is sampled at time T, as this enables the maximum pulse width to be tolerated. Based on the worst case propagation delay of the path, we offer the

following guidelines for choosing the soft error protection scheme in a path.

- The slack available in a path while using error masking alone is given by:

$$S_{EM} = T - (t_{pd,worst} + t_{D-C1} + t_{D-CK} + t_{CK-Q}). \tag{4.2}$$

  The effective transient pulse width that can be tolerated is then $S_{EM}/2$. If the transient pulse width tolerated is sufficient, then sampling and majority voting can be done within the slack available.

- If error masking done within the slack available does not provide sufficient soft error protection, then EDR should be used in a path. This requires the use of the modified flip-flop shown in Figure 4.1. The first and third sampling are done at $t'_1 = T - t_{D-CK}$ and $t'_3 = 2 \cdot T - (t_{C2-fb})$. The second sampling is done at $t'_2 = \frac{t'_1+t'_3}{2}$. The effective slack available in a path with EDR is:

$$S_{EDR} = T - (t_{D-CK} + t_{D-C2} + t_{c2-fb}).$$

  The error detection is done by XORing samples latched at t'_1 and t'_2.

4.3.2 Circuits for Error Detection and Recovery

The flip-flop used for sampling the PO values within the slack available in a circuit was described in the previous chapter. The master stage samples the PO values thrice, while majority voter is embedded into the slave stage of the flip-flop. The flip-flop for doing both error detection and recovery on a single path is given in Figure 4.1. As the first sampling of the PO is done at t'_1, D1 is latched by CK signal. The signals

C_1 and C_2 go high corresponding to sampling times t'_2 and t'_3, respectively. The slave stage passes the value latched at time T, i.e. D1, to the next pipeline stage. Due to an SET, D1 could have latched and passed on the wrong value to the next pipeline stage. If the width of the SET is bigger than $t'_2 - t'_1$, then the error detect signal (ED) which is the XOR of D1 and D2 changes to one, which leads to the majority voter output being fed back into D1 and D2.

Once an error has been detected, error recovery is done by clock gating and pipeline stalling. All the error detect (ED) signals from a pipeline stage are ORed to generate a single error detect signal for the stage. The error detect signals from different stages are ORed to generate a global pipeline stall (PS) signal. The PS signal is used to gate the clock signal that is being fed to pipeline latches. Clock gating prevents the CLB output generated in the next clock cycle from conflicting the output of the multiplexor (fb) fed into D1.

The generation of the clock gating signal needs to be done before the next clock cycle begins. This can be done due to the following reasons. (1) As the error detect signals are generated using D_1 and D_2, approximately half-a-clock cycle ($\frac{T}{2}$) is available for generating the pipeline stall and clock gating signals. (2) High-speed circuits, such as domino logic, can be used to generate the clock gating signal, as soft errors in these circuits do not lead to a functional failure (explained later). (3) Moreover, since EDR is applied to only the most critical paths in a circuit, the number of ED signals to be ORed are expected to be few (Section 4.5). In case generation and distribution of PS exceeds half clock cycle, we suggest use of counter-flow pipelining techniques as done in [38]. The correct value is fed back into D2 too, so that ED selects the new

circuit output latched in future clock cycles.

We assume that once a particle strike has occurred in the logic circuit, the chances of a new particle strike occurring in the latch and majority voter in the same clock cycle are negligible. Hence, a transient pulse without sufficient width to overlap two sampling points can always be masked. However, the chance of a particle strike occurring on the extra circuitry and causing an error is analyzed separately. There are three different cases which need to be considered for particle strikes.

- Output of error detector: If a particle strike changes the output of the XOR gate when there has been no SET generated in the CLB block, the circuit still functions correctly. This is because the output of the majority voter still remains same and the correct value is put into the pipeline in the next clock cycle. However, there is a one-cycle penalty due to wrong case of error detection. Since particle strikes on the error detection and pipeline stall circuits do not affect the circuit output, fast circuits constructed using domino logic can be used to generate pipeline stall and global clock gating signals.

- Output of majority voter: In the case of combined error detection and recovery in a single path, if a particle strike flips the output of a majority voter, the circuit still functions correctly. This is because, correct value has been passed on to the next pipeline stage, and since the ED signal in Figure 4.1 is not one, no feedback happens into the pipeline. In the case of error masking alone, particle strike at the output of majority voter can be corrected in the next stage, assuming all pipeline stages implement soft error protection schemes.

4.4 Techniques to Enhance Error Masking

In this section, we discuss two techniques to increase the slack available for doing error masking. The first technique exploits the input vector characteristics of the circuit. The second technique exploits time borrowing to increase the slack for most soft error vulnerable blocks in a pipelined circuit.

4.4.1 Exploiting Circuit Timing Dependence on Input Vector

The SER reduction obtained from the error masking technique can be improved further if the value of S_{EM} in Eq. 4.2 can be increased. This increases the width of the transient pulse and hence the particle charge required to cause an error. To increase S_{EM}, the sampling time t_1 can be shifted earlier than the worst case arrival time in a path. This means that the probability of a correct output being available at the sampled PO gate at time t_1 (P(t_1)) is less than one. The sampling times t_2 and t_3 should be positioned such that the probability of sampling a correct value in D_2 and D_3, P(t_2) and P(t_3), respectively are one. The SER of a gate with sensitized path to a flip-flop where $t_1 \leq t_{pd,worst}$ is:

$$\begin{aligned} SER_{new} &= P(t_1) \times SER(w \geq S_{new}/2) + \\ &\quad (1 - P(t_1)) \times SER(w \geq t_{lw}), \end{aligned} \tag{4.3}$$

where SER($w \geq S_{new}/2$) is the SER of a path, when the transient pulse width required to cause an error is greater than half of the new path slack, obtained by

shifting t_1 before t_{pd}, and SER($w \geq t_{lw}$) is the SER of the path when an SET of width greater than the latching window (t_{lw}) of the original latch can cause an error. The above equation represents the fact that when the first sample D_1 is wrong, SER of the path with error masking is same as the original circuit.

In order to reduce SER compared to the case when error masking was not used in the path:

$$SER_{new} < SER_{orig}, \tag{4.4}$$

where SER_{orig} is the soft-error rate of the path without error masking and is equal to SER($w \geq t_{lw}$). As SER is proportional to $e^{-\frac{Q}{Q_s}}$, where Q is the charge collected around a gate output, Q_s is the charge collection efficiency of the technology, Eq. 4.3 can be re-written as:

$$\begin{aligned} SER_{new} \quad \propto \quad & P(t_1) \times e^{-\frac{Q_{Snew/2}}{Q_s}} \\ & +(1 - P(t_1)) \times e^{-\frac{Q_{min}}{Q_s}} \end{aligned} \tag{4.5}$$

Here $Q_{Snew/2}$ and Q_{min} refers to the charge required to create transient pulses of width $S_{new}/2$ and t_{lw}, respectively. When $Q_{Snew/2} > Q_{min}$, then SER($w \geq S_{new}/2$) $<< SER(w \geq t_{lw})$. Therefore, approximating by ignoring the contribution of SER($w \geq S_{new}/2$) to SER_{new}, implies that SER_{new} linearly decreases with increasing $P(t_1)$. Thus, sampling earlier gives a much better SER reduction. This means if the application does not excite the worst case delay of the path for all inputs, then the

sampling point t_1 can be shifted earlier. In arithmetic units such as adders, multipliers and comparators narrow width input vector has been exploited for energy reduction. For example, during addition when the sixteen most significant bits (MSB) of a 32-bit vector are zero, the sum and carry output settle much earlier, which can be exploited to increase the SER reduction. A high level functional description of ISCAS85 circuits was used to determine the effect of input vectors on the outputs. We consider here input vectors which lead to least delay in the most critical paths of the circuit. For example, in C1908 when the inputs n953 and n952 are one and zero respectively, output check bits from the error settle with only a AND gate delay. This allows us to sample the output check bits much earlier than the worst case critical path delay. For results, the top five timing critical paths in all circuits were considered to settle at their minimum delay. The top five critical path outputs in circuits considered (all circuits in Table 4.1, except C432), were sampled much earlier and the resulting SER reduction was calculated. We found that the average SER reduction increased to 91%.

4.4.2 Slack Redistribution to Enhance Error Masking

In a symmetric latch based pipeline, time borrowing can be used to improve the SER reduction obtained. Time borrowing here, refers to utilizing the time voluntarily passed by a previous pipeline stage (usually referred to as slack passing) or taking up time from the next pipeline stage. The discussion of time borrowing is done with respect to the PO gate connected to latch, so as to remain consistent with

previous sections. In common pipelined circuits (such as those present in super-scalar processors), the sum of total logic delay across N pipeline stages and the latch t_{D-Q} overhead is usually less than N×T/2, where T is the clock cycle time. This is because it is impossible in practice to construct the pipeline such that data always arrives at a latch input when the latch is transparent [39]. Figure 4.2 shows three stages of such a pipeline with an ideal latch (t_{setup} and t_{D-Q} for the latches are zero) and assuming ideal clock signals. The output of CLB B settles 0.3×T time units before latch L_3 opens, which is not utilized by any logic. This creates a dead time which can be utilized for SER reduction. Figure 4.3 shows combinational logic block (CLB) A using the dead time to shift the sampling time in its critical path. In Figure 4.3, t_{early} is the contamination delay of CLB B. Effectively, the width of the error pulse that can be tolerated by logic in CLB A increases by 0.15 × T. The latch connected to PO gates in CLB A is similar to the master stage of the flip-flop shown in Figure 4.1, without multiplexor in the feedback path. The latch is clocked by C_1, C_2 and CK1, while the majority voter is not clocked. C_1 and C_2 close the second latch in Figure 4.3, corresponding to the sampling times t_1 and t_2, respectively. Separate signal for sampling at t_3 is not used, as the sampling is set by the timing constraints of the circuit.

In pipelines where cycle time cannot be further reduced for correct operation, i.e., the sum of total logic delay across N pipeline stages and the latch t_{D-Q} overhead is equal to N×T/2, we present a selective time borrowing technique that can be used in non-critical PO gates of a pipeline stage. This is due to unequal SER contribution and slack distribution of different PO gates in different pipeline stages. Soft error

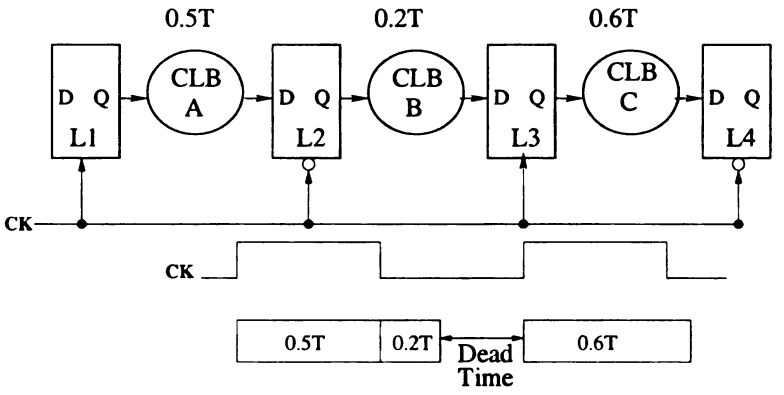

Figure 4.2. A pipeline with dead time.

rate of a PO gate i in stage j (SER($PO_{i,j}$)) refers to sum of soft errors of all it's fanin gates. A PO gate in pipeline stage j can borrow a maximum slack of $0.5 \times T$ (minus the latch $D-Q$ delay plus any jitter or skew, due to data settling close to clock closing edge) from pipeline stage $j+1$ and ahead. This time borrowed can be used to reduce SER contribution of PO gate in stage j by increasing slack available for sampling the output. However, time borrowed by $PO_{i,j}$ can reduce the slack available for another gate $PO_{i,j+1}$ in the fanout cone of $PO_{i,j}$. This requires us to borrow time based on the SER contribution of PO gates in a path. We present an algorithm in Figure 4.5, to do selective phase time borrowing for SER reduction, where two consecutive pipeline stages operating on the high and low phase of the same clock pass slack between them. This is done to make the problem of slack distribution across the pipeline stages simpler.

The algorithm assumes that the minimum clock cycle time in the latch-based pipeline has been determined and also the data arrival time at all the latches are known. The slack available for a non-critical $PO_{i,j}$ is $t_{crit} - t_{pd,i}$, where t_{crit} and $t_{pd,i}$

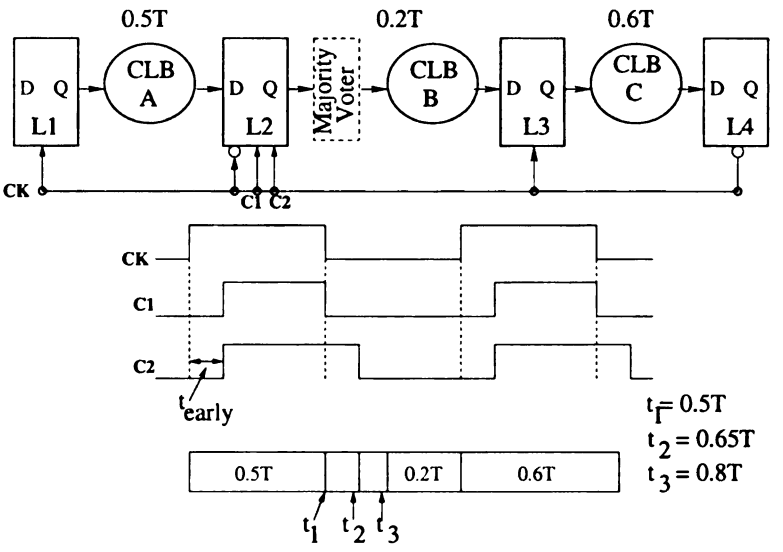

Figure 4.3. Dead time being used to increase the width of error pulse that can be tolerated.

are the critical path delay and the worst case propagation delay of the path ending at PO i, respectively. The time borrowing potential for each $PO_{i,j}$ is then calculated as the minimum slack among the PO gates in its fanout cone from stage $j+1$. The SER of $PO_{i,j}$ and all its fanout gates in stage $j+1$ are calculated when error masking technique is applied with the current slack available. Then the sum of SER of $PO_{i,j}$ and all its fanout gates are calculated and stored in $SSER_{i,j}$. The slack distribution between $PO_{i,j}$ and gates in its fanout cone is done based on the SER of $PO_{i,j}$. If SER of $PO_{i,j}$ is greater than the sum of SER of its fanout cone gates, slack available at $PO_{i,j}$ is increased through time borrowing. The slack is iteratively increased, in steps of 10% of the total time available for borrowing. During each iteration of the algorithm when steps 3-6 are executed, it is always made sure that the total sum of SER in pipeline stages j and $j+1$ remains same or decreases. This ensures that the algorithm converges, and the final SER of the pipeline system is equal to or lower

than the original SER. The results for SER reduction on ISCAS85 circuits due to time borrowing are presented in section 4.5.1.

4.5 Simulation Results

The EDR flip-flop, shown in Figure 4.1, was simulated using TSMC 180 nm models to calculate the values for t_{D-ck}, t_{D-C2}, and t_{ck-Q}. The value for t_{D-ck}, t_{D-C2} and t_{ck-Q} in the modified design were found to be 125, 115, and 50 ps, respectively. A delay chain capable of generating phase shifted clock signals every 200ps was constructed and hence the sampling times t_1'' and t_2'' were determined from the control signal availability. The complete methodology for determining the sampling times t_1'' and t_2'' are described in Chapter 5. Based on the setup time and CK-Q delay for the EDR flip-flop, we first estimate the slack S_{EDR} available at each flip-flop for sampling the PO values. The width of transient pulse that can be tolerated in the modified circuit t_w' is then calculated as $\min(t_3 - t_2', t_2' - t_1')$. The charge required to cause a transient pulse of width t_w' and $t_w = 100ps$ are then obtained from the lookup table. If $t_w' \leq t_w$, then we use EDR in the path, else error masking is used as described in Chapter 3. The results on applying EDR for ISCAS85 circuits are given in Table 4.1.

In Table 4.1, the column N_{trig} represents the number of flip-flops (FF) modified. Sub-column EM gives the number of FFs where error masking was used, while EDR gives the number of POs connected to flip-flops shown in Figure 4.1. As the average number of paths on which EDR is applied equals to 5.6, domino logic can be used to generate ED signals without much delay. The average SER reduction on using error masking alone is 82.67%, while combining error masking with EDR raised it to

Circuit	**Circuit Features**			N_{trig}		**SER Redn. %**		**Area Ovhd. %**	**Power Ovhd. %**
	Gates	PIs	POs	EM	EDR	EM	Both		
c432	210	36	7	3	4	55.66	89.66	30.8	72
c1908	1005	33	25	16	9	66.74	85.2	19.4	65
c2670	1498	233	128	44	4	99.58	99.81	30.8	52
c3540	2176	50	22	18	4	95.8	98.73	18.9	34
c7552	4785	207	108	58	8	81.14	90.14	22.63	27
c5315	3712	178	106	102	4	97.12	99.14	27.75	45
Avg.	2231	122.8	70.8	43.2	5.6	82.67	93.78	25.05	49.17

Table 4.1. SER reduction for ISCAS85 circuits. The power overhead in practice would be lower than the one presented above due to: (1) The original power has been estimated using zero delay model, which does not take into account glitchy or partial transitions. (2) The leakage energy, which has not been taken into account, consumed by the overhead circuit is far lower than the leakage of the CLB, due to fewer components.

93.78%. The original area of ISCAS85 circuits were obtained from Synopsys design compiler, while the area overhead is equal to the sum of area occupied by the delay lines and associated buffers, the modified FFs, circuit required to generate ED signals and a five percent wiring overhead. The overhead for generating PS signals are not included as the ISCAS85 circuits considered are not pipelined.

The area overhead depends on the number of modified FFs, the number of distinct sampling times and the maximum sampling time which contribute to the delay element overhead. If a number of sampling times are close together, then the delay element overhead can be reduced more (by clustering) without significant loss of SER reduction, as compared to circuits with sampling times wide apart. The delay lines can be shared across multiple modules which would further reduce their area as

well as power overheads. The power consumed by the original ISCAS85 circuits was estimated in Primepower using zero delay model (i.e. delays and switching activity have not been back annotated). The power overhead is calculated by simulating the delay and buffer chain in SPICE. The construction of the delay and buffer chain, along with SPICE details, are described in Chapter 5. The average power overhead in practice would reduce due to: (1) the original power has been estimated using zero delay model, which does not take into account glitchy or partial transitions; (2) the leakage energy, which has not been taken into account, consumed by the overhead circuit is far lower than the leakage of the CLB due to fewer components in the delay and buffer chain. In comparison to the overhead of 200% obtained from modified TMR techniques, the overheads incurred by the techniques presented are significantly lower.

The results presented here are for zero delay overhead i.e., the critical path delay is not affected. C499/C1355 which have the same overall function are not selected due to the presence of balanced paths in the circuit. As technology scales, clock frequency is increasing which decreases the absolute value of slack in circuits. However, as the time constant for charge collection process of a device decreases exponentially with minimum gate length [6], current pulse width due to particle strike also decreases. The decrease in current pulse width coupled with decrease in gate output capacitance leads to a decrease in the width of SET as technology scales. This should allow us to exploit the reduced slack available in a path to decrease SER using the technique discussed.

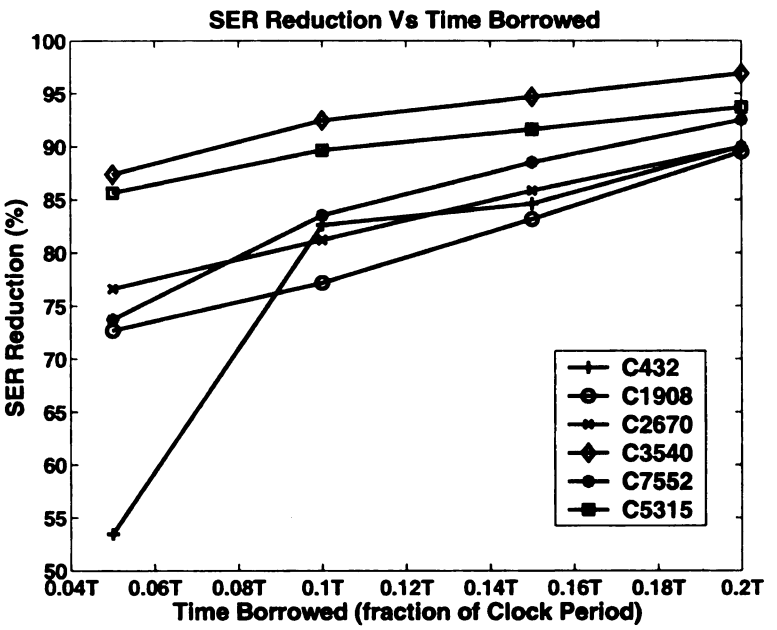

Figure 4.4. Plot showing the SER reduction achieved versus the time borrowed.

4.5.1 SER Reduction Using Slack Redistribution

In order to simulate the effect of time borrowing on SER reduction when using error masking alone, we increase the slack available across all paths and recalculated the SER reduction. Slack available in all paths were increased up to 0.2×T, where T is the clock cycle time. The results are plotted in Figure 4.4. As can be seen, SER reduces with small increase in the slack time. SER reduction in C432 increases from 52% to 82% as time borrowed is increased from 0.05×T to 0.1×T. This is because the number of latches that are triggered and used for sampling doubles from three to six when the slack across all paths is increased by 0.1×T.

4.6 Conclusion

A technique for doing both error detection and recovery (EDR) on short-slack paths was presented in ths chapter. This technique masks SETs of width approximately equal to half the clock cycle time. In case an error is detected, correction can be done with a single cycle penalty. This technique in combination with the error masking technique of previous chapter provides an average 93.78% SER reduction for ISCAS85 circuits. Two other techniques to improve the SER reduction provided by error masking were proposed. The first technique increases the slack available in logic circuits, by exploiting the circuit delay dependence on input vectors. The second technique utilized time borrowing in latch-based pipeline circuits to increase the slack. The potential for SER improvement on using these techniques was demonstrated in Section 4.5.

Algorithm Phase_Time_Borrow

Begin

1. **Initialization:**

   - Slack vector $\overrightarrow{S_j^{PO}}=\{s_{1,j}^{PO}, s_{2,j}^{PO}, \ldots, s_{n,j}^{PO}\}$ /* $s_{i,j}^{PO}$ is the slack at $PO_{i,j}$ in pipeline stage j. */
   - Potential time borrowable by $PO_{i,j}$: $tb_{i,j}=\min(s_{k,j+1}^{PO})$, $\forall\, PO_{k,j+1} \epsilon$ fanout cone (FOC) of $PO_{i,j}$.
   - $SER(PO_{i,j}) = \sum SER(g_{i,j})$, $\forall\, g_{i,j} \epsilon$ fanin cone of $PO_{i,j}$
   - $SSER_{i,j} = SER(PO_{i,j}) + \sum SER(PO_{k,j+1})$, /* sum of SER of PO gate i in stage j, and SER of PO gates in the fanout cone of $PO_{i,j}$ before slack redistribution /*
   - $SSER'_{i,j} = SER'(PO_{i,j}) + \sum SER'(PO_{k,j+1})$, /* SER' and SSER' are SER after slack redistribution */

For each PO i in stage j

2. **If** $(SER(PO_{i,j}) > \sum SER(PO_{k,j+1}))$, $\forall\, k \epsilon$ FOC of $PO_{i,j}$

Repeat

3. Calculate new sampling times and SER for $PO_{i,j}$ and all $PO_{k,j+1}$ in FOC of $PO_{i,j}$, when time borrowed from stage $j+1$ is $0.1 \times tb_{i,j}$.
4. Calculate $SSER_{i,j}$ and $SSER'_{i,j}$.
5. **If** $(SSER'_{i,j} < SSER_{i,j})$
6. Increment and decrement $s_{i,j}^{PO}$, $s_{k,j+1}^{PI}$ by $0.1 \times tb_{i,j}$, respectively.
7. Decrement $tb_{i,j}$ by $0.1 \times tb_{i,j}$.

Endif

Until $(tb_{i,j} > 0)$ **and** $(SSER'_{i,j} < SSER_{i,j})$

Endif

End For /* End of for loop */

End

Figure 4.5. Algorithm for time borrowing to reduce SER

CHAPTER 5

Robust Delay Chain Construction

5.1 Introduction

In this chapter, we explain the methodology for constructing the delay chain, which is used in both the EM and EDR techniques. First, we analyze three different families of delay elements for their robustness to process variation, and then determine the appropriate delay element for the delay chain construction [40]. Later in section 5.7, a delay chain, which produces control signals phase shifted from the system clock by every 200 ps, is constructed using the most robust delay element. Finally, we explain the construction of a buffer chain to distribute the phase shifted clock signals using the method of logical effort.

The three different delay element families analyzed are: (1) transmission gate based, (2) cascaded inverter based, and (3) voltage-controlled ones. We compare the delay element's effectiveness in terms of yield, which is defined as the number of circuits within a specified delay range. The delay variations are obtained through HSPICE

Monte Carlo simulations (MCSs) and the delay sensitivity to different process and environmental variations are studied using the simulation results. This enables us to select a robust delay element for constructing the delay line.

Process variation refers to random die-to-die and within die parameter fluctuations during the manufacturing process. The within-die variations can be classified as either correlated (systematic) or uncorrelated (random). Meindl suggests that correlated variations could occur due to aberrations in the stepper lens, whereas placement of dopant atoms in the device channel region, which varies randomly and independently from device to device within a die could cause uncorrelated variations [41]. As technology scales and transistor gate lengths become smaller than the wavelength of light used in the lithography process, the uncorrelated within-die parameter variations are expected to become a major design concern [41]. Therefore, we consider only random within-die parameter variations in this chapter. In the next few sections, we study the robustness of the delay elements under consideration.

5.1.1 Delay Elements

A *delay element* is a circuit that produces an output waveform similar to its input waveform, only delayed by a certain amount of time. Delay elements find wide use in digital systems [42]. Asynchronous or self-timed designs, in which the global clock is eliminated, make extensive use of delay elements [43]. Most asynchronous cells need to generate a completion signal to indicate that their outputs have been evaluated. A delay element can provide this as long as its delay amount is larger than the worst-case

delay of the cell [44]. For such structures as self-timed multipliers, delay elements are needed in the micropipeline [43]. Even circuits that perform complex mathematical calculations, such as computing the discrete cosine transform require delay elements in their architecture [45]. Finally, delay elements are used for phase modulation in delay-locked loops and phase-locked loops [46]. To our knowledge, there is no previous study on the effect of process variation on delay elements. Therefore, a study of delay elements vis-a-vis process variation would prove helpful in designing robust circuits.

5.2 Yield Definition

We define *yield* as the percentage of the total delay elements fabricated, whose propagation delay falls within a certain critical delay cut-off. In the presence of parameter variations, delays are distributed over a certain range. The distribution of delays can be evaluated by normalized variability - $\frac{3\sigma}{\mu}$, where σ and μ are the standard deviation and mean of the measured delay, respectively. As the proposed techniques are highly sensitive to delay of control signals, we define cut-off delay as $\pm 10\%$ of the mean delay.

5.3 Parameters Studied

The delay of a circuit depends on gate length L, width W, supply voltage V_{DD}, and NMOS (V_{Tn}) and PMOS (V_{Tp}) threshold voltages. These parameters are modeled as normally distributed random variables to study their effect on the delay variation. The standard deviation of each of these parameters and their nominal values for TSMC 180 nm are given in Table 5.1. The nominal value of the gate width for both PMOS and NMOS transistors in an inverter are given in Table 5.1. For other delay

	Nominal Value	$\frac{3\sigma}{\mu}$
Gate Length	180 nm	15%
Width	2/1 μm	15%
V_{Tn} (V)	0.445	15%
V_{Tp} (V)	-0.44	15%
V_{DD} (V)	1.8V	10%

Table 5.1. Parameter variations for the process considered.

elements, the width is scaled based on the circuit design and the delay required.

In all our simulations comparing delay elements, we fixed the fan-in to be a single minimum-sized inverter and the load was FO4 inverters. These inverters are provided with their own power supply, separate from the one connected to the delay element. The propagation delay of the delay elements is calculated by averaging the rise and fall delays. A square pulse with a slew of 50 ps was applied as input in all the experiments.

5.4 Simulation Methodology

Using MCSs performed in HSPICE, we sample a significant number of points from the normal distribution of each parameter and calculate the delay in each iteration. The resulting data gives the delay distribution, when the delay elements are manufactured under the given parameter variations. For each delay element we perform MCS with 500 iterations. We then calculate the mean, variance, and normalized variability for each delay element from the simulation results.

5.5 Delay Element Analysis and Yield Results

5.5.1 Transmission Gate Based Delay Element

A transmission gate (T-gate) is a bidirectional switch consisting of a parallel connection of an NMOS and a PMOS transistor that are controlled by complementary control signals as shown in Fig. 5.1(a). The NMOS and PMOS transistors pass logic 0 and 1, respectively, without degradation. By keeping the two transistors always on ($S = 1.8$V and $\bar{S} = 0$V in Fig. 5.1(a)), the transmission gate acts as a delay element.

Delay

The delay of a transmission gate is effectively determined by the time to charge or discharge a load capacitance C_L at its output through the equivalent resistance R_{eq}. The output voltage: $V_{out}(t) = (1 - e^{-t/R_{eq}C_L})V_{\mathrm{DD}}$. The propagation delay, which is the time taken for $V_{out}(t)$ to reach $V_{\mathrm{DD}}/2$, i.e., $V_{out}(t_p) = V_{\mathrm{DD}}/2$ is given by [44]:

$$\begin{aligned} t_p &= \ln(2) R_{eq} C_L \\ &= \ln(2) \frac{2V_{\mathrm{DD}}}{k_n(V_{\mathrm{DD}} - V_{Tn})^2 + k_p(V_{\mathrm{DD}} - |V_{Tp}|)^2} C_L. \end{aligned} \tag{5.1}$$

Here V_{Tn} and V_{Tp} denote NMOS and PMOS transistor threshold voltages, respectively, and k_n and k_p denote gain factors (which are proportional to the ratio of width over length ($\frac{W}{L}$)) of the two transistors. For a given fan-out, delay can be increased, compared to that of a minimum-sized transmission gate, by increasing L of the transistors, which linearly increases R_{eq}. Delay may be decreased by increasing W of the

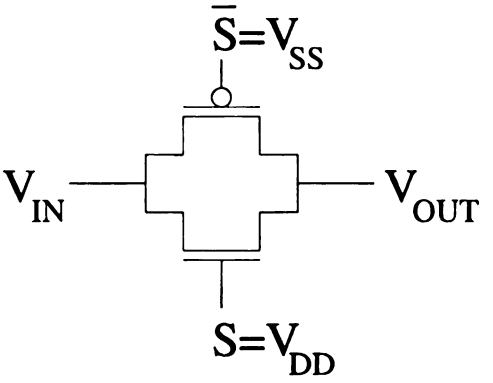

Figure 5.1. Schematic diagram of a transmission gate.

transistors, which decreases R_{eq}; however, this effect is limited by the diffusion (or junction) capacitance also increasing, which contributes to more load capacitance C_L. A chain of n transmission gates has a delay of [44]:

$$t_p(chain) = \ln(2) R_{eq} C_L \frac{n(n+1)}{2}, \tag{5.2}$$

where C_L is the load capacitance at the output of each transmission gate. Therefore, delay increases quadratically with the number of transmission gates in the chain and hence with area. A MCS with 500 iterations was done by varying parameters gate length, width, supply voltage V_{DD}, and threshold voltage within the range given in Table 5.1. The delay distribution plot when gate length, width, V_{DD}, and V_T are varied is given in Figure 5.2. As can be seen from Figure 5.2, the delay values are distributed around a mean value of 180 ps with significant number of delay values between 160 to 200 ps. The yield of the transmission gate delay element was found to be 97.8% and 95.8% when V_{DD} variation is 10% and 20%, respectively, and gate

length is varied by 10%. All other parameters were fixed to their nominal value. This shows that the yield of the transmission gate is affected significantly due to supply voltage variation.

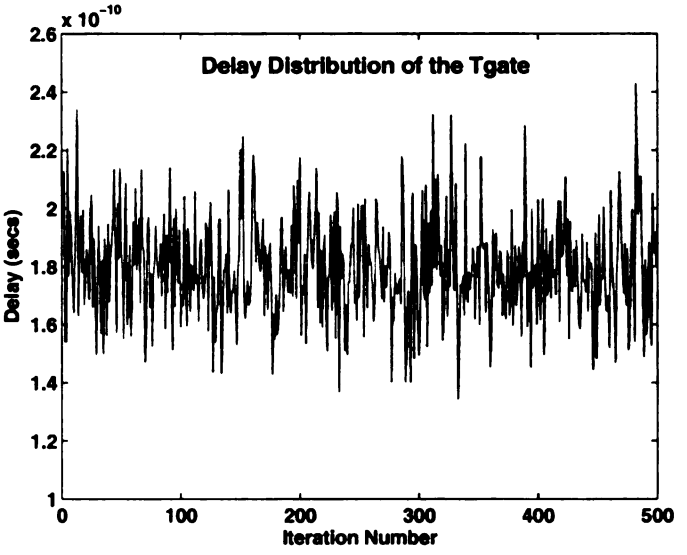

Figure 5.2. Delay of transmission gate for different iterations of a MCS.

5.5.2 Cascaded Inverter Based Delay Element

A pair of cascaded inverters can also function as a simple delay element that delays the input signal by an amount equal to the combined propagation delays of the two inverters (see Figure 5.3).

Delay

The propagation delay of an inverter depends upon the time taken to (dis)charge the load capacitance. An exact computation of this delay is nontrivial because of the nonlinear dependence of the (dis)charging current on the output voltage. An

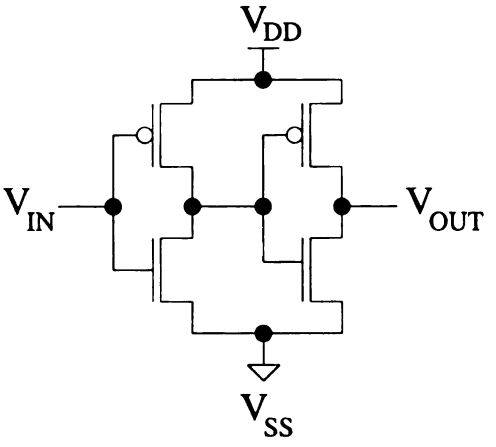

Figure 5.3. Schematic diagram of a cascaded inverter.

approximate expression is derived by using an average value of this current equal to the saturation current of the PMOS (NMOS) transistor given by:

$$\begin{aligned} I_{av} &= \frac{k_p}{2}(V_{GS} - |V_{Tp}|)^2 \\ &= \frac{k_p}{2}(V_{\rm DD} - |V_{Tp}|)^2 \approx \frac{k_p}{2}V_{\rm DD}^2. \end{aligned} \tag{5.3}$$

The above holds since $V_{\rm DD} \gg |V_{Tp}|, V_{Tn}$. Based on this I_{av} value, the propagation delay is as follows [44]:

$$t_p = \frac{1}{2}(t_{pLH} + t_{pHL}) = \frac{C_L}{2V_{\rm DD}}\left(\frac{1}{k_p} + \frac{1}{k_n}\right), \tag{5.4}$$

where t_{pLH} and t_{pHL} denote propagation delays for low to high and high to low output transitions, respectively. The above expression is valid when the input signal makes an abrupt transition from $V_{\rm DD}$ to V_{SS} or vice versa. The effect of a nonzero input rise time $t_r > t_{pHL}$ on propagation delay t_{pHL} is captured by the following

equation [44]:

$$t_{pHL(actual)} = \sqrt{t_{pHL(step)}^2 + (t_r/2)^2} \tag{5.5}$$

A MCS with 500 iterations was done by varying parameters gate length, width, supply voltage V_{DD}, and threshold voltage within the range given in Table 5.1. The delay distribution plot when gate length, width, V_{DD}, and V_T are varied is given in Figure 5.4. As can be seen from Figure 5.4, the variation in delay for the cascaded inverter occurs over a small range of 20 ps as compared to that of the transmission gate where the delay variation is spread over 40 ps. The delay of a cascaded inverter depends on the ratio of the gate length to V_{DD}, which leads to the small delay variation. The yield of the cascaded inverter delay element was found to be 100% and 99.8% when V_{DD} variation is 10% and 20%, respectively, with gate length variation set to 10%.

5.5.3 NP-Voltage Controlled Delay Element

An NP-voltage-controlled delay element is shown in Fig. 5.5(a). It consists of a cascaded inverter pair with an additional series-connected NMOS and PMOS transistor in the pull-down and pull-up of each inverter controlled by a global control voltage V_n and V_p, varying which changes the delay of this delay element.

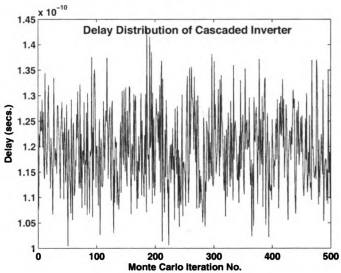

Figure 5.4. Delay of cascaded inverter for various Monte Carlo iterations.

Delay

The delay for this element can be altered by changing the control voltages V_n and V_p. One advantage is that the delay can be adjusted post-fabrication too. The (dis)charging takes place through a controlled transistor. The propagation delay of this element is:

$$\begin{aligned} t_p &= \frac{1}{2}\left(t_{pLH} + t_{pHL}\right) \\ &= \frac{C_L V_{\mathrm{DD}}}{2}\left(\frac{1}{k_p V_p^2} + \frac{1}{k_n V_n^2}\right) \end{aligned} \tag{5.6}$$

Note that in this case, t_p is inversely proportional to both ${V_p}^2$ and ${V_n}^2$. Both V_n and V_p are fed from a stable source, due to which they do not vary. A MCS with 500 iterations was done by varying parameters gate length, width, supply voltage V_{DD}, and threshold voltage within the range given in Table 5.1. The nominal value of

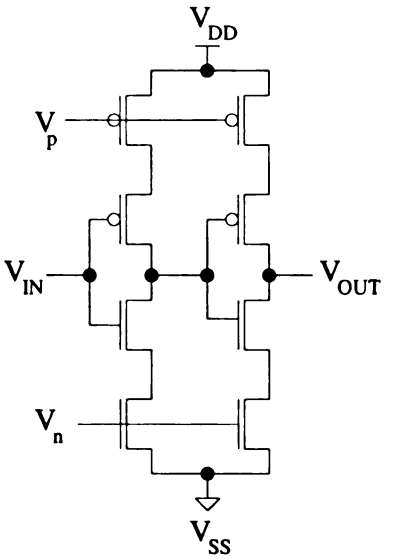

Figure 5.5. Schematic diagram of a NP-voltage cascaded inverter.

PMOS and NMOS widths were 4 μm and 2 μm, respectively. The delay distribution plot is given in Figure 5.6. Since parameter variations in both the stacked transistors affect delay, the variation is more as compared to that of the transmission gate. The yield of the NP-voltage controlled delay element has been found to be 92.2% and 84.8% when V_{DD} variation is 10% and 20%, respectively, and gate length variation is 10%.

5.6 Comparison of Delay Elements

In this section, we present results and analyze the delay sensitivity of the delay elements to various parameters.

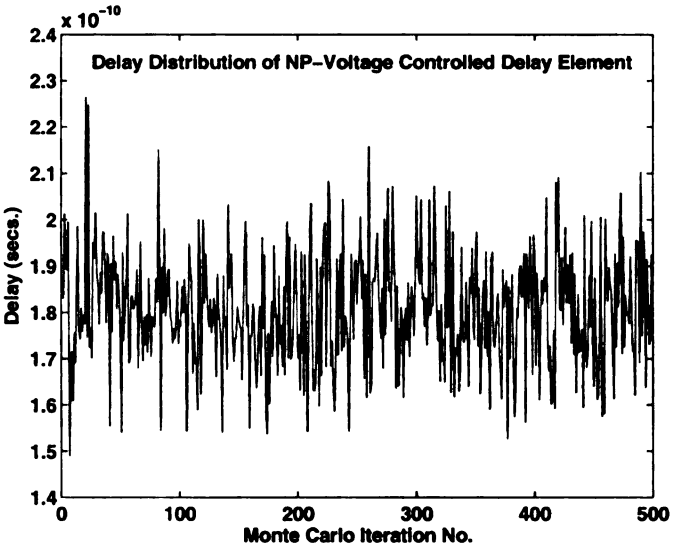

Figure 5.6. Delay distribution of NP-voltage controlled delay element for various Monte Carlo iterations.

5.6.1 Effect of V_{DD} and Gate Length Variation

The MCSs were run with gate width and threshold voltage fixed, while length and supply voltage were randomly varied. The 3σ variation of length was fixed at 10%, while 10% and 20% V_{DD} variations were applied. Table 5.2 presents mean, yield and 3σ variability for both 10% and 20% V_{DD} variation. NP-voltage delay element has the maximum variability of 28.5% among the three delay elements considered. Cascaded inverter is the most robust delay element as it has an almost 100% yield.

Delay element	Mean μ (ps)		3σ (%)		Yield (%)	
	10%	20%	10%	20%	10%	20%
Trans. gate	167.2	164.8	12.4	13.5	97.8	95.8
Cas. Inv.	225.3	224.6	7.9	8.43	100	99.8
NP-Volt.	193.5	194.6	21.1	28.5	92.2	84.8

Table 5.2. Mean delay and variability of the delay elements when V_{DD} variation is 10% and 20%, and gate length variation is 10%.

5.6.2 Effect of V_{DD} and Width Variation

We performed another set of experiments with gate length and threshold voltage fixed, while gate width and the supply voltage are varied. Table 5.3 shows delay results when V_{DD} and transistor width have 10% variation, with length constant. This table shows that the yield of the transmission gate, cascaded inverter, and the NP-voltage controlled delay element, increases, remains same, and decreases, respectively, when width changes randomly as compared to random variation in length. NP-Voltage delay element is more sensitive to width variation while transmission gate is more sensitive to variation in gate length.

Delay element	Mean μ (ps)	3σ (%)	Yield (%)
Trans. gate	165.54	11.9	98.2
Cas. Inv.	253.52	8.6	100
NP-Volt.	193.1	20.7	85.6

Table 5.3. Mean delay and variability of the delay elements when V_{DD} and gate width variation are 10%.

We now summarize the area, power, and signal integrity characteristics of the three different types of delay elements and offer some suggestions for choosing the appropriate delay element.

- The advantage of the transmission gate is the small area overhead and power dissipation. But it has poor signal integrity, which is defined as the maximum of rise and fall times. Moreover, the signal integrity and power consumed by the transmission gate degrade rapidly for producing large delays and hence the transmission gate is suitable for delays within 200-300 ps. The signal integrity

of transmission gate can be improved using Schmitt trigger, but it increases the area and power overhead significantly. Moreover, process variation can affect the Schmitt trigger too. Process variation also introduces uncertainty in rise and fall time, which can cause further delay variations.

- The cascaded inverter consumes more area and power than the transmission gate. Its signal integrity is good for delay values of less than 500ps. As the cascaded inverters have the highest yield, they can be used to construct delay chains with intermittent Schmitt triggers to provide higher delays. Some variations of the cascaded inverter such as replacing each inverter with a cascoded version with multiple PMOS and NMOS transistors in the pull-up and pull-down network can also be used to obtain higher delay values. However, cascoded inverters have lower robustness as compared to that of cascaded inverters, due to stacked transistors present in them.

- The NP-voltage controlled inverter's delay can be changed by altering its controlling voltage. Based on our experiments, we find that it has the least robustness to process variation and very poor signal integrity. It occupies a bigger area and consumes more power when compared to cascaded inverters because of the extra NMOS and PMOS control transistors. This delay element has a very poor signal integrity response and hence process variation can introduce more delay because of high slew uncertainty. Thus, we find that a cascaded inverter offers the best trade-off between area, power, and robustness metrics among the three delay elements considered.

5.7 Control Signal Generation and Distribution from Delay Chain

A delay chain which takes the system clock as an input and generates control signals used in the EM and EDR techniques is constructed using cascaded inverters. Each delay tap in the delay chain produces a control signal which is delayed from the previous tap or clock input by 200 ps, with the final control signal delayed by 2 ns from the clock input. The load driven by each tap in the delay chain is limited to 10-20 fF, so that the maximum delay variation at each delay tap is limited to ±10 ps from the intended 200 ps. This is done by limiting the gate driven by each delay tap to a minimum-sized-inverter or -AND gate.

The control signals from the delay taps are driven to the flip-flops using a buffer chain. A single inverter with varying drive strengths is used as a buffer. The load driven by the buffer chain and hence its delay depends on the number of flip-flops to which the control signal distributed by the buffer chain is routed. The methodology used for determining the load and delay of the buffer chains are summarized next.

1. Initially, only flip-flops in paths, which have half-the-slack ($S_{max}/2$) greater than the latching window time are chosen as candidates for error masking.

2. The sampling times t'_1 and t'_2 of the candidate flip-flops are determined based on the slack (S_{max}) available in the path, using steps explained in Chapter 3.

3. Then the initial t''_1 and t''_2 of the candidate flip-flops are determined based on the control signal availability, such that $t''_1 \geq t'_1$, and $t''_2 = t'_2 \pm \delta_t$.

4. Once the initial t_1'' and t_2'' are determined, the total flip-flops and hence the number of sampling transistors driven by each delay tap is known. The total capacitance for each delay tap is calculated as the sum of gate capacitances of the flip-flops driven by them. The interconnect capacitance is small compared to gate capacitance, as the delay chain is local to the CLB, and hence it is ignored.

5. Based on the capacitance driven by each delay tap, buffer chain which drives the control signals with minimum delay is constructed using the method of logical effort.

6. After the buffer chain construction, t_1'' and t_2'' are re-calculated based on the delay of the buffer chain and availability of the control signals.

7. Control signals which are used to sample data before $\frac{T}{2}$ are generated by ANDing CK and $\overline{DCK}$ as shown in Figure 3.4(ii).

We now demonstrate the construction of the buffer chain using logical effort. Figure 5.7 shows the number of flip-flops driven by each delay tap for the largest ISCAS85 circuit c7552. In each flip-flop, the control signals drive both a PMOS and NMOS transistor (sized 4/2 um), whose total width is double the minimum inverter size.

Logical effort states that the minimum path delay occurs when each stage in the path bears the same effort [47]. In logical effort, the unitless delay of a single stage in a path is given by:

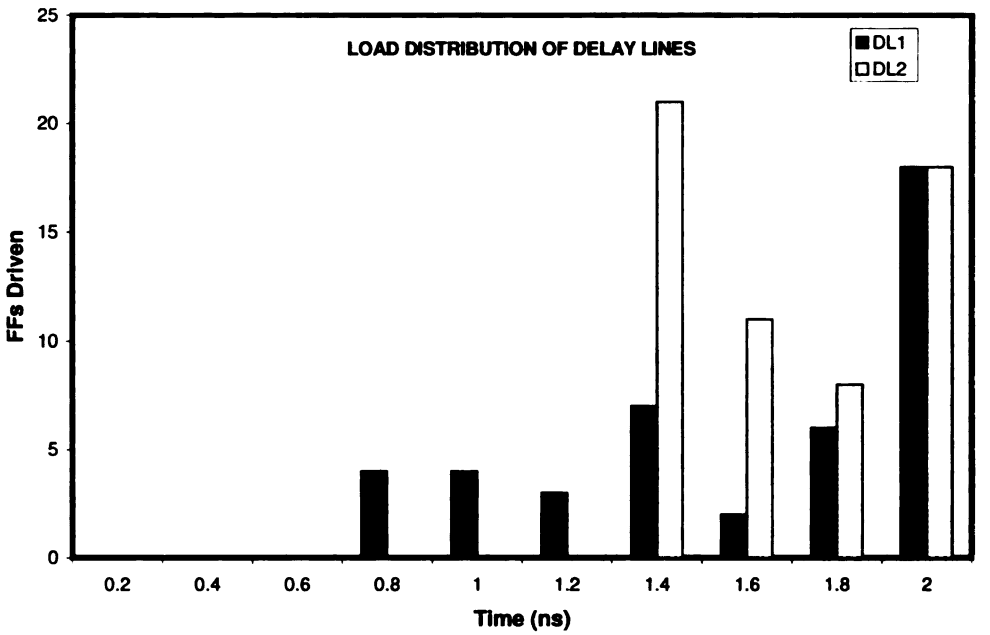

Figure 5.7. The number of flip-flops driven by each delay tap in the delay line of c7552. Two separate delay chains -DL1, DL2- are used to prevent soft errors from occurring due to particle strikes on the delay chain itself.

$$\begin{aligned} f_i &= g_i \times b_i \times h_i \\ d_i &= f_i + p_i. \end{aligned} \tag{5.7}$$

The parameters f_i, p_i, represent the stage effort and the unit-less parasitic delay of that stage, respectively. The absolute value of stage delay - $d_{abs} = d_i * \tau$, where τ is the technology time constant, defined as the average drive resistance of an inverter multiplied by its input capacitance. Each stage effort is in turn a product of the logical-(g_i), branching-(b_i), and electrical-effort (h_i) of that stage. Logical effort is a measure of a gates drive strength relative to a minimum sized inverter in the same technology. Electrical effort is the ratio of gates output to input capacitance.

Branching effort is a ratio of total output to off-path capacitance. The values for the above discussed parameters for a complete path are:

$$\begin{aligned} G &= \Pi g_i \\ B &= \Pi b_i \\ H &= \Pi h_i \\ F &= G \times B \times H \\ P &= \sum p_i \\ D &= F + P. \end{aligned} \tag{5.8}$$

As all stages in a path bear the same effort for minimum delay, the total delay can also be written as:

$$D = NF^{1/N} + P, \tag{5.9}$$

where N is the total number of stages in the path. The minimum delay is obtained when the partial derivate of D with respect to N is zero. This leads us to a relation between the parasitic delay (p_{buf}) and the best stage effort (γ), and is given by equation 5.9 [47].

$$p_{buf} = \gamma(log_e(\gamma) - 1) \tag{5.10}$$

As inverters are used as buffers, we determine the absolute value of p_{buf} by fitting a straight line through FO1, FO2, FO3, and FO4 delays of an inverter and measuring

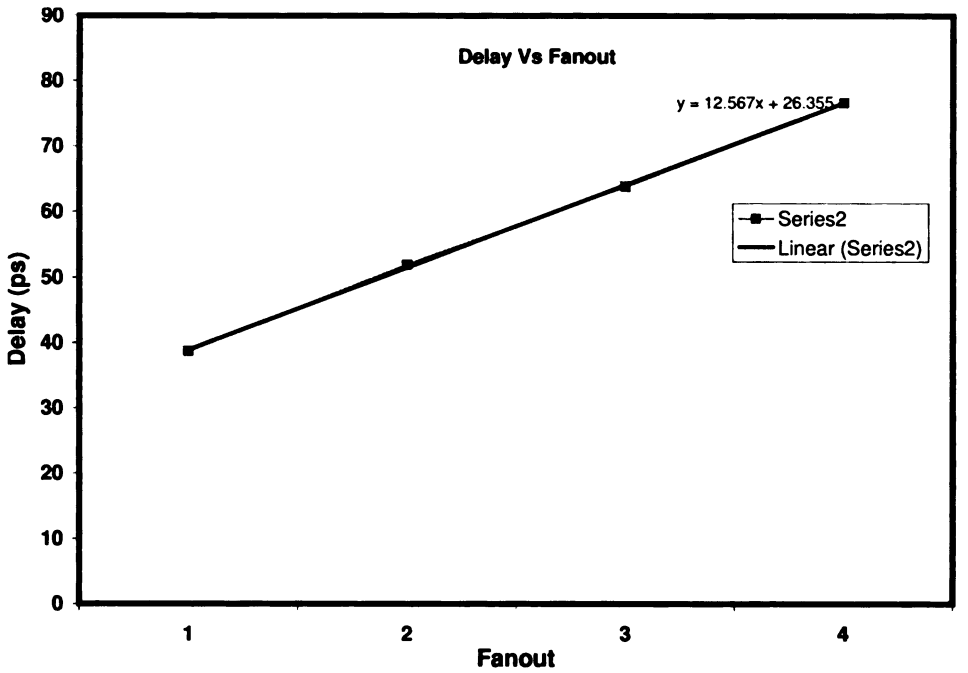

Figure 5.8. Delay versus fanout for an inverter in TSMC 0.18 micron technology. The absolute value of the parasitic delay of an inverter is the Y-intercept of the line shown, and has a value of 26.4 ps.

the Y-intercept of the line. The Y-intercept of the line, which is the delay of the inverter for zero external load, is the required value for the absolute value of p_{buf}. The straight line equation for an inverter in TSMC 0.18 micron is shown in Figure 5.8.

The unit-less p_{buf} is calculated by dividing its absolute value by τ. As the value of τ for TSMC 180 nm technology is approximately 13.13 ps, unit-less p_{buf}=2.01. Once the parasitic delay is known, Eq. 5.9 needs to be solved to obtain γ. As there is no closed-form solution for Eq. 5.9, we solve it graphically by plotting the value of $f(\gamma)(=\gamma(log_e(\gamma)-1))$ versus γ. The value of γ for which $f(\gamma)$ equals 2.01 is looked up from the graph, as shown in Figure 5.9, and is found to be 4.32. Hence, a stage effort of f_i=4.32 results in minimum delay through an inverter chain in the technology we

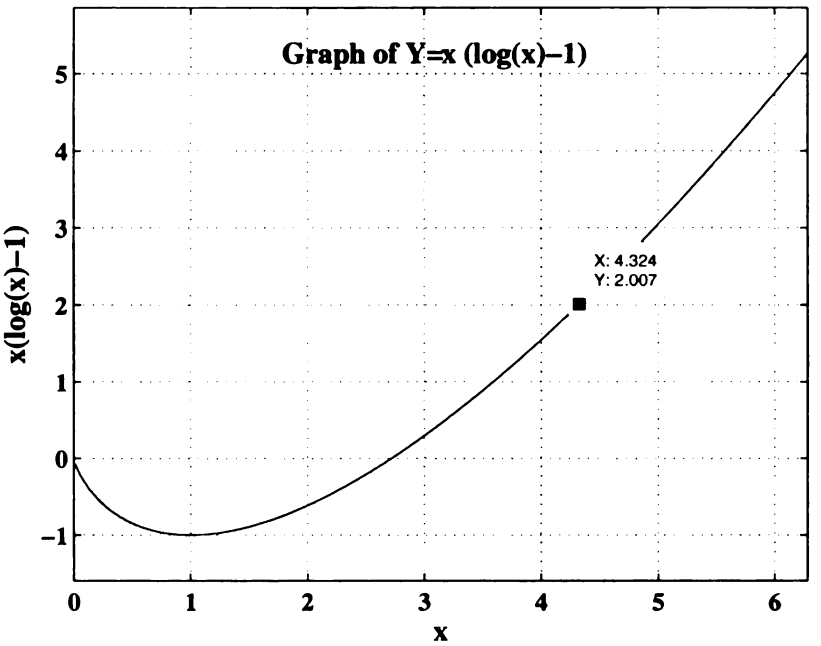

Figure 5.9. Graph used to find the best stage effort.

used. The total load at each delay tap is calculated based on the number of flip-flops driven by that tap. The path electrical effort (H) is ratio of the path load capacitance to input capacitance of a minimum sized inverter. The logical effort (G) of the buffer chain is one, as it is made up of inverters only. There is no branching in the path and hence B=1. The total path effort F is calculated as shown in Eq. 5.8. The number of stages or inverters in a path is calculated as $\log_{4.32}$F. We round off the number of stages to the next highest integer. Thus a buffer chain is constructed and its delays are simulated using HSPICE. The values of t_1'' and t_2'' are calculated by summing t_1' and t_2' with the delay of the buffer chain.

5.8 Conclusion

In this chapter, we analyzed the robustness of delay elements to process variation and explained the methodology used to construct a delay and buffer chain for our SER reduction techniques. A cascaded inverter was found to give a better yield under process variation, since its delay is less sensitive to V_{DD} and gate length variations. A delay chain with a delay tap every 200 ps was constructed using cascaded inverters. The delayed clock signals are then distributed using buffer chains. The construction of buffer chains with the least delay was demonstrated using the method of logical effort.

CHAPTER 6

Analysis and Design of Soft Error Hardened Latches

Previous study on soft error vulnerability of flip-flops and scannable latches considered latches designed without any explicit soft error protection [48]. As latches in commodity applications are being increasingly protected for soft errors, new soft-error hardened latch designs have been presented. In this chapter, we compare the performance and power cost of the existing designs and also propose efficient latch designs for soft error protection [49]. We use the following metrics to compare the existing latch designs. (1) Robustness of latches to charge collection at their drain nodes. We first investigate whether particle strikes on a latch change its output value. If the latch output changes, we determine if an error recovery function exists and the time taken for error recovery. If the latch output does not change value, we check if it is held stable by a static or dynamic node. If the latch output is held stable by a dynamic node, we study the effect of leakage on this stored value. (2) Soft error

protection to transient pulses originating from a combinational logic block (CLB). (3) Robustness to single event multiple-upsets. (4) Setup, hold time and the Data-to-output (D-Q) delay. (5) Power overhead of the soft error hardened latches. (6) Issues such as power and performance cost to be considered for system-level integration, especially when using some of the latch designs for CLB protection.

The chapter is organized as follows. Section 6.1 explains the simulation setup for measuring the critical charge, the latch delay and the power consumption. In Section 6.2, we analyze the various existing soft error hardened latches based on the metrics presented above. Section 6.3 presents our proposed latch designs for soft error immunity, some of which are affected by SEMUs on more than two nodes only. Finally, we conclude in Section 6.4.

6.1 Simulation Methodology

6.1.1 Latch Delay and Power Calculation

All simulations were done in TSMC 0.18 micron technology with a supply voltage V_{DD} of 1.8V. All the latches were designed with minimum sizes for the sake of comparison. To calculate setup and hold time, all the latch outputs were connected with a fanout of four inverter load (F04). The setup time t_s is defined as the minimum D-to-CK offset that causes the Data-to-output (D-Q) delay to be 5% higher than its nominal value [50]. Based on this definition of setup time, the minimum clock cycle time when flip-flop (FF) A is driving FF B is given by :

$$T \geq 1.05 \cdot t_{D-Q,A} + t_{Logic} + t_{setup,B} + t_{skew} \tag{6.1}$$

The first term of Eq. 6.1 accounts for the worst-case D-Q delay of FF A, when data arrives exactly one setup time before the active clock edge. The second term, t_{Logic}, captures the worst case propagation delay through the combinational logic, while the third parameter, t_{skew}, captures the clock skew. The D-Q delay is the delay measured from the active clock edge to the output. It depends on the clock slope and the output load, apart from the D-CK offset. The clock slope was fixed at 50 ps both in the rising and falling directions. The total delay of the latch is defined as the sum of D-Q delay (measured at the setup time) plus the setup time itself. The total delay of a basic transmission gate latch, similar to the one in Figure 6.1, (but without the explicit capacitance in node fb) was first calculated. The delay values of all other latches are reported after normalizing with the standard latch delay. The power consumed by the latch is calculated as the average of the power consumed for latching a logic 0 and 1. The energy values reported here are per clock cycle. Similar to delay, the power values are normalized with respect to the transmission gate latch, for ease of comparison.

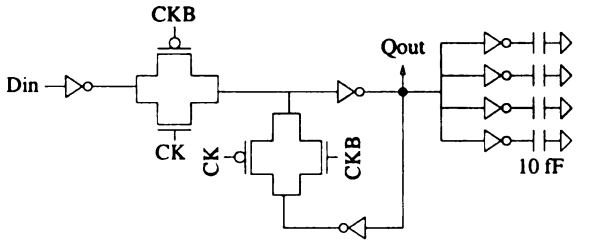

Figure 6.1. Basic transmission gate latch used to normalize delay and power values of other latch designs. The delay and power values were measured by connecting a FO4 inverter at the latch output.

6.2 Comparison of Latch Designs

6.2.1 SEU Tolerant Latch

The schematic of a SEU tolerant latch latch is shown in Figure 6.2 [2]. The latch stores data D at PP and NP, while $\overline{D}$ is stored at QP and QN. PP and QP are driven only by PMOS transistors while NN and QN are driven only by NMOS transistors. This latch utilizes the fact that only a 0→1 flip can occur in a PMOS, and only a 1→0 flip can occur in an NMOS, due to which nodes QP and QN are soft error hardened. Particle strikes at any of the two nodes PP or NN does not change the output Q, however they cause node Q to become dynamic for a short period of time.

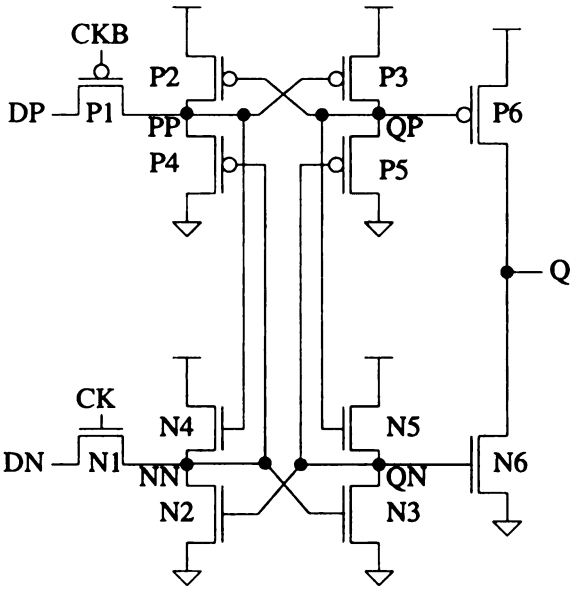

Figure 6.2. Schematic of single event upset tolerant latch.

This latch can be used to prevent soft errors due to particle strikes in combinational logic. This requires inputs DP and DN to be fed from two different CLBs or delayed by certain time interval. Delaying the PO to create DP and DN could introduce

performance overhead for the circuit. To avoid performance overhead such latches can be used only in non-critical paths. The maximum transient pulse width tolerated by using this latch in a path with slack S is $S/2$. If the latch delay - t_{D-Q} - is considered then the transient pulse width tolerated reduces to $(S - t_{D-Q})/2$. Introduction of time delay between DP and DN requires a delay chain inside each latch, which adds to the overall system area and power overhead.

The probability of a SEMU affecting more than two nodes is very low due to high energy of the particle required to cause multiple upsets. Hence, we consider SEMUs occurring on two nodes only. There are six different node combinations which need to be analyzed for SEMUs. The four node combinations PP-NN, QP-QN, PP-QP, NN-QN are not affected by SEMUs, since simultaneous logic flips can not occur on these node pairs. However, SEMUs occurring in QN-PP or QP-NN combinations have the potential to cause soft errors. But by carefully spacing them apart in the latch layout, the critical charge required to cause an upset can be made quite high, and hence the latch can be assumed to be SEMU hardened for most sea-level applications. The delay of this latch was found to be five times the original latch delay, due to its high D-Q delay. This latch consumes static power due to some of the PMOS and NMOS not completely turning off. This latch consumes 52% more power than the original latch.

6.2.2 Soft Error Hardened Latch Scheme for SoC

The schematic of the latch is shown in Figure 6.3 [51]. The latch stores $\overline{D}$ at node DH, and D at nodes PDH and NDH. Node DH is kept static by either transistor P1 or N1, depending on the value of input D, after CK becomes low. Whenever a particle strike occurs at the node DH, a glitch occurs at the latch output Q. For example, when DH flips from 1→0 transistor N2 is cut-off. But node PDH can be maintained at zero by the parasitic capacitances, which enables P1 to pull DH back to logic one. Therefore, transistors P1 and N2 should be sized larger than P8, such that DH is pulled to logic one before P8 pulls PDH to one. The width of a glitch at the output node due to a logic flip at DH is small, as node DH is restored to its correct value within 50 ps, even at V_{DD}=0.8V. While using this latch in a pipelined circuit, it has to be taken care that this glitchy output is not capable of causing an error in the next stage. A particle strike at PDH does not affect nodes DH and Q. For example, in case of a 0→1 logic flip in PDH, transistor N2 pulls node PDH back to logic 0 and P1 also turns ON. Particle strikes on transistors P7, N3 and N7, P3 also do not change DH, and hence the output value. It has been experimentally verified that soft errors with charges up to 1000 fC are corrected by this latch [51].

This latch design can only handle SEUs on the latch itself. Any transient pulse with width equal to the latching window of the modified latch could cause a soft error if it is not logically or latching window masked. The probability of soft errors occurring due to particle strikes in CLB is same as any original latch without soft error hardening. Now we consider the possibility of SEMU causing an error. Let's

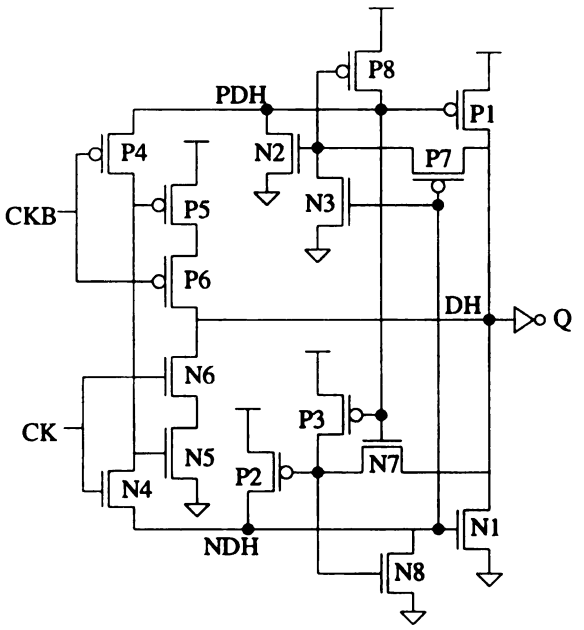

Figure 6.3. Schematic of soft error hardened latch.

consider again DH being held at logic 1 by P1. When a particle strike flips both DH, from 1→0, and PDH, from 0→1, then the output value flips permanently, until new inputs arrive. Thus, soft errors can be caused due to SEMUs on the latch. For both DH and PDH to flip as mentioned, sufficient electrons have to accumulate around N7 or N1 drain, or sufficient holes should accumulate around P8. This can be avoided by increasing the spacing between these transistors in the layout.

The delay of this latch was reported to be 5-20% higher than that of the original latch. Higher delay overhead is due to the transistors P1 and N1 being sized bigger to reduce SEU at DH. The latch consumes 4-6% more power than the latch in Figure 6.1 [51]. The SER reduction is 25x for neutrons and 99x for alpha particles compared to the original latch.

6.2.3 Dual Interlocked Storage Cell

The schematic of the latch is shown in Figure 6.4 [52, 53]. The latch stores the data values at D0a, D0b, D1a and D1b, which are vulnerable to particle strikes. Let's assume the latch stores logic 0 at D0a and D0b. When a particle strike flips D0b from 0→1, N1a is enabled and thus D1a may reach an intermediate voltage between logic 0 and 1. A glitch may also occur at output node Q because of a flip in D0b. Nodes D0a and D1b maintain their value dynamically. As D1b stays at 1, D0b discharges to 0 through N2b and N3b. The effect of leakage on the recovery time was studied in [53]. The leakage effect was studied by connecting four current sources of same value to nodes D0a, D0b, D1a and D1b. Significant increase in SER was observed when the leakage current was 20% of static current noise margin (I_{nm}). Thus, this latch design is sensitive to leakage, but whether the leakage observed is 20% of I_{nm} needs to be evaluated based on the process technology used.

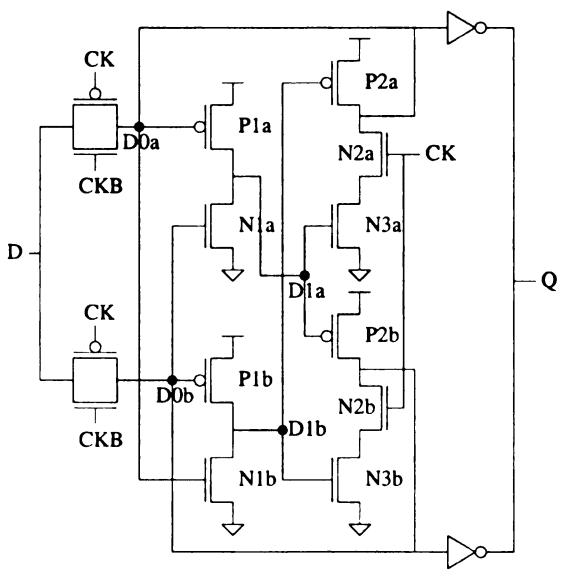

Figure 6.4. Schematic of dual interlocked storage cell.

The dual interlocked storage cell (DICE) design does not mask transient pulses originating in CLB. If the input D is split, and temporally separated before feeding it to transmission gates driving D0a and D0b, then the latch becomes unstable, and the output Q could flip to the wrong value. A variant of DICE, which uses a C-element in each of its inputs to protect SETs generated in CLBs, was presented in [54]. The C-element is fed both the normal and a delayed input, with delay equal to the width of the SET to be tolerated. This introduces big performance penalty and high area overhead due to separate delay chains being used in every flip-flop. This latch is also vulnerable to SEMUs, because all the nodes are vulnerable to both 0→1 and 1→0 logic flips. This causes the stored value to flip permanently until a new input is applied.

The worst case delay and power penalties of this latch as compared to that of the standard latch was reported to be 2-3% and 34%, respectively [53]. This latch provides 10x SER reduction as compared to an original latch.

6.2.4 Single Event Resistant Topology Latch

The schematic of the latch is shown in Figure 6.5 [55]. The clocked transistors and buffers are not shown in the schematic for clarity. The data values are stored at both Y0 and Y1, while their complements are stored at Y2 and Y3. The cross-coupling of the transistors prevents an upset at any one of the four nodes from changing the output value and hence the relative sizing of the transistors does not matter as compared to latches in Figure 6.2 and Figure 6.3. Let us consider an initial case when

nodes Y0 and Y1 are both 1 and Y2 and Y3 are at 0. If a particle strike flips Y0 from 1→0, transistors N1c, N0d are disabled while P0d is enabled, which flips Y3 to 1. Y2 still retains 0 because N0c remains ON and P0c is disabled. As Y2 is still at 0, P0a charges Y0 and brings it back to 1, which makes Y3 to go low. Thus, the initial state of the latch is restored. A glitch could possibly occur at the output of the latch.

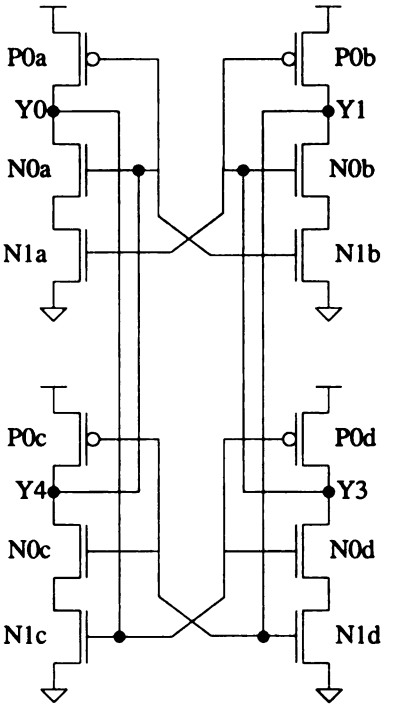

Figure 6.5. Schematic of single event resistant topology.

The proposed latch can also be used to protect transient faults originating in the CLB. This was reportedly done by duplicating the CLB, and connecting the output of each block to the right (b,d) and left sections(a,c) of the latch shown in Figure 6.5 [55]. This could also be achieved by using temporal separation of PO signals and using the original and delayed versions to drive the right and left sections. This latch is sensitive to SEMUs. For example, a particle strike which can flip both Y0 and Y1

from 1→0 will change the data stored permanently without any chance of recovery, until the next input is applied. Similar to previous approaches, the nodes YO-Y3 can be spaced apart to reduce probability of SEMUs.

The delay of this latch was found to be 37% higher than the original latch. The power consumed by the latch was found to be 70% of the original latch due to the complex interconnection structure for holding the data values.

6.2.5 Other Latch Designs

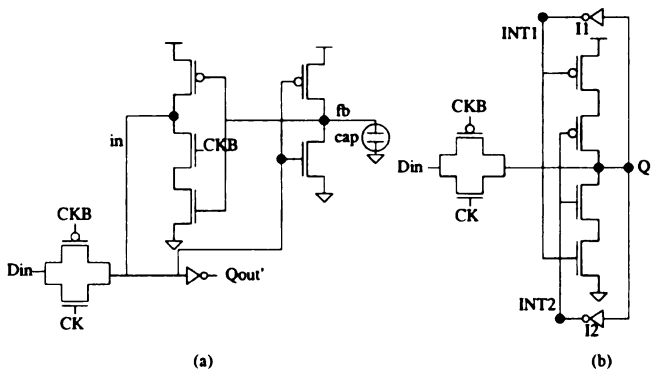

Figure 6.6. Hardening of the feedback node in a latch (a) Feedback node fb hardened by adding explicit capacitances. (b) Feedback node hardened by duplicating feedback inverters.

A few other designs which try to harden latches against soft errors have been presented. A latch design which hardens the feedback node (fb) in a latch was presented in [35]. The schematic of the latch is shown in Figure 6.6(a). An explicit gate capacitance is added to fb to increase the charge required to flip the node. This leads to a 2x reduced SER of the latch, but degrades the speed of the latch. Another latch design which hardens the feedback node was presented in [56]. The schematic

of the latch is shown in Figure 6.6(b). This latch uses two inverters I1 and I2 in the feedback instead of one. A particle strike on INT1 and INT2 alone cannot produce an upset. However, a particle strike on Q can produce a soft error. Also, SEMUs are not tolerated by this latch.

6.3 New Latch Designs with Soft-Error Immunity

We propose new latch designs which utilize some of the techniques used in asynchronous circuits. The schematic of the two basic latch designs are shown in Figure 6.7(a) & (b). Latch A stores data at nodes D0 and D1 which are held static (after CK goes high) by transistors P2 and N2, respectively. This means nodes D0 and D1 do not reach true 0 and 1 (V_{dd}). Due to this, transistors P3 and N3 are not enabled completely. In latch B, the nodes D0 and D1 (which store data) are held static by N2 and P2, with the help of inverters I1 and I2, which means P3 and N3 are enabled completely. Therefore, latch A has a higher delay compared to latch B, while latch B consumes more power due to the inverters and full voltage swing at nodes D0 and D1. Both the latches are negative-level sensitive, open when the clock is low. The vulnerability of latches to particle strikes are analyzed by looking at the effect of charge collection at data storage nodes. We only consider transistors P2 and N2, as only these transistors have their drain/source nodes connected to D0/D1. In latch A, charge collection at DO or D1 can only turn P3 and N3 off, respectively. In latch B, charge collection at D0 or D1 can only turn ON P3 and N3. This leads to different vulnerabilities for latches A and B.

A SEU in latch A, at either D0 or D1, only makes D' dynamic as transistors P3

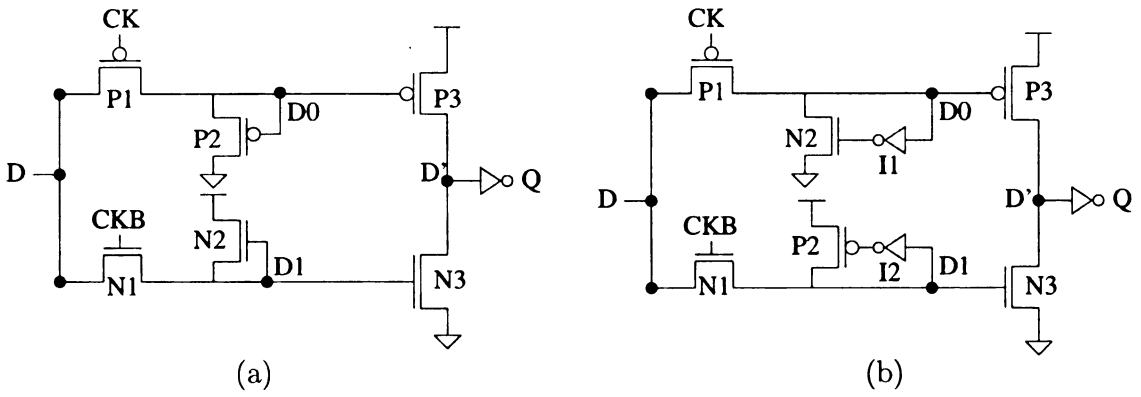

Figure 6.7. (a) Latch A vulnerable only to SEMUs. (b) Latch B having lower delay and higher power consumption than latch A, but vulnerable to SEUs.

and N3 can only be disabled. Therefore, latch A is susceptible only to SEMUs, while in latch B soft error can occur due to SEUs at either D0 or D1. A particle strike at D0 or D1 in latch B can cause D' to reach an intermediate voltage between 0 and V_{dd}, which could lead to a wrong value at output Q. In latch A, SEMU at either D' and D0 (or) D' and D1 could cause a soft error. A SEMU at nodes D' and D0 or D' and D1 in latch B creates a temporary 1→0 or 0→1 glitch at node Q which finally settles at a voltage between 0 and V_{DD}. Stick diagram for the layout of latch A, such that SEMU vulnerable nodes are spaced apart is shown in Figure 6.8.

The delay of latch A and latch B were found to be 1.12x and 0.67x of the latch shown in Figure 6.1. As analyzed before latch B has lower delay and lower recovery time for a particle strike. However, the power consumed by latch B was found to be 5x compared to original latch, while latch A just consumes 40% power of the original latch.

The two basic latch configurations A & B are presented to explain the concept of

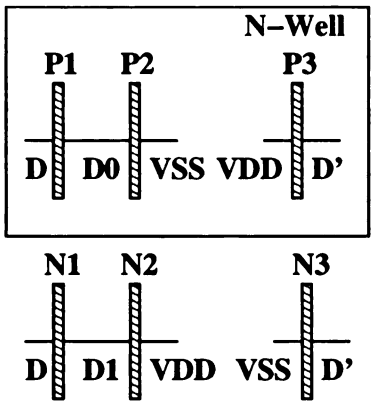

Figure 6.8. Stick diagram for layout of latch A, such that nodes D'-D0 and D'-D1 are spaced apart with minimum area overhead.

new latch designs. Latch designs which have more redundancy and provide increased soft error tolerance as compared to latches A and B, without having too much area and power overhead, are shown in Figure 6.9(a) & (b). Again we analyze the susceptibility of the latches by studying the effect of charge collection around the data storage nodes D0-D5. Simultaneous charge collection on both nodes D4 and D5 can result in a voltage flip at output node Q. As nodes D4 and D5 are restored to their initial value after a short duration, only a small glitch results in Q. SEMUs on nodes D4 and D0 or D5 and D3 can only result in output node Q reaching a voltage between 0 and V_{DD}. Latch C is not susceptible to simultaneous charge collection at any combination of two and three data nodes. Only a SEMU on four nodes could lead to a logic flip at node Q. Thus latch C is much less vulnerable compared to latch A. Particle strikes which cause SEMU at nodes Q and D4 (or) Q and D5, only cause a glitch at output node Q. As both the node combinations Q-D4 and Q-D5 are kept static all the time, the probability of such a glitch occurring is low. Thus SEMU in latch C could either

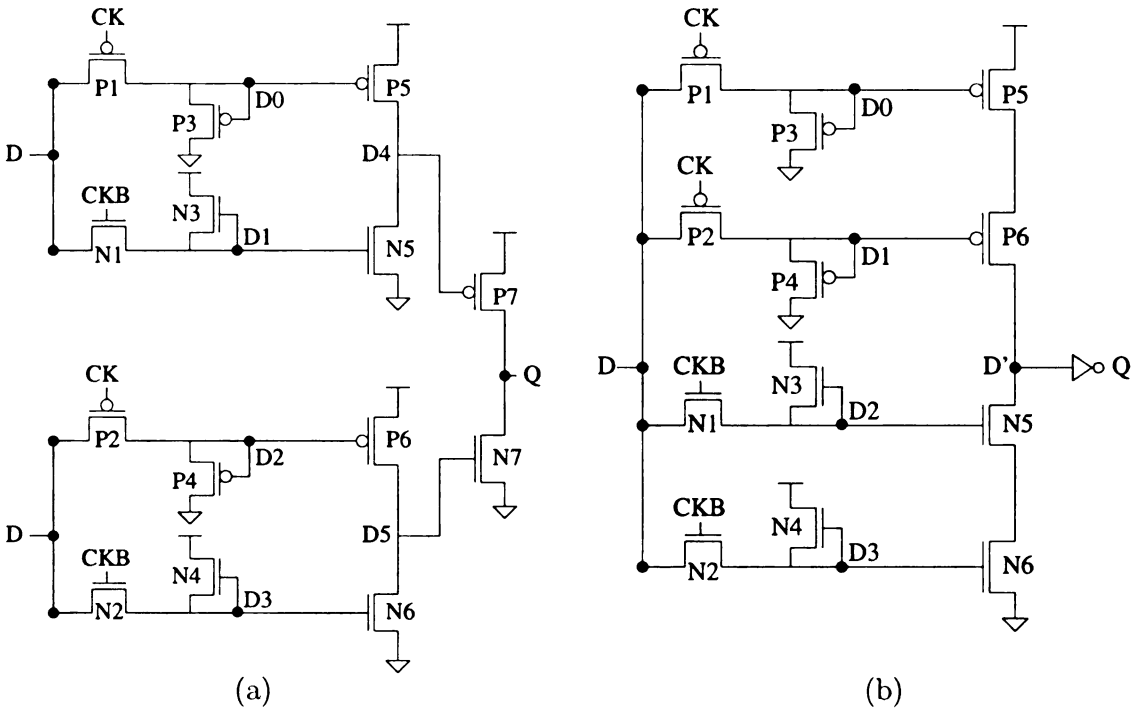

Figure 6.9. (a) Latch C having best power, performance, and soft error immunity. (b) Latch D can be used to provide soft error protection for CLBs also.

lead to Q being at an intermediate voltage between 0 and V_{DD}, or cause a glitch, or in the worst-case of SEMU at four nodes a complete logic flip. To avoid node Q from reaching an intermediate voltage between 0 and V_{DD}, layout of latch C with data nodes spaced apart is shown in Figure 6.10.

In the case of latch D, shown in Figure 6.9(b), charge collection at the data nodes D0-D3 only turns off P5, P6 and N5, N6. Therefore, SEMU on only two nodes, such as D0 and D1 or D2 and D3, does not result in an error at output Q, but it turns D' into a dynamic node. Leakage in transistors N5 and N6 should be considered while using this latch configuration. An error in latch D can occur when SEMU changes the value stored in D', D0, and D1, or D', D2, and D3. Spacing apart these nodes

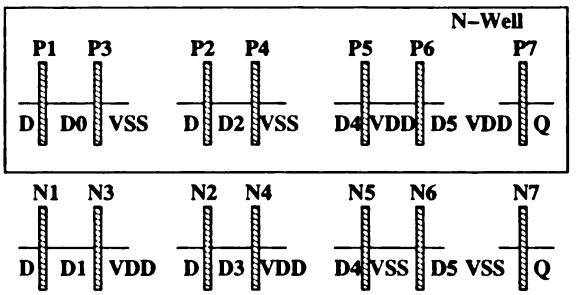

Figure 6.10. Stick diagram for layout of latch C, such that nodes D4-D0 and D5-D2 are spaced apart with minimum area overhead.

would make these latches SEMU resilient.

As most of the new latch designs are susceptible to only SEMU the actal SER reduction compared to original latch is difficult to calculate through simulation. Hence, we have presented only analytical results for the soft error tolerance of these latches. The new latch designs apart from being vulnerable only to SEMUs, also have significant lower area and power cost compared to earlier latches surveyed. The latches C and D can be customized according to application requirements for speed and power. The latch D can also be used to provide soft error protection for transient faults in CLB. This can be done by creating temporally separated signals D and D' and driving for example D0-D2 with D and D1-D3 with D'. The latches in Figure 6.7 designed with only two nodes for storing data D0 and D1, could be used when soft error protection for CLBs is not necessary. Table 6.1 presents the delay and power overhead of the proposed designs compared to the standard latch.

The delay of latch C is 1.2x of the standard latch and that of latch D is 2.3x. Latch C also consumes just 74% power of the original latch. Hence, latch C is the best configuration considering power, performance, and delay. But latch D which

Latch type	Delay (ps)	Power (μ W)
Standard Latch	194.92 (1x)	55.7 (1x)
Latch C	239.45 (1.2x)	41.32 (0.74x)
Latch D	455.37 (2.3x)	37.3 (0.67x)

Table 6.1. Delay and power overhead of the proposed latch designs.

consumes just 67% of original power can be used to provide protection for CLBs also.

6.3.1 Customizing Latches for Performance and Power Requirements

The latches presented in Figure 6.9 can be customized based on the power and speed requirements. Latch designs which need to be optimized for speed could use storage nodes similar to latch B, while those optimized for power could use storage nodes similar to latch A. For example, to improve speed at the cost of increased power, inverters at selective data nodes can be added to latches C and D. Two such configurations are shown in Figure 6.11. The speed and power consumption of the customized latches are shown in Table 6.2. Both the latches E and F have lower delay and higher power compared to latches C and D, respectively. Both latches E and F are vulnerable to SEMUs on more than two nodes only. Hence, the soft error vulnerability of these latches are similar to that of C and D.

Latch type	Delay (ps)	Power (μ W)
Latch E	172.6	260.64
Latch F	429.20	707

Table 6.2. Delay and power overhead of the customized latch designs.

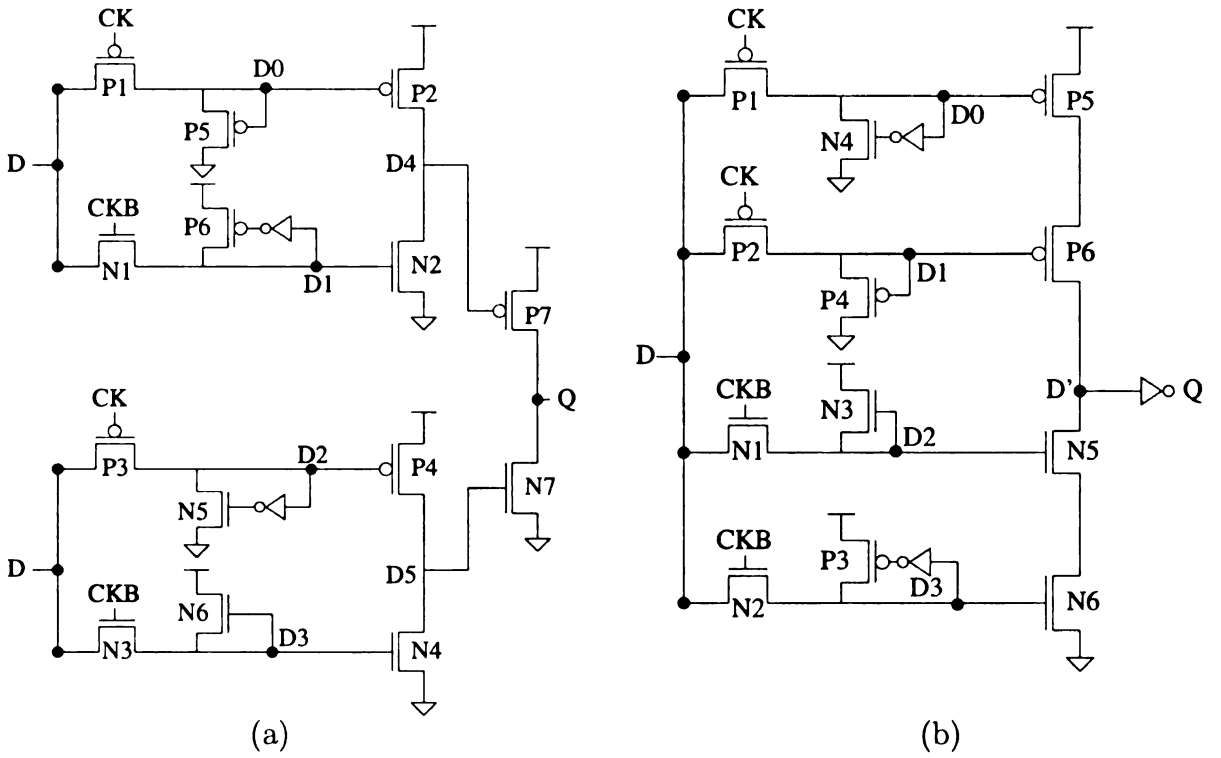

Figure 6.11. (a) Latch C customized for reduced delay with higher power cost. (b) Latch D can be used to provide soft error protection for CLBs also.

6.4 Conclusion

In this chapter, we analyzed existing latch designs for their soft error vulnerability, their power, and performance overheads. The latch design presented in [51] was found to provide good trade-off between power, performance, and soft error protection. However, this latch cannot be used for soft error protection in CLBs. We also proposed new latch designs, the best of which is vulnerable only to SEMUs with a performance penalty of 1.12x and which consumes only 40% power compared to a standard transmission gate latch. The latches C and D can be customized according to application requirements for speed and power. Latch D can also be used to provide soft error protection for transient faults in CLB. This can be done by creating temporally separated signals and driving the signals D0-D3 separately.

CHAPTER 7

Conclusion

In this dissertation, we presented our research on soft-error modeling and mitigation techniques for logic circuits. Performance-, power-, and area-efficient techniques to reduce the soft-error vulnerability of both combinational and sequential logic circuits, along with an LUT-based methodology to estimate the SER reduction of these techniques was explained in depth. The important contributions and results from this research are summarized in detail below.

7.1 Key Contributions

In Chapter 2, we described a fast and accurate LUT-based methodology to calculate both SET width due to particle strikes, and SER reduction for time-redundancy-based error mitigation techniques. Previous techniques for SET width calculation use complex expressions or large LUTs and have greater than 15% error for inputs not close to pre-characterized points. We studied the sensitivity of an SET to various gate and circuit characteristics, and determined the parameters to be used, their

spacing, and their lower and upper-bounds for constructing the LUT. The LUT uses non-uniform spacing and surface-based interpolation between its indices to obtain the SET width generated at a gate and primary output. We found the LUT to provide greater than 1000 times speedup compared to HSPICE simulations, with less than 10% error.

In Chapter 3, we presented an efficient and systematic error masking (EM) technique that can be applied to combinational logic circuits which have a significant fraction of non-critical paths with sufficient slack. This error masking technique prevents an SET pulse of width less than approximately half of the slack available in the propagation path from latching and turning into a soft error, without any performance overhead. Previous techniques incur a performance overhead of $2w$ for masking an SET pulse of width w. We control flip-flops only in paths with sufficient slack which ensures that the delay increase caused by the addition of majority voter and control transistors to the flip-flops does not affect the timing of the overall circuit. Additionally, our technique uses a single delay chain which produces phase-shifted signals used to sample POs. The results obtained on ISCAS85 benchmark circuits show an average SER reduction of 82.67% from the original unprotected circuit.

In Chapter 4, we presented a design technique to combine error masking with error detection and recovery (EDR). This technique can be used to improve the reliability of a circuit without sufficient number of non-critical paths for applying the EM technique. The EDR technique tolerates transient pulses with width up to half a clock cycle period. In case a soft error occurs, a very low-probability event for an application run, and is detected, recovery can be completed within a single

clock cycle. The results obtained show an average SER reduction of 93.78% for the EM+EDR method. Apart from the EDR technique, we also described simple and efficient methods, using input vector characteristics and time borrowing in latch-based pipeline circuits, to improve the SER reduction obtained from the error masking technique. Both soft-error mitigation techniques presented in this dissertation can be used to reduce transient faults caused not only due to particle strikes, but also due to cross-talk or power supply noise.

We explained the construction of the delay chain used in EM and EM+EDR techniques explained in Chapter 5. We first analyzed the robustness of three different families of delay elements to process variation using Monte Carlo simulations in HSPICE. A cascaded inverter was found to give a better yield under process variation, since its delay is less sensitive to V_{DD} and gate length variations. A delay chain with a delay tap every 200 ps was constructed using cascaded inverters. The delayed clock signals are then distributed using buffer chains. The construction of buffer chains with the least delay was demonstrated using the method of logical effort.

In Chapter 6, we analyzed existing latch designs for their soft-error vulnerability and overheads. The latch proposed in [51] was found to provide good trade-off for power, performance, and soft-error robustness. However, this latch cannot be used for soft-error protection of CLBs. We also proposed new latch designs, the best of which is vulnerable only to SEMU with a delay overhead of just 12% and consumes only 40% power of a standard transmission gate latch. Further, we presented efficient approaches to layout the proposed latch designs, such that vulnerability to SEMUs can be reduced. Finally, we presented two configurations of the proposed latch designs,

customized according to application power and performance requirements.

Our SER mitigation work represents a significant advancement over previous approaches which, in contrast, rely on introducing explicit hardware or time redundancy or on redundant computation, often both. Consequently, our methods provide substantial energy and performance/hardware advantages.

7.2 Future Work

Three potentially fruitful directions for future research are briefly outlined next.

1. The delay chain and distribution of the control signals contributed to the maximum power overhead. The power overhead can be reduced using low-voltage-swing delay chain and control signals. The low-voltage-swing control signals should be converted to full $0 \rightarrow V_{DD}$ swing before they can be fed to the flip-flops using efficient level converters. This is necessary to ensure correct operation of the flip-flops. Therefore, generation and distribution of low-voltage control signals should be investigated.

2. The distribution of slack for improving error masking using time-borrowing was performed between two successive pipeline stages in Chapter 4. The potential of this technique can be increased further by doing slack redistribution across the entire pipeline. Future work can consider slack redistribution across pipeline stages and explore the feasibility of obtaining a globally optimal solution for this problem.

3. The SERs of logic circuits vary by orders of magnitude depending upon the

application being executed. Efficient and accurate SER characterization of logic circuits for various applications can be performed. This will be useful in customizing SER reduction techniques to lower overhead based on the most frequent application executed. Such customized techniques are beneficial for embedded processors, most of which are used only in single-application systems.

BIBLIOGRAPHY

[1] Semiconductor Industry Association, "The international technology roadmap for semiconductors," http://www.itrs.net/Common/2004Update/2004Update.htm, 2004.

[2] K.J. Hass, J.W. Gambles, B. Walker, and M. Zampaglione, "Mitigating single event upsets from combinational logic," in *Proc. 7th NASA Symposium on VLSI Design.* 1998, NASA.

[3] P. Hazucha, *Background radiation and soft errors in CMOS circuits*, Ph.D. thesis, Linkoping University, Sweden, 2000.

[4] K. Bernstein, "High speed CMOS logic responses to radiation-induced upsets," in *Designing Robust Circuits and Systems with Unreliable Components Workshop*, 2002.

[5] M. J. Gadlage et al., "Single Event Transient Pulsewidths in Digital Microcircuits," *IEEE Transactions on Nuclear Science*, vol. 51, no. 6, pp. 3285–3290, 2004.

[6] Premkishore Shivakumar, Michael Kistlerand, Stephen W. Keckler, Doug Burger, and Lorenzo Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," in *Proc. International Conference on Dependable Systems and Networks*, June 2002, pp. 389–398.

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

[8] S. Hareland, J. Maiz, M. Alavi, K. Mistry, S. Walsta, and C. Dai, "Impact of CMOS process scaling and SOI on soft error rates of logical processes," in *Symposium on VLSI Technology, Digest of Technical Papers.* 2001, pp. 73–74, IEEE.

[9] Tanay Karnik, Bradley Bloechel, K. Soumyanath, Vivek De, and Shekhar Borkar, "Scaling trends of cosmic rays induced soft errors in static latches beyond 0.18u," in *Symposium on VLSI Circuits Digest of Technical Papers.* 2001, pp. 61–62, IEEE.

[10] S. Mukherjee, C. Weaver, J. Emer, S.K. Reinhardt, and T. Austin, "A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor," in *Proc. IEEE/ACM International Symposium on Microarchitecture*, Dec. 2003.

[11] C. Constantinescu, "Impact of deep submicron technology on dependability of VLSI circuits," in *Proc. International Conference on Dependable Systems and Networks*, June 2002, pp. 205–209.

[12] R. Baumann, "Soft Errors in Advanced Computer Systems," *IEEE Design and Test of Computers*, vol. 22, no. 3, pp. 258–266, May 2005.

[13] J.F. Ziegler et al, "IBM experiments in soft fails in computer electronics," *IBM Journal of Research and Development*, vol. 40, no. 1, pp. 3–19, 1998.

[14] T. C. May and M. H. Woods, "Alpha-particle induced soft errors in dynamic memories," *IEEE Transactions on Electron Devices*, vol. 26, no. 2, 1979.

[15] R. Baumann, "Soft errors in commercial semiconductor technology: overview and scaling trends," in *IEEE 2002 Reliability Physics Tutorial Notes, Reliability Fundamentals*, Apr. 2002, pp. 121.1–121.14.

[16] H. Ando et. al., "A 1.3GHz fifth generation SPARC64 microprocessor," in *Proc. IEEE/ACM Design Automation Conference*, June 2003, pp. 702–705.

[17] M. Santarini, "Cosmic radiation comes to ASIC and SOC design," *Electronics Design Network*, pp. 46–56, 2005.

[18] Cadence Design Systems, "Pacific: User guide," 2004.

[19] L.W. Massengill, A.E. Baranski, D.O. Van Nort, J. Meng, and B. Bhuva, "Analysis of single-event effects in combinational logic-simulation of the AM2901 bitslice processor," *IEEE Transactions on Nuclear Science*, vol. 47, no. 6, pp. 2609–2615, Dec. 2000.

[20] K. Mohanram, "Closed-form simulation and robustness models for SEU-tolerant design," in *Proc. International VLSI Test Symposium*, Apr. 2005, pp. 327–333.

[21] Y. S. Dhillon, A. U. Diril, and A. Chatterjee, "Soft-error tolerance analysis and optimization of nanometer circuits," in *Proc. Design Automation and Test in Europe*, Mar. 2005, pp. 288–293.

[22] L.B. Freeman, "Critical charge calculations for a bipolar SRAM array," *IBM Journal of Research and Development*, vol. 40, pp. 119–129, Jan. 1996.

[23] D.G. Mavis and P.H. Eaton, "Soft error rate mitigation techniques for modern microcircuits," in *IEEE Reliability Physics Symposium*, 2002, pp. 216–225.

[24] G. Hubert et al., "Study of basic mechanisms induced by an ionizing particle on simple structures," *IEEE Transactions on Nuclear Science*, vol. 47, no. 3, pp. 519–525, 2000.

[25] S. Krishnamohan and N.R. Mahapatra, "A highly-efficient technique for reducing soft errors in static CMOS circuits," in *Proc. IEEE International Conference on Computer Design (ICCD)*, Oct. 2004.

[26] J.C. Lo, "A novel area-time efficient static CMOS totally self-checking comparator," *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 165–168, Feb. 1993.

[27] C. Metra, M. Favalli, and B. Ricco, "Self-checking detection and diagnosis of transient, delay, and crosstalk faults affecting bus lines," *IEEE Transactions on Computers*, vol. 49, pp. 560–574, June 2000.

[28] L. Anghel and M. Nicolaidis, "Cost reduction and evaluation of a temporary faults detecting technique," in *Proc. Design Automation and Test in Europe*, 2000.

[29] J. B. Nickel and A. K. Somani, "REESE: A Method of Soft Error Detection in Microprocessors," in *Proc. International Conference on Dependable Systems and Networks*, June 2001, pp. 401–410.

[30] M. Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," in *Proc. International VLSI Test Symposium*, 1999.

[31] K. Mohanram and N. A. Touba, "Partial error masking to reduce soft error failure rate in logic circuits," in *Proc. International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2003, pp. 433–440.

[32] Q. Zhou and K. Mohanram, "Cost-Effective Radiation Hardening Technique for Combinational Logic," in *Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Oct. 2004.

[33] M. Zhang and N.R. Shanbag, "An energy-efficient circuit technique for single event transient noise-tolerance," in *Proc. IEEE International Symposium on Circuits and Systems*, May 2005, pp. 636–639.

[34] H. Cha and J.H. Patel, "Latch design for transient pulse tolerance," in *Proc. IEEE International Conference on Computer Design (ICCD)*, Oct. 1994, pp. 385–388.

[35] T. Karnik, S. Vangal, V. Veeramachaneni, P. Hazucha, V. Erraguntla, and S. Borkar, "Selective node engineering for chip-level soft error rate improvement," in *Symposium on VLSI Circuits Digest of Technical Papers*, June 2002, pp. 204–205.

[36] J. Grad and J. E. Stine, "A standard cell library for student projects," in *International Conference on Microelectronic Systems Education*, 2003, pp. 98–99.

[37] S. Krishnamohan and N.R. Mahapatra, "Combining Error Masking and Error Detection Plus Recovery to Combat Soft Errors in Static CMOS Circuits," in *Proc. International Conference on Dependable Systems and Networks*, June 2005.

[38] D. Ernst et al., "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. IEEE/ACM International Symposium on Microarchitecture*, Dec. 2003.

[39] K. Bernstein et al., *High speed CMOS design styles*, Kluwer Academic Publishers, first edition, 1998.

[40] S. Krishnamohan and N.R. Mahapatra, "An Analysis of the Robustness of CMOS Delay Elements," in *Proc. Great Lakes Symposium on VLSI*, Apr. 2005.

[41] K.A. Bowman, S.G. Duvall, and J.D. Meindl, " Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE Journal of Solid-State Circuits*, pp. 183–190, 2002.

[42] G. Kim, M.-K. Kim, B.-S. Chang, and W. Kim, " A low-voltage, low-power CMOS delay element," *IEEE Journal of Solid-State Circuits*, pp. 966–971, 1996.

[43] Y. W. Pang et al., "An asynchronous cell library for self-timed system designs," in *IEICE Transactions on Information and Systems*, Mar. 1997, pp. 296–305.

[44] J.M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital integrated circuits*, Prentice Hall, first edition, 1996.

[45] M. F. Aburdene, J. Zheng, and R. J. Kozick, "New recursive VLSI architectures for forward and inverse discrete cosine transform," in *Proceedings of SPIE - The International Society for Optical Engineering*, 1996.

[46] A. W. Buchwald, K. W. Martin, and A. K. Oki, " A 6GHz integrated phase-locked loop using AlGaAs/GaAs heterojunction bipolar transistors," *IEEE Journal of Solid-State Circuits*, pp. 1752–1762, 1992.

[47] I. Sutherland, B. Sproull, and D. Harris, *Logical effort: designing fast CMOS circuits*, Morgan Kaufmann Publishers, first edition, 1999.

[48] R. Ramanarayanan et al., "Analysis of soft error rate in flip-flops and scannable latches," in *Proc. IEEE International System on Chip Conference*, Sept. 2003.

[49] S. Krishnamohan and N.R. Mahapatra, "Analysis and Design of Soft Error Hardened Latches," in *Proc. Great Lakes Symposium on VLSI*, Apr. 2005.

[50] D. Markovic, B. Nikolic, and R. Broderson, "Analysis and design of low-energy flip-flops," in *Proc. IEEE/ACM International Symposium on Low Power Electronics and Design*, 2001, pp. 52–55.

[51] Y. Komatsu et al., "A soft-error hardened latch scheme for SOC in a 90nm technology and beyond," in *Proc. IEEE Custom Integrated Circuits Conference*, Oct. 2004.

[52] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron CMOS technology," *IEEE Transactions on Nuclear Science*, pp. 2874–2878, Dec. 1996.

[53] P. Hazucha et al., "Measurements and analysis of SER tolerant latch in a 90nm dual-Vt CMOS process," in *Proc. IEEE Custom Integrated Circuits Conference*, Oct. 2003, pp. 617–620.

[54] R. Naseer and J. Draper, "The DF-DICE storage element for immunity to soft errors," in *Proc. IEEE Midwest Symposium on Circuits and Systems*, 2005.

[55] J. Gambles et al., "An ultra low-power, radiation-tolerant reed solomon encoder for space applications," in *Proc. IEEE Custom Integrated Circuits Conference*, Oct. 2003.

[56] M. Omana, D. Rossi, and C. Metra, "Novel transient fault hardened static latch," in *Proc. International Test Conference*, Sept. 2003, pp. 886–892.