NOVEL FLIP-FLOP DESIGNS TOLERANT TO SOFT-ERRORS AND CROSSTALK EFFECTS

BY ADITYA JAGIRDAR

A thesis submitted to the
Graduate School—New Brunswick
Rutgers, The State University of New Jersey
in partial fulfillment of the requirements
for the degree of
Master of Science
Graduate Program in Electrical and Computer Engineering

Written under the direction of
Prof. Michael L. Bushnell
and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2007

ABSTRACT OF THE THESIS

Novel Flip-Flop Designs Tolerant to Soft-Errors and Crosstalk Effects

by Aditya Jagirdar

Thesis Director: Prof. Michael L. Bushnell

The desire to make technology faster, smaller and more affordable compels us to shrink transistors further. As we realize designs with millions of transistors, most of the existing problems increase in severity and newer problems crop up. One major new problem is *Soft-Errors* in logic and the result is a severe decrease in circuit reliability. This problem has been common in static memories since 1970 and, hence, fault-tolerant memory techniques are well developed. However, soft-errors today affect sequential logic as well. Interconnect crosstalk gets severe as we move towards higher operational frequencies and must be dealt in conjunction with soft-errors.

In this work, we propose novel flip-flop designs, which, unlike previous designs, are immune to soft-errors and crosstalk effects during the entire *Window of Vulnerability* (WoV), even around the clock edge. The *Crosstalk and Soft-Error Tolerant Flip-Flop* (XSEUFF2) can recover from transient pulses generated in the combinational logic and on internal nodes of the master and slave latches. It is also tolerant to any signal delays arising due to crosstalk. The area, timing and power overheads of this design over Mitra's *Basic Scan Flip-Flop* (BSFF) are 37%, 30% and 250% while those of Mitra's *Error Blocking Scan Flip-Flop* (EBSFF) and *Error Trapping Scan Flip-Flop*

(ETSFF) are 13%, 23%, 213%, 15%, 5% and 226%. The area overhead of Roy's *Error Blocking Scan Hold Flip-Flop* (EBSHFF) is 9% lower than that of *BSFF*, while the timing and power penalties are 25% and 72%, respectively. Designs of the *EBSFF*, *ETSFF* and *EBSHFF* are vulnerable to soft-errors affecting the master latch around the active clock edge and hence, do not provide total immunity from soft-errors, particularly around the active clock edge. Further, they are not tolerant to crosstalk effects.

We also calculate overheads for more ISCAS '89 benchmark circuits and the average overhead of the *XSEUFF2* is about 20%. Thus, with reasonable increase in area, timing and power penalties we design a flip-flop completely tolerant to soft-errors and crosstalk. In conjunction with the *XSEUFF2*, we also propose the *Crosstalk Tolerant Flip-Flop* (XTFF) and the *XTFF2* that are immune to only crosstalk effects and incoming transients from combinational logic. They have much lower overheads and have a different level of trade-off between reliability and performance.

Acknowledgements

I would like to thank Prof. Tapan Chakraborty for his invaluable guidance during the course of this work. His encouragement and inspiration goes well beyond the academic realm. I would also like to thank Prof. Michael L.Bushnell for his support and permission to work with Prof. Chakraborty. Roystein Oliveira, a good friend and researcher, has been an integral part of my life here and his constant feedback has played a significant role in my work. I would also like to thank my friends from the VLSI research group, Omar, Hari and Rajamani who partook in all my activities, from gate-crashing conference luncheons to avoiding ones we were supposed to attend.

Dedication

To my Mother

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

The role of electronics in every branch of science has become pivotal. Integrated circuits, digital and analog, are used extensively in applications ranging from medical to home-automation. As the demand for systems with greater functionality increases, it becomes imperative that we develop newer technologies that provide greater chip density. Transition from existing to newer technologies entails overcoming issues varying from accurate modeling of smaller topologies to their fabrication and field reliability. As we move further into deep sub-micron technology, i.e., 70nm and below, certain undesirable effects that were less severe and hence did not affect circuit performance, have now become critical factors, detrimental to chip reliability. The most significant of these problems is generation of spurious noise pulses when a high-energy particle (such as a neutron or α particle) impinges on a circuit node. Under certain conditions, this may cause a temporary change in the state of the circuit. Errors induced in such a way are commonly referred to as *Soft-Errors*[67, 72]. Another source of noise pulses that we consider in this work is the mutual coupling between transmission lines, which is also known as *Interconnect Crosstalk*. We discuss the above mentioned effects in great detail in the following sections.

1.1 Contribution of this Work

Work presented in this thesis assumes a simplistic model of a SET and achieves an optimum trade off between space and time redundancies. Concurrent error recovery is the main target of our fault-tolerance schemes. The proposed flip-flop design, *Crosstalk and Soft-Error Tolerant Flip-Flop* (XSEUFF2) is tolerant to soft-errors

throughout the entire *Window of Vulnerability* (WoV) including the time around the active clock edge. It can also recover from noise pulses and signal delays arising due to interconnect crosstalk. This design is vastly superior to existing solutions in terms of higher reliability and incurs reasonable design overheads. The area, timing and power overheads of this design over Mitra's *Basic Scan Flip-Flop* (BSFF) are 37%, 30% and 250% while those of Mitra's *Error Blocking Scan Flip-Flop* (EBSFF) and *Error Trapping Scan Flip-Flop* (ETSFF) are 13%, 23%, 213%, 15%, 5% and 226% [41]. The area overhead of Roy's *Error Blocking Scan Hold Flip-Flop* (EBSHFF) [20] is 9% lower than that of *BSFF*, while the timing and power penalties are 25% and 72%, respectively. The *EBSFF*, *ETSFF* and *EBSHFF* are vulnerable to incoming transients on the data line, around the active clock edge. With shrinking feature sizes, this becomes a critical reliability issue. Thus, we propose a design that has an average area overhead of only 20% on the ISCAS '89 benchmark circuits and is capable of handling both soft-errors and crosstalk effects.

We also propose two other designs, the *Crosstalk Tolerant Flip-Flop* (XTFF) and *XTFF2* that are immune to all crosstalk effects, that is, noise pulses and signal delays. These designs have significantly lower design overheads than the XSEUFF2 and represent different points on the reliability-performance curve.

1.2 Introduction to Soft-Errors

During every clock cycle of operation of a chip, each circuit node is charged either to a '0' or a '1.' This charge is stored temporarily by the node capacitance, which is the cumulative capacitance between various terminals of a transistor. To ensure correct operation of the circuit, each node must reach a stable voltage during the clock cycle and remain at that voltage until the end of that clock cycle. Any unexpected disruption of this node voltage, either due to a faulty circuit or a temporary noise pulse, can propagate to the fanout gates and eventually affect the primary output or a *Pseudo-Primary Output* (PPO) [9]. Transients generated in existing designs are mostly a result of bombardment of high-energy particles on these sensitive nodes.

Since these effects usually last only for a short duration (relative to the clock cycle) and seldom cause permanent damage, they are referred to as soft-errors.

1.2.1 Sources of Soft-Errors

Particles from two sources are mainly responsible for the total number of soft-errors occurring in a device[72]:

- Decay of radioactive impurities, which remain in trace amounts in the packaging material, and
- Extra-terrestrial cosmic rays that bombard the earth from the far depths of outer space.

The particles that have a higher probability of causing soft fails among all those emitted by the packaging materials are α-particles. They are emitted during the decay processes of Uranium, Thorium, Polonium (Po^{210}) and Lead (Pb^{210}). Pb^{210} decays to Po^{210} with a half-life of 22 years. Hence, trace quantities of these elements remain with the packaging material even after many years. The other source of soft fails lies in the cosmic radiation received by Earth. Nuclear reactions between cosmic rays and the packaging materials can create additional electron-hole pairs, which then drift to various nodes and diffusion regions of transistors in the vicinity. The manner in which these high-energy particles interact with various components of packaging is complex. Intergalactic particles with a mean energy of $2GeV$ and a flux of $0.2/cm^2$-s can penetrate to sea-level.

Figure 1.1 plots the variation of cosmic flux with altitude. It can be inferred that flux can vary in the field by a factor of $10\times$ or more.

However, during their journey through the atmosphere, cosmic particles break up, creating a cascade of more complex particles and what hits the sea-level is usually a sixth generation particle. Hence, the final batch of particles includes not only protons, neutrons and electrons but also pions and muons, which are harder to analyze. Figure 1.2[72] illustrates the generation and scattering of this spectrum of particles.

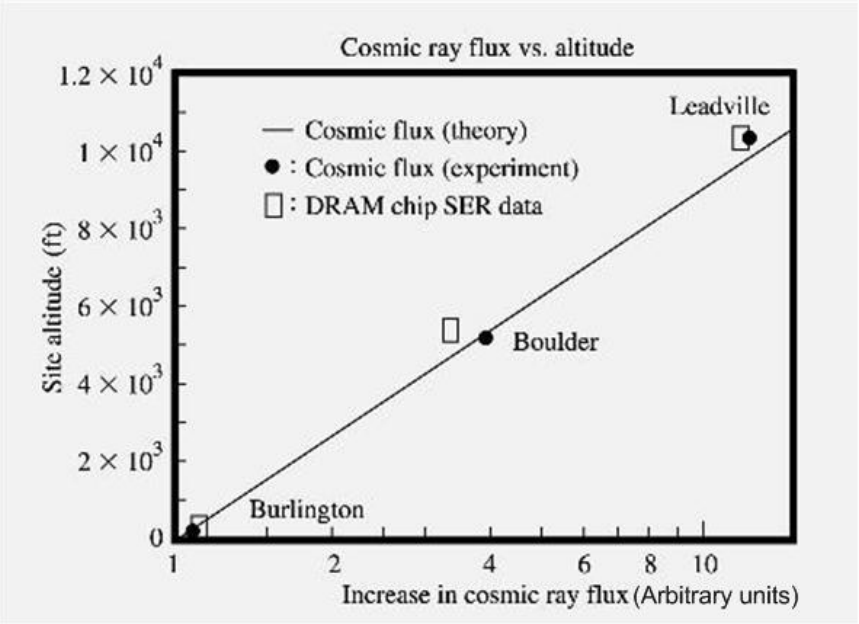


Figure 1.1: Variation of Cosmic Ray Flux with Altitude[72].

1.2.2 Effects of a Particle Hit

Since every node in the circuit holds a certain amount of charge during its operation, any disruption of this charge can cause it to malfunction. Particle hits can cause this disruption in a variety of ways. An α-particle, traveling through the substrate or bulk can create additional electron-hole pairs along the way. On the other hand,a neutron-hit can cause complex nuclear reactions within the substrate. These may in turn give rise to α-particles, neutrons etc.(also known as secondary effect[15]), which can generate additional electron-hole pairs. The noise thus created can propagate to the fanout nodes.

Figure 1.3[72] illustrates the burst of electronic charge when an energetic particle passes through the bulk of a transistor. Cosmic rays at the sea-level have a flux of $1/cm^2$-s and energy high enough to penetrate through simple walls and building ceilings. A fragmenting silicon nucleus as the one shown above would create a noise burst as much as 1M-electrons/μm in silicon[72]. One of the following may take place when a single-event transient is generated on a combinational node of the circuit:

- The noise pulse propagates to the terminal node of that path. This may cause

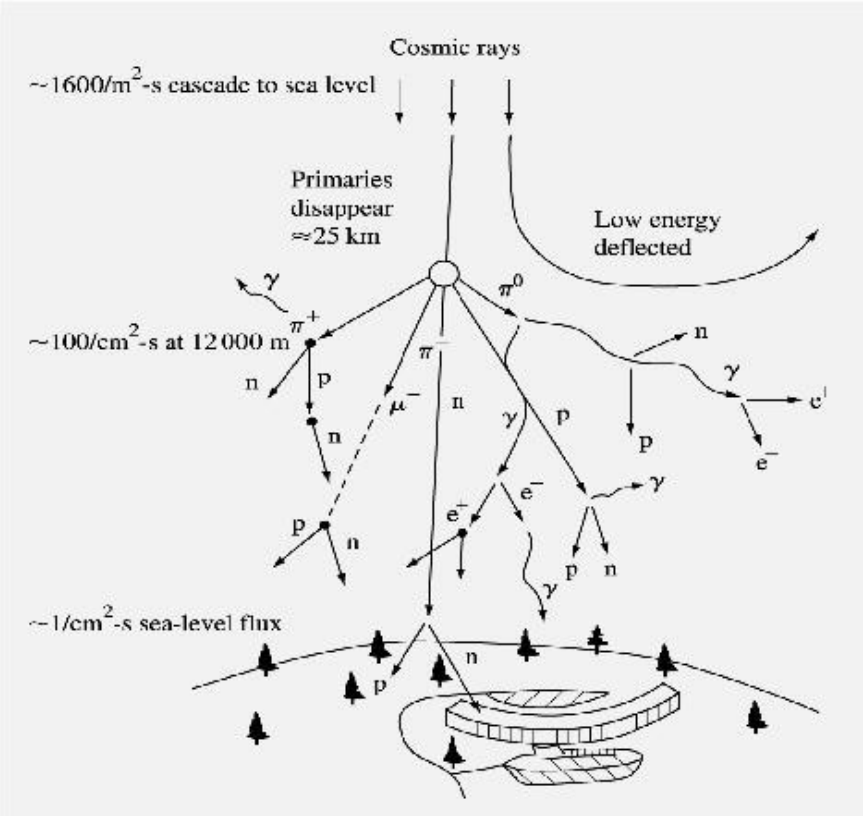


Figure 1.2: Pattern of Cosmic Flux Cascades[72].

a wrong value to be loaded into the latch/flip-flop depending on the arrival instant.

- The pulse may attenuate significantly or be completely masked due to the logical or electrical setup of surrounding nodes [64].

Many factors affect the characteristics of a noise pulse. Depending on the circuit technology and supply voltage, a minimum amount of charge needs to collect at a node before the voltage can disrupt the noise margins of the fanout gates. This magnitude of charge is called *critical charge* (Q_{crit}) [59]. Its value decreases with decreasing feature sizes and supply voltages.

Figure 1.4 illustrates the variation of soft-error rate with respect to chip technology. Soft-error rate is measured in units of *Failure in Time* (FIT). One FIT is equivalent to one failure in 1 billion (10^9) operational hours. The second factor that affects the characteristic of a noise pulse is the *Linear Energy Transfer* (LET) of the

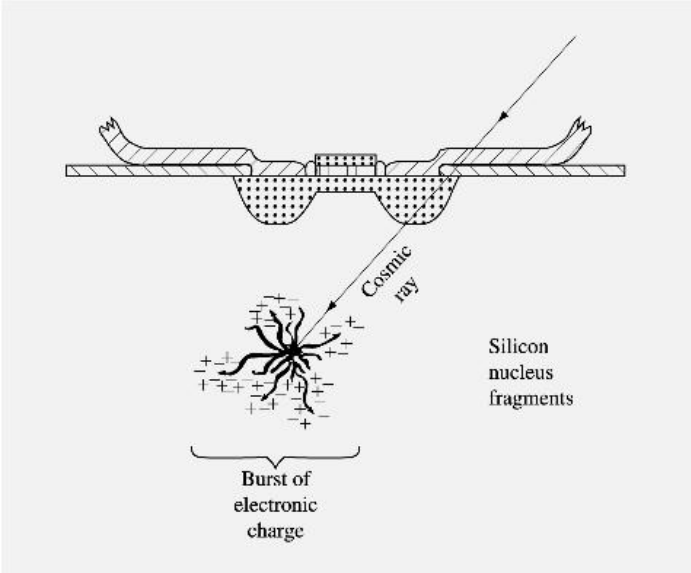


Figure 1.3: Effect of Neutron Hit[72].

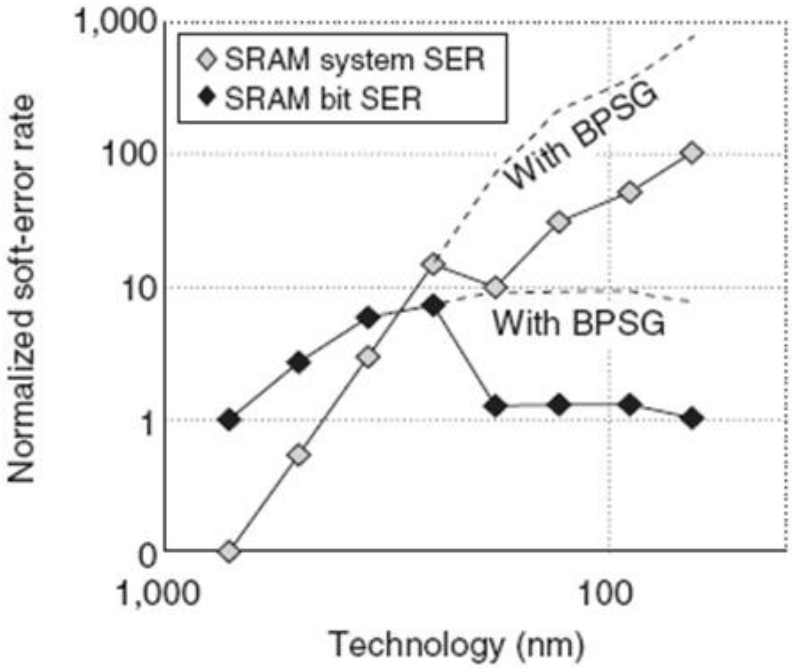


Figure 1.4: Variation of SER with Technology [3].

particle. An energetic charged particle generates electron-hole pairs in the substrate as it travels through it. After losing its entire energy, it comes to rest. The term LET refers to the energy loss of the particle per unit length of its track in the semiconductor. Since cosmic particles have varied LETs, the amount of charge that is deposited at the surrounding nodes varies. The amount of charge collected (Q_{coll}) is given by Equation (1.1) [34]:

$$Q_{coll} = 0.01036 \times L \times t \tag{1.1}$$

where 'L' refers to the LET, 't' is the depth of the collection volume (in microns) and Q_{coll} is in pC. Hence, only when $Q_{coll} > Q_{crit}$[62], a soft-error can be induced. An in depth study of the impact of LET on the characteristics of a noise pulse was carried out by Benedetto *et al.* [5].

1.2.3 Modeling Soft-Errors

Depending on the nature of the circuit surrounding a node, the noise generated due to a particle-hit on it can cause one of the following events to take place

- If the node is part of a combinational path, i.e, there is no regenerative feedback, then a transient pulse is generated, which may propagate through the path. The shape of this pulse changes gradually as it propagates and its peak voltage decreases. If it arrives at the input of a flip-flop or latch at the terminal node of this path during the *Window of Vulnerability* (WoV)[49], it may load incorrect data into it and affect system operation. Such a pulse is called a *Single-Event Transient* (SET) and the resulting error, a *Combinational Soft-Error* (CSE).

- If the node is part of a regenerative circuit, such as the internal node of a latch/flip-flop, this SET may corrupt the state of the memory element. Such a bit flip is called a *Sequential Soft-Error* (SSE).

Consider a latch, as shown in Figure 1.5, that is transparent when the gating/control signal is '1' and opaque when it is '0'. The WoV for this latch is defined as the period of the clock cycle during which it is susceptible to either a CSE or a SSE.

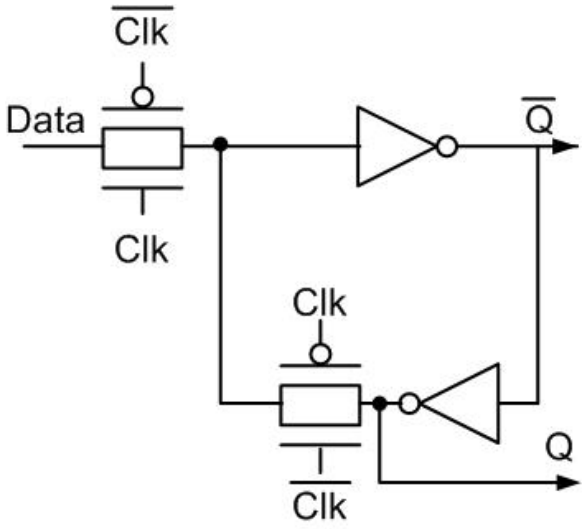


Figure 1.5: Static Inverter Latch.

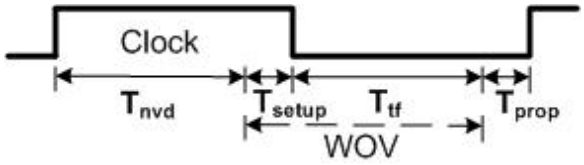


Figure 1.6: Window of Vulnerability for a Latch [49].

In Figure 1.6, 'T_{nvd}' represents that part of the clock cycle during which the fanin gates compute the correct data. Hence, a SET arriving during this period will create only a temporary disruption on the internal nodes of the latch without affecting its final operation. At the end of 'T_{nvd},' required data must be available on the 'Data' pin. 'T_{setup}' is the setup time of the latch and depends on various parameters, such as the rise/fall transition times of the control signal ('Clk'), the threshold voltage of the transistor and the supply voltage. Any noise pulse violating the value on the 'Data' line during this interval may cause incorrect data to be loaded into the latch. Hence, it forms a part of the WoV. The other fraction of time is indicated by the time 'T_{tf}.' A soft-error on an internal node of the latch during this period can flip the state of the latch, affecting subsequent data transmission from it. Together, 'T_{setup}' and 'T_{tf}' comprise the WoV for the latch. 'T_{prop}' is the remaining period of the opaque phase of the latch during which the effect of a particle-hit may not propagate to the output of the latch, thus preventing any erroneous data transmission. Figure 1.7 illustrates

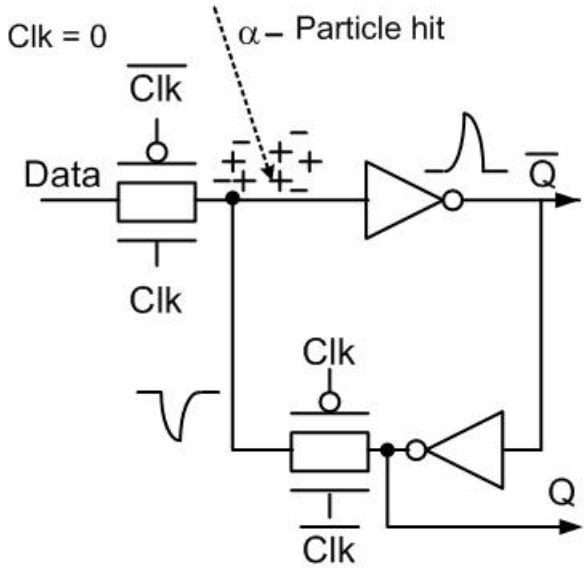


Figure 1.7: Particle Hit Causing an SEU.

the regenerative effect of a latch when a transient pulse strikes an internal node of the latch during the period 'T_{tf}' (from Figure 1.6), when the control signal (CLK) is '0.'

However, under certain conditions, a transient pulse induced in the combinational

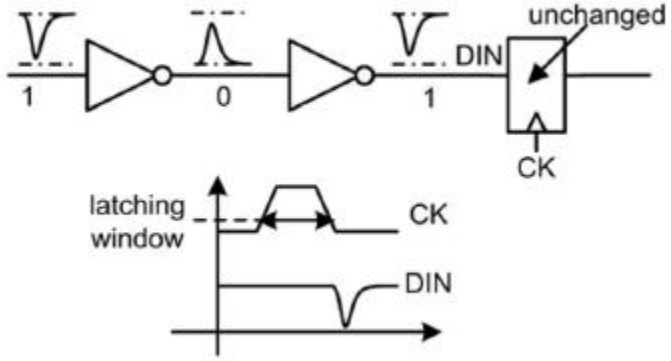


Figure 1.8: Temporal Masking of a SET [27].

logic may not disrupt transmission of correct data through the flip-flop/latch. They are:

- *Temporal masking*: This occurs when a transient pulse generated on the combinational side of the circuit arrives at the data input of the latch/flip-flop outside

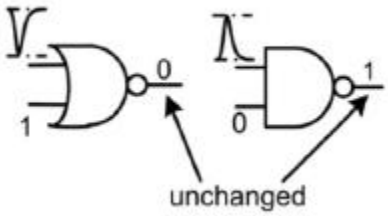


Figure 1.9: Logical Masking of a SET [27].

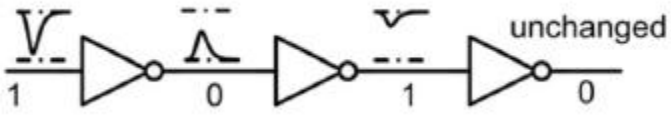


Figure 1.10: Electrical Masking of a SET [27].

the setup/hold time window. As shown in Figure 1.8, this does not affect the value loaded into the flip-flop.

- *Logical masking*: This occurs when the path through which the transient pulse has to propagate is not sensitized. When logic values on other fanins of the corresponding gates are at a controlling value, this type of masking occurs (refer to Figure 1.9).

- *Electrical masking*: The propagation path can be modeled as a circuit with distributed resistances and capacitances. A spike at a certain point along the path may have to propagate through many such passive elements before it reaches the terminal point. As shown in Figure 1.10, the peak voltage decreases gradually as the transient travels along this path. Often, this voltage decreases to the extent that it no longer violates the noise-margins on the receiving end, allowing correct data to be transmitted. This effect is called electrical masking.

1.3 Introduction to Crosstalk

When two or more transmission lines run parallel to each other for long distances and in close proximity, their coupling inductances and capacitances may become

significant relative to the capacitance between the substrate and each individual line. As a result, signal transmission on one line may affect the integrity of signal on the adjacent line. A typical address/data bus in a processor is susceptible to such effects. There are many factors that affect the extent of this disruption. Frequency of data transmission, signal rise/fall times, driver strengths, interconnect spacing and the number of parallel transmission lines are the main ones. The line that induces a noise pulse on its adjacent line is called the *Aggressor* line, while the other one is known as the *Victim* line. Crosstalk may lead to one of the following effects:

- A noise pulse on the victim line due to a transition on the aggressor line.
- A signal transition slowdown on the victim line due to a simultaneous transition in the opposite direction on the aggressor line.
- A signal speedup on the victim line, due to a simultaneous transition in the same direction on the aggressor line.

An elegant study of these effects under various conditions was presented by Breuer *et al.* [60]. Under certain conditions, a signal slowdown may occur on the victim line, even when transitions are in the same direction on the aggressor and victim lines. Figure 1.11 represents a typical setup for a pair of adjacent transmission lines. L_{s1}, L_{s2}, C_{s1} and C_{s2} represent the self-inductances and capacitances of the lines, while C_m and L_m represent the coupling inductance and capacitance, respectively. The values of coupling inductance and capacitance vary with distance between the two lines.

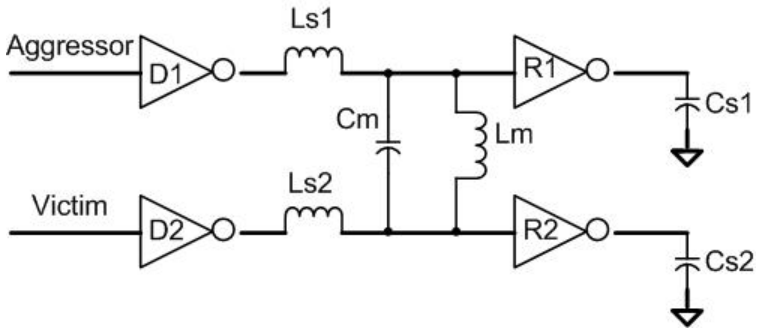


Figure 1.11: A Typical Setup for Transmission lines.

With decreasing feature sizes and increasing operating frequencies, inductive coupling becomes dominant over capacitive coupling. Accurate characterization of noise induced involves use of inductance extraction tools such as *FASTHENRY*, which are computationally intensive and may be commercially unfeasible for larger designs[26]. Current designs incorporate a large number of buses for data transmission. Under such conditions, a victim line may have multiple aggressors affecting it. Characterization of the resultant noise pulse may become more complex. However, it is necessary to ensure correct operation of the device, as such a spurious pulse may affect latching in of the correct data in a latch/flip-flop at the receiving end.

1.4 Roadmap of the Thesis

In Chapter 2, we analyze various techniques proposed so far to overcome crosstalk and soft-errors. Some deal with determining crosstalk prone sites during the design phase, and rectifying them. Some post-silicon techniques exist to weed out unexpected crosstalk prone sites that may have been generated as a result of process variations. Others introduce *Error-Correcting Codes* (ECC) to rectify this problem. We also look at prior work done towards mitigating CSEs/SSEs due to high-energy particles. In Chapter 3, we discuss our proposed solutions, their trade offs with respect to speed, area overhead and design effort. In Chapter 4, comparisons of our work with existing designs in terms of various constraints that affect implementation feasibility are carried out. Chapter 5 concludes this work.

Chapter 2
Prior Work

Researchers have studied the problem of soft-errors since the 1970's when an extensive amount of data about cosmic radiation was collected. Problems with data corruption first started appearing in the static and dynamic memories [6, 17, 36, 44, 53, 57, 71]. The effect of particle hits on the combinational logic elements was insignificant then, due to the relatively large transistor sizes (a larger value of Q_{crit}) and higher supply voltages. Due to a greater density of transistors in the memory, caches were the first to exhibit bit flips and consequently error protection techniques for them were conceived early in the day [18, 19, 28, 51] and today, memories are protected by *Error Correcting Codes* and *Parity Protection Schemes* [22, 58]. Usually, these techniques have the *Single Error Correction-Double Error Detection* (SEC-DED) [7, 50] capability. Although these techniques have their limitations, they have been proven to be effective during real-time operation of the memory and, hence, are used throughout the industry today. However, as feature sizes decreased and supply voltages were lowered to save power, errors due to particle hits cropped up in the combinational circuitry, forcing designers to come up with solutions unique to combinational logic [3]. Although many techniques have been developed to overcome this problem, efforts to improve them in terms of better efficiency and lower timing penalties are being carried out. In this chapter, we present a brief introduction to techniques that have been developed so far and are being widely used in the industry.

2.1 Mitigation/Error-Correction Techniques in Sequential Circuits

Various techniques that would ensure correct operation of a circuit in the presence of radiation have been studied by researchers and these methods can be broadly classified as:

- Selective node hardening aimed at reducing the severity of the noise pulse either by reducing the peak voltage/pulse width or both.
- Using space and time redundancies to detect and correct the effect of the noise pulse, also known as *Concurrent Error Detection Schemes* (CED).

$$SER\ \alpha\ N_{flux} \times A_{diff} \times \exp(-Q_{crit}) \tag{2.1}$$

According to Equation 2.1[59], the *Soft Error Rate* (SER) depends on the cosmic radiation flux, diffusion area and the critical charge. Since critical charge depends on the node capacitance and supply voltage, by suitably sizing the transistors connected to a particular node, optimum values of diffusion area (A_{diff}) and Q_{crit} can be obtained. Increasing the drive strength of a gate can decrease the peak voltage of a transient on the output node[66]. Davis and Eaton have studied various implementations of space and time redundancies and the trade-offs involved [35]. Increasing the delay within a storage cell (latch) can reduce its susceptibility to SEUs. RC Filtering is a technique where feedback resistances are inserted on the connecting nets of cross-coupled inverters[40]. However, this results in generation of more heat and increased write access time. It also makes the circuit harder to scale down. Another manner of hardening the latch involves duplicating the number of output nodes and rewiring the circuit so that a transient pulse affects either the pMOS or nMOS of an inverter gate, but not both. The *Dual Interlocked Storage Cell* (DICE) uses this principle to achieve soft-error resiliency[10]. The *Delay-Filtered-DICE* (DF-DICE) is a modification to the DICE cell that immunizes it to SETs generated in the combinational block[42]. Another method of hardening devices is by introducing *Guard-Gates* [2]. Susceptibility to soft-errors can be reduced by switching to the *Silicon-on-Insulator*

(SOI) process. Research described by Massengill *et al.*, Schwank *et al.* and Oldiges *et al.* describes advantages of this process in terms of reduced CSEs/SSEs and discusses possible modifications to improve reliability of the device[33, 45, 55]. Additional designs described by Metra *et al.*, Patel *et al.* and Drake *et al.* achieve resiliency by introducing additional nodes within the latch and prevent corruption of data in the presence of an SEU[11, 16, 49]. However, there continue to exist certain nodes in these circuits that act as single points of failure. A particle-hit on any one of them can invalidate the corrective operation of the latch.

A second approach to solving this problem involves the use of spatial and temporal redundancies. The concept of spatial redundancies was first introduced by VonNeumann [63]. The primitive idea involves using multiple copies of the data and voting on these copies during real-time operation. *Triple Modular Redundancy* (TMR), which is used today by most of the hardware on board aircraft and space vehicles, maintains three copies of the required data and uses a majority voter circuit to pass the result onto the next stage [39]. An upset in any one of the copies will be voted out this way and correct operation of the system is ensured. However, studies have shown that TMR is effective only when mission times are short and operational frequency is not a significant criteria [40]. The hardware/area overhead is high enough to obviate its use for commercial applications. Temporal redundancy involves sampling the same signal at different instants of time. The decision-making circuitry will detect noise by comparing these multiple samples and voting out any error[43]. Another effective technique involves using parity checkers[52]. This involves using a parity prediction circuit that compares this predicted value with the parity of the final computed data. Any mismatch is considered an error and appropriate action can be taken. Variants of this scheme are used by designers [12, 21, 61, 65]. The simplest is a Single-Bit Parity Checker. Double or *Multiple-Bit Upsets* (MBU) can invalidate this technique. Although more complex schemes can be incorporated to detect MBUs[38], the hardware and timing penalties incurred outweigh the marginal increase in reliability of the circuit. Moreover, synchronization of the parity prediction circuit with the main

hardware is crucial and may involve additional design effort. Other methods involving residue codes exist, however, the main disadvantages of these methods are summarized as follows[32]:

1. Significant hardware/timing penalties
2. Detected but uncorrected errors
3. False positives − generated transients may have no impact on the functioning of the system under certain circumstances. However, these methods may raise a false flag, forcing unnecessary shutdown of the system.
4. At the layout level, routing the additional hardware may involve significant design time and effort.
5. Validating the *Concurrent Error Detection* (CED) system for all input combinations may be cumbersome.

In the next section, we describe a widely used design for the Scan Flip-Flop and briefly describe the modifications that were carried out by Mitra *et al.* to make it resilient to SEUs[40].

2.2 The Basic Scan Flip-Flop

The Scan Flip-Flop can be used in a typical master/slave configuration during the functional mode of operation of the system. During the testing phase of the chip (stuck-at fault, transition, path-delay, etc.), test vectors can be scanned into these flops, which are configured as a giant shift-register[9].

These vectors are applied to the combinational circuit from the output 'Q' of the flip-flop. A vector is usually applied for one clock cycle of operation (the circuit is switched back to functional mode during this period) and the response is captured and scanned out by switching back to the scan mode. Mismatches between expected and observed responses can be used to detect and diagnose various types of faults in the circuit. In Figure 2.1, latches *PH2* and *PH1* form the functional part of the

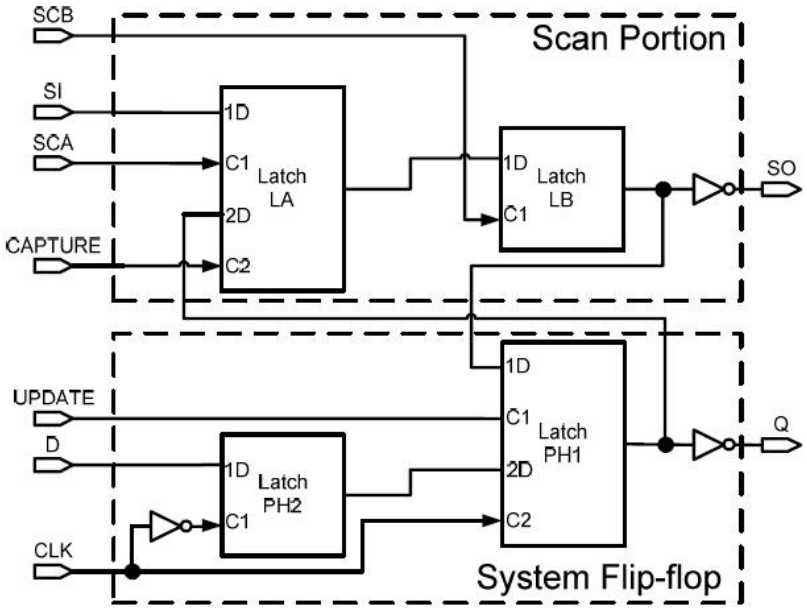


Figure 2.1: Basic Scan Flip-Flop[41].

flip-flop. During the functional mode, PH2 transmits data to PH1 at the positive clock edge[41]. Latches 'LA' and 'LB' are used during the scan mode of operation. 'SI' is the data pin of LA and is connected to the *Scan-Out* (SO) of another scan flip-flop. Signals 'SCA' and 'SCB' act as the control/gating signals for LA and LB. During scan mode, LA/LB operate in a master/slave configuration. The 'UPDATE' signal, activated only during the scan mode, loads the vector from LB into PH1. The entire module is clocked for one cycle and the response is captured in the receiving flip-flops by activating the 'CAPTURE' signal. The response is then scanned out by toggling the signals SCA and SCB. This design of the basic scan flip-flop has been modified by Mitra *et al.* to include a C-element, which can detect errors[41]. A keeper on the output node of the flip-flop loads the correct data just after the active edge of clock has arrived. Any SEU occurring after this instant will force the C-element into a high-impedance state, while the keeper feeds the correct data to other gates. The *Built-In Soft Error Resilience* (BISER) flip-flop, although simple and efficient, has its limitations. First, it does not cover the entire *Window of Vulnerability* (WoV)[49]. The WoV for a latch is defined as the time interval during a clock cycle that a latch is susceptible to either a CSE or a SSE. An incoming SET from the combinational side

may load an incorrect value (due to a setup/hold-time violation) into PH2, which then is passed on to the output of this flip-flop, invalidating its operation. Soft-error tolerant designs based on that of scan flip-flops have been proposed by Roy *et al.*, Dasgupta *et al.* and Krishnamohan *et al.* [14, 20, 30]. However, the designs cited so far have limited tolerance, that is, they are vulnerable either to incoming transients or internal node upsets. They are resilient to only a portion of the WoV. Although certain techniques may be resilient to both of the effects, they involve high area/timing penalty and may be commercially infeasible for consumer electronics[4]. To increase the resiliency of the BSFF, we propose combined use of space and time redundancies with minimal area overhead and tolerable timing penalties. However, we first demonstrate the vulnerability of the BSFF to CSEs/SSEs.

2.2.1 Detection and Mitigation of Crosstalk Faults

Much work dealing with detecting fault sites prone to crosstalk error has been done in the past. Chen *et al.* proposed a mixed-signal test generation scheme that uses PODEM-like static values as well as dynamic values to generate tests for crosstalk faults[13]. The most favored approach has been to generate vectors that excite the worst-case delay at crosstalk prone sites. However, as circuits increase in size and density, it is observed that the efficiency of such algorithms decreases [13]. This is because a larger primary input space needs to be searched[13]. Also, a higher correlation between gate inputs, which might exist due to a large number of reconvergent fan-outs, may leave many of the faults undetectable. Another drawback of these algorithms is the fact that they aim at detecting only capacitive coupling faults [54]. This is not a precise measure of crosstalk effect, as inductive crosstalk is now a major part of the coupling [23]. Work on developing novel algorithms has been carried out and methodologies have been developed to evaluate the defect-coverage of these algorithms. Work along this line of study has been carried out [1, 70].

Another method of dealing with the problem of crosstalk was to budget delay when routing the hardware at the layout level[68]. This however, involves much time

and effort during the design phase. For crosstalk estimation, accurate extraction of interconnect inductance is very important. Extraction of inductance is a complex task and using electromagnetic field solvers such as *FASTHENRY* [26] would involve much time in the design phase. The significance of inductive interconnect crosstalk demands a different approach to overcoming its effects. This approach should aim at developing hardware that detects it and recovers from the spurious pulse or delayed signal accordingly. Work targeted at detection of noise pulses, generated on interconnects, was first presented by Metra *et al.*[37]. Similar work was presented by Yi *et al.* [69]. Both of these methods aimed at detecting the occurrence of a crosstalk noise pulse. However, a correction mechanism for such lines was not considered. This work also did not address the issue of a signal delay being caused due to the presence of a crosstalk fault. Lajolo proposed an architecture involving dedicated circuitry to monitor interconnects and targeted at detecting and correcting such faulty data[31]. The success of this implementation is limited by its single-error handling capability.

Oliveira *et al.* proposed designs tolerant to only SSEs [47, 48]. Designs tolerant to both CSEs and SSEs have been proposed by Oliveira[46], but they have relatively higher hardware overhead as compared to designs proposed in this thesis[24, 25]. The main difference between designs by Oliveira and designs proposed here lies in the trade off between design effort and design area overhead. Designs proposed here need additional precise synchronous signals, which may be hard to generate at very high clock speeds. Designs by Oliveira use signals derived by simple linear delays of the system clock. In the next chapter, we propose various modifications to the BSFF that immunize it from both CSEs/SSEs over the entire Window of Vulnerability.

Chapter 3

Proposed Flip-Flop Designs to Immunize Circuits to SEU/SETs

In the previous chapter, existing techniques to mitigate SEU/SETs were discussed. In this chapter, we propose circuit-level error-detection and correction techniques that make optimal use of spatial and temporal redundancies. We target scan flip-flops at the end of every pipeline stage, as an SEU/SET will need to affect their state to cause the circuit to malfunction. Also, we need to immunize them for the entire WoV for every latch in the BSFF. Our aim also includes recovery from minor signal delays, that is, desired transitions missing the hold-time of the flip-flop. Severe signal delays arising due to process variations are not the target of our recovery scheme. Our initial designs are immune only to combinational SETs and crosstalk induced effects. Later designs were made immune to both combinational and sequential soft-errors[24, 25].

3.1 The Crosstalk Tolerant Flip-Flop (XTFF)

The XTFF can handle two types of crosstalk induced effects. The first one is a noise pulse. A noise pulse is transient in nature and can cause errors when its peak voltage (or maximum undershoot for a logic high data) exceeds half the supply voltage ($V_{DD}/2$). Sample noise waveforms are shown in Figure 3.1. A general purpose scan flip flop is susceptible to noise pulses because it observes the incoming data only at the instants, *Sample1* (setup time) and *Hold* time. The logic value during the hold time is more important in controlling the value latched into the master latch (PH2). For a correct value to be propagated or latched into PH2, it must remain stable during this interval. Thus, there arise two situations in which a hold-time violation may occur.

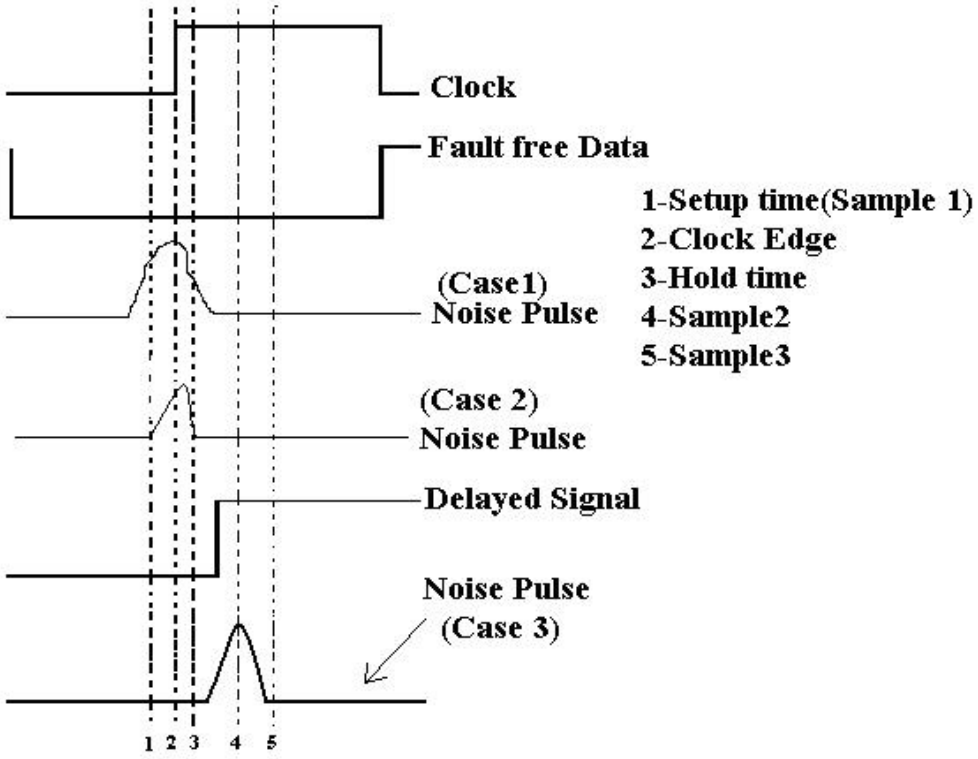


Figure 3.1: Setup/Hold-time Violations Occuring at the Input of a Flip-Flop.

The first case occurs when a noise pulse is induced before the setup time and dies away after the hold time has elapsed. The second case occurs when a noise pulse has a steeper rise time. It occurs after the setup time and remains high during the hold time. Both of the cases have been illustrated in Figure 3.1. The logic values received by the flip-flop for these two cases, during the setup time and hold time are shown in Table 3.1. Case 1 causes a '1' to be latched into PH2 (Figure 3.1), however, Case 2 may latch in either a '0' or a '1.' This condition needs to be handled by the XTFF as an 'X' (*don't care*) value being latched into PH2 at the end of its transparent phase. We introduce a second sampling interval, *Sample2*, to detect this latching of incorrect data. This signal remains active for a small period beyond the hold time of the flip-flop. As we can see in Figure 3.1, the signal on the data line falls back to '0' (correct value) by the end of the instant, *Sample2*. Therefore, monitoring the data line for this interval of time, will allow detection of a noise pulse. Table 3.1 shows the values of different types of signals at the various sampling intervals. *Sample2* is a

Table 3.1: Data Values at Various Sampling Instants in Figure 3.1.

Type of Data	*Setup Time*	*Hold Time*	*Sample2*	*Sample3*	Value in BSFF	Value in XTFF
Fault-Free	0	0	0	0	0 (Correct)	0
Noise-Pulse (Case 1)	1	1	0	0	1 (Wrong)	0
Noise-Pulse (Case 2)	0	1	0	0	X (Wrong)	0
Delayed Data	0	0	1	1	0 (Wrong)	1
Noise-Pulse (Case 3)	0	0	1	0	0 (Correct)	0

synchronous signal, which will be fed to the XTFF. It helps the XTFF distinguish a noise pulse, occurring around the active clock edge, from a good signal by using the values of *Sample1*, *Hold-time* and *Sample2* as shown in Table 3.1.

The second crosstalk induced effect manifests itself in the form of excessive delay in the signal arrival time or transition time. Signal delay calculations involve various inductive and capacitive coupling coefficients. However, a delay greater than the period of a clock cycle can be regarded as a severe defect and usually involves not only crosstalk induced effects but also other defects that may have occurred due to traditional delay faults. We consider signal delays that occur within the same clock cycle. Figure 3.1 illustrates a signal that misses the active (rising) clock edge by a small interval of time. This is a delayed signal. The values at the instants, *Sample1* and *Hold-time* are given by Table 3.1. The value at *Sample2* is a '1' (correct value), however, from Figure 3.1 and Table 3.1 we can observe that *Sample2* can be a '1' for two types of signals. The first is a delayed signal while the other is a noise pulse (Case 3, Figure 3.1). Hence, if the first edge on the data line occurs during the active period of *Sample2* and then, dies away after *Sample2* has become inactive, this type of noise pulse (Figure 3.1, Case 3) will be confused with a delayed signal. Therefore, relying on only one sampling signal may lead to inaccurate analysis of a delayed signal. In order to avoid this erroneous interpretation of the data, we introduce another signal, *Sample3*. This signal is activated at the falling edge of *Sample2*. *Sample3* has a

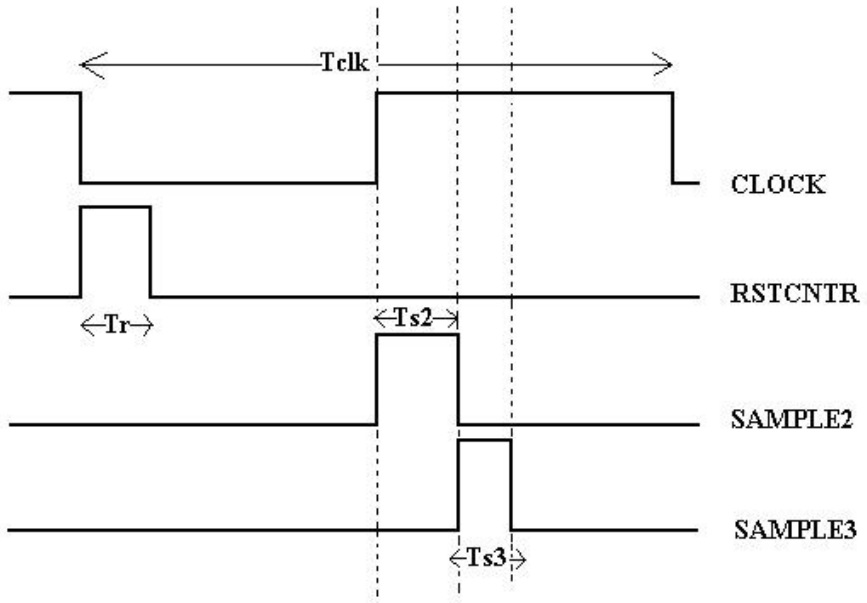


Figure 3.2: Timing of Synchronous Signals Relative to System Clock.

pulse width smaller than that of *Sample2*. From Table 3.1 we can observe that the logic values for *Sample3* are different for the delayed signal and noise pulse (Case 3). Therefore, observing and comparing the values at *Sample2* and *Sample3* will help us distinguish a delayed signal from a noise pulse, as shown in Case 3. Thus, based on the patterns shown in Table 3.1, the XTFF decides the nature of an incoming signal.

The XTFF relies on a counter to record arrival of the edges on the data line. It can be seen that latch *LB* is not involved during the functional mode of operation in a generic scan flip flop (Figure 2.1). In the XTFF, this latch is converted into a 1-bit counter. The state of this counter gets complemented whenever an edge arrives on the data line. This counter needs to be initialized to a '0' at the beginning of every clock cycle. This is to prevent the state of the counter in the current clock cycle being affected by its state from a previous clock cycle. A signal, *RSTCNTR*, is generated to carry out this initialization operation at the beginning of every clock cycle. Successful operation of the XTFF depends on reliable generation of the signals, *RSTCNTR*, *Sample2* and *Sample3*. Their waveforms relative to that of the system

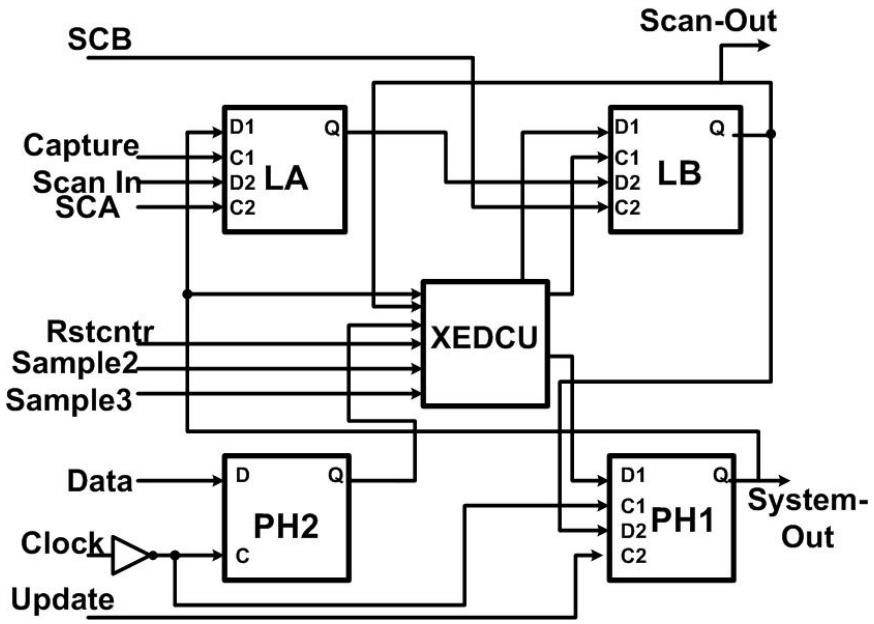


Figure 3.3: Schematic of the XTFF.

clock are shown in Figure 3.2.

We now discuss the general structure of the XTFF. Figure 3.3 gives an architectural view of the XTFF. It consists of the primitive four latches: PH2, PH1, LA and LB that are also present in a scan flip-flop (Figure 2.1). In addition to these, a *Crosstalk Error Detection and Correction Unit* (XEDCU) is employed. The XTFF can operate in two different modes. It can switch between the functional mode of operation that employs its concurrent fault-recovery scheme, and the scan mode of operation, which helps carry out stuck-fault testing of the internal hardware on the chip. PH2 and PH1 are the system master and system slave latches, while the latches LA and LB are the scan master and scan slave. The state of the *TESTCONTROL* signal, determines the mode of operation ('0' for functional mode, '1' for scan mode). The signals, $UPDATE$, $CAPTURE$, SI, SCA, and SCB in Figure 3.3 have the same functionality as in a general purpose scan flip flop (Figure 2.1). The XEDCU splits its fault-recovery operation into three steps:

- Detecting a transition on the data line.
- Filtering an incoming SET.

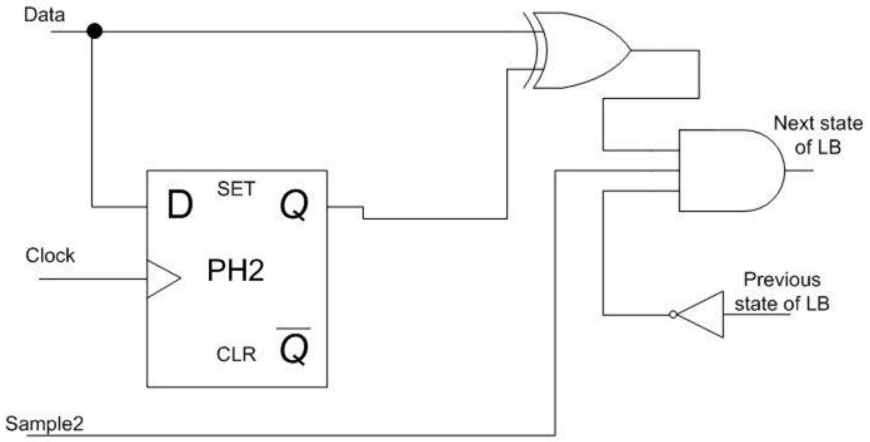


Figure 3.4: Edge Detection Circuit within the XEDCU.

- Recovering data from a delayed signal.

The corrected value is then latched into PH1. These three parts are briefly described below. In order to detect the occurrence of an edge on the data line, a comparator (XOR gate) is used. An incoming data pulse is first latched into PH2 during its transparent cycle. Once PH2 becomes opaque, the value on the output node is compared with the signal on the data line. The circuit used for this operation is shown in Figure 3.4. For every edge that occurs on the data line after the arrival of the clock edge, the state of latch LB is complemented. This process continues only until *Sample2* remains '1' (Figure 3.1). The second part of the XEDCU detects noise and latches in the corrected data into the system slave latch (PH1). Figure 3.5 shows the circuitry

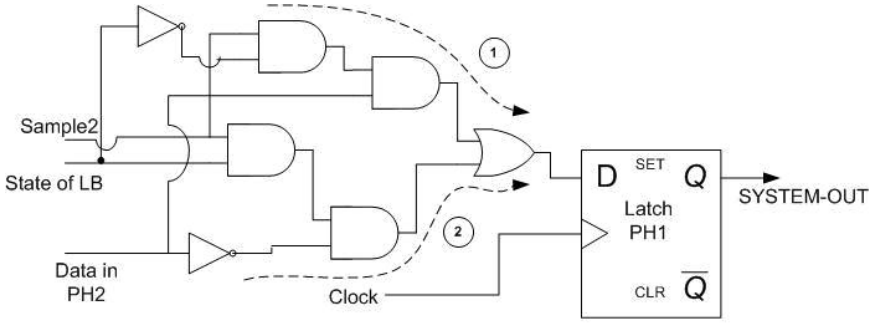


Figure 3.5: Detection and Correction Circuitry within XEDCU.

involved during this phase of operation. In this case, if a noise pulse arrives on the data line (Figure 3.1), path 2 is selected when transmitting data from PH2 into PH1.

The pattern shown in Table 3.1 is used for this operation. The circuit complements the contents of PH2 before latching it into PH1. This is because the value latched into PH2 was an erroneous one. For the case shown in Figure 3.1, a '1' (wrong value) would latch into PH2, but a '0' (corrected value) would be latched into PH1. If no edge occurs, then path 1 is followed when transmitting data from PH2 (correct value in this case) into PH1.

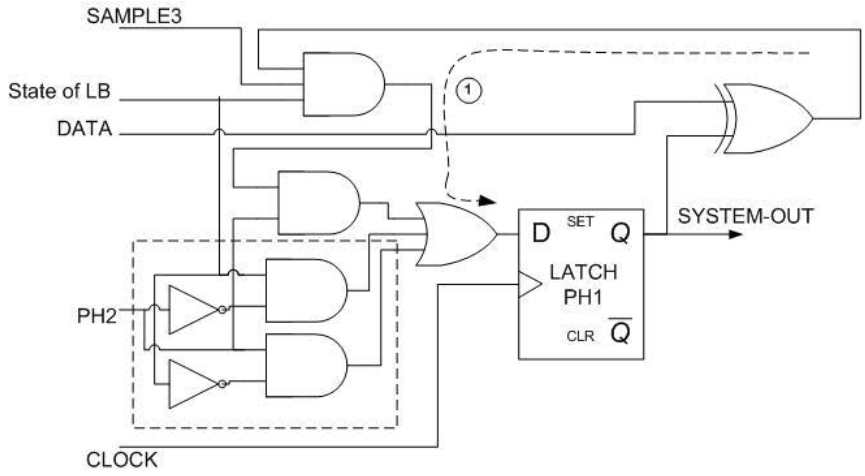


Figure 3.6: Delayed Signal Recovery Circuit within the XEDCU.

The third part of XEDCU recovers data from a delayed pulse. The data recovery procedure is described in the earlier part of this section. Signal values on the data line at *Sample2* and *Sample3* are both used in determining the final correct value that must be latched into PH1 (Table 3.1). Figure 3.6 illustrates the circuitry involved in this process. In Figure 3.6, the part of the circuit enclosed by the dotted rectangle indicates the part activated during the previous phase when *Sample2* is active. When *Sample3* is made active (high), distinction between a noise pulse (Case 3, Figure 3.1) and a delayed signal is carried out. In case the pulse was as shown in Figure 3.1, Case 3, path 1 is activated that latches the correct value stored in latch PH2. The XEDCU consists of circuits illustrated in Figure 3.4, Figure 3.5 and Figure 3.6 integrated into a single unit. Digital simulations of the XTFF were carried out using the VCS simulator provided by Synopsys. Figure 3.7 indicates the results of these simulations. Here, (1) indicates arrival of a noise pulse on the data line and (2) indicates delay in signal arrival time relative to the active (rising) clock edge. It is observed that a corrected

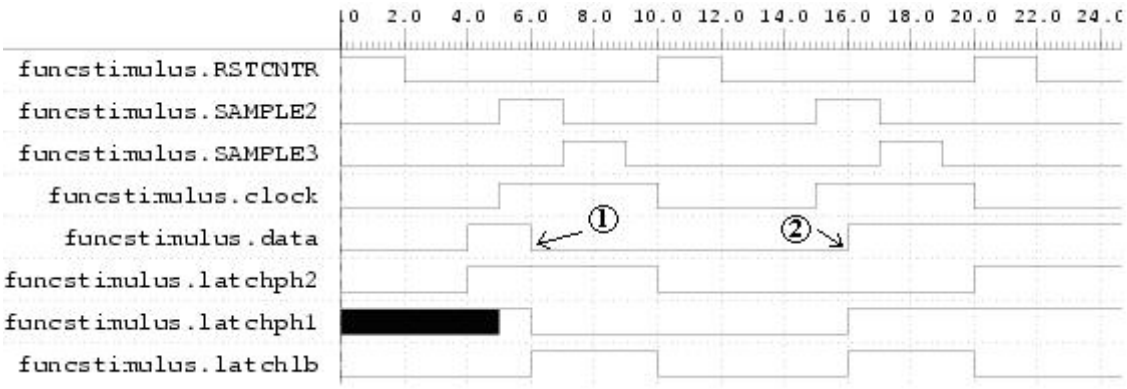


Figure 3.7: Result of Logic Level Simulation of the XTFF.

value gets latched into PH1, for both cases (1) and (2) even though an incorrect value is stored in PH2.

The flow of logic inside the XEDCU can be described comprehensively, through the flowchart illustrated in Figure 3.8. The XTFF was implemented using transistor models from the TSMC 0.18 μ m library. Switch level simulations were carried out using the Spectre tool (from Cadence, Inc.) and verified. Successful operation and verification of the XTFF requires observing the constraints mentioned in Equations 3.1, 3.2 and 3.3. Routing delays vary depending on the layout. These delays need to be considered when implementing the XTFF at the layout level.

$$T_{s2} > T_{s3} > T_{xpulse} \tag{3.1}$$

$$T_{clk/2} > T_{s2} + T_{s3} + T_{xedcu} \tag{3.2}$$

$$T_{maxdelay} < T_{s2} - T_{xedcu} \tag{3.3}$$

T_{s2}, T_{s3} and T_{xpulse} refer to the widths of $Sample2$, $Sample3$ and the noise pulse, respectively. $T_{clk/2}$ refers to half the clock period while T_{xedcu} refers to the delay of the decision making circuitry. The maximum tolerable delay of the XTFF is indicated by $T_{maxdelay}$.

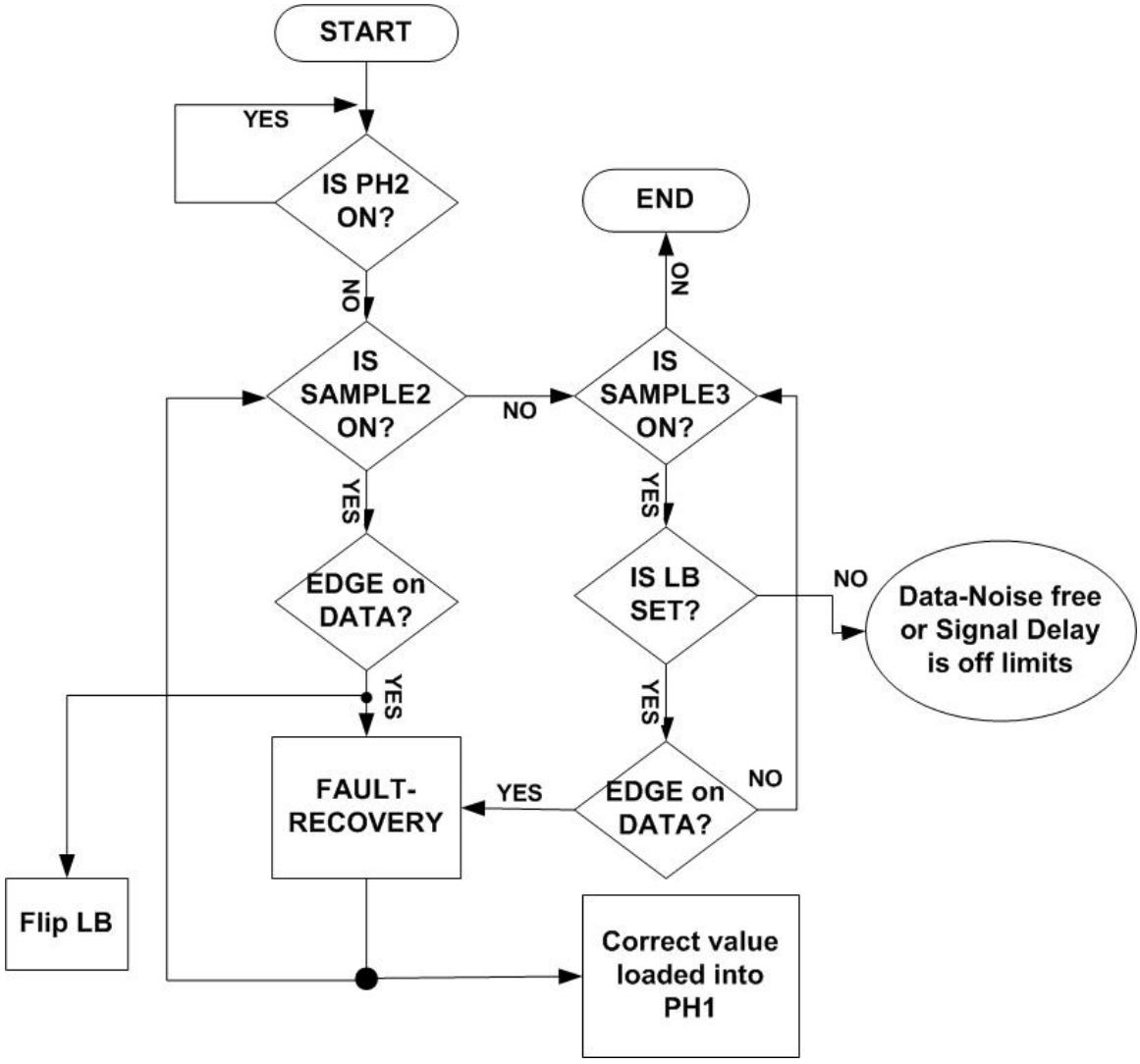


Figure 3.8: Flowchart Indicating Sequence of Decisions Taken by the XTFF.

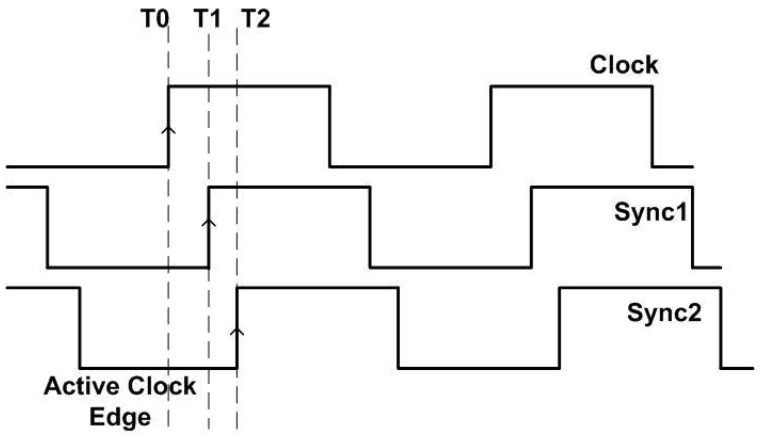


Figure 3.9: Timing of Synchronous Signals Relative to System Clock.

3.2 The Crosstalk Tolerant Flip-Flop-2 (XTFF2)

In this section, we propose a second flip-flop design tolerant to SETs. The XTFF2 is a significant improvement over the XTFF. It is capable of recovering from erroneous latching of data caused due to noise pulses and signal delays. It reuses the scan portion of the basic scan flip-flop to create redundant samples of the incoming data. This is achieved through temporal sampling of the data line at multiple instants of time, a technique explained by Mavis *et al.* [35]. These instants are chosen such that a typical noise pulse does not overlap any two sampling instants. Also, a limit must be set on the amount of crosstalk induced delay that the XTFF2 needs to tolerate. Since these multiple samples need to be latched in at different instants of time, two additional, synchronous signals need to be generated. These signals can be generated by delaying the clock pulse through static buffers. Their timing diagrams with respect to the system clock are shown in Figure 3.9. Signals *Sync1* and *Sync2* are gating signals for latches LA and LB respectively (during the functional mode of operation). Here, we consider LA, LB and PH2 to be transparent when *Clock* is '0' and opaque (or in the store mode) when it is '1.' Latch 'PH1' is considered to be transparent when *Clock* is '1' and opaque when it is '0.' From Figure 3.9, it can be observed that *Sync1* and *Sync2* latch redundant values into LA and LB at instants 'T1' and 'T2,' respectively. Time instants T1 and T2 can be suitably fixed by sizing the buffers appropriately in the signal generator circuit, shown in

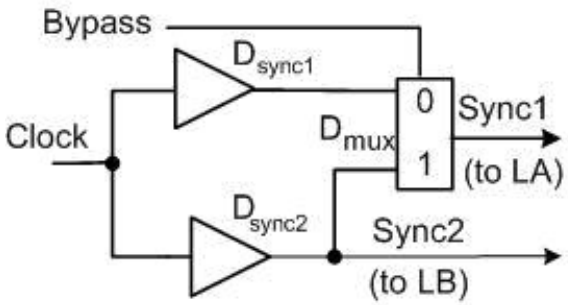


Figure 3.10: Signal Generator Circuit.

Figure 3.10. The values D_{sync1}, D_{sync2} and D_{mux} represent propagation delays of the corresponding buffers and multiplexer, respectively. The addition of the multiplexer allows the designer to resolve issues with race-conditions during the debugging phase of the design. During regular operation, the *Bypass* signal is '0.' When the design must be analyzed for occurrence of any hold-time failures, the *Bypass* signal is set to '1.' This allows latching of the same faulty data (data arriving too soon in the case of a hold-time failure) simultaneously in LA and LB (*Sync1*, *Sync2* now have similar waveforms), thus forcing the majority voter to load this incorrect data into PH1. This faulty value can then be loaded into LA and scanned out. During regular operation

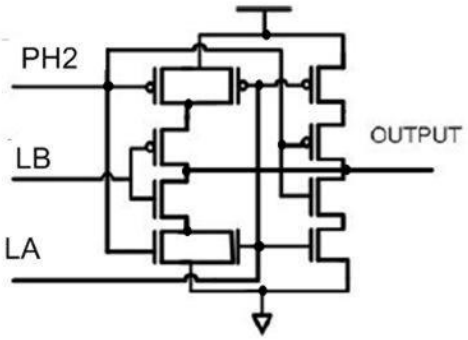


Figure 3.11: Schematic of a 3-Input Majority Voter.

of the XTFF2 ($Bypass = 0$), once the values taken at multiple time instants are available in the corresponding latches, they can be fed to a majority voter (Figure 3.11) that resolves any errors and loads the corrected value into the system slave, latch PH1. The truth table for this operation is shown in Figure 3.2. This bit is then

Table 3.2: Operation of the Majority Voter in Figure 3.11.

Value in PH2	Value in LA	Value LB	Voter Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

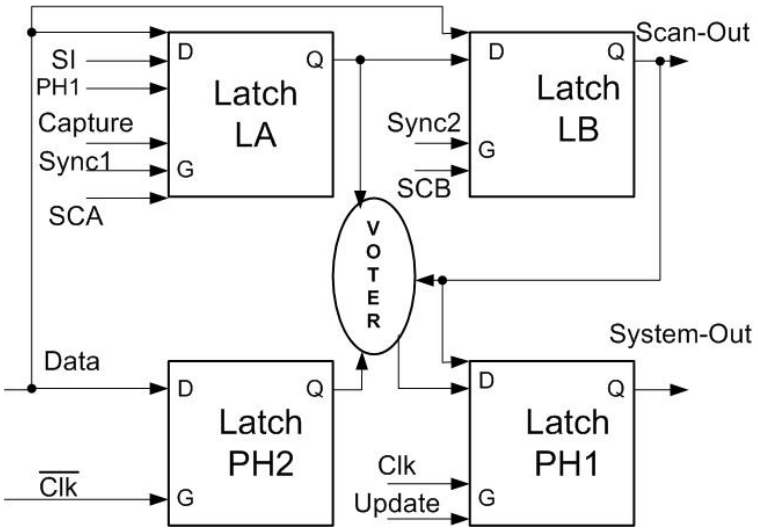


Figure 3.12: Schematic of XTFF2.

propagated to the inputs of the combinational block of the next stage through PH1.

The structure of the XTFF2 is illustrated in Figure 3.12. Latches LA and LB can be used for shifting data when operating in scan mode. Hence, the XTFF2 can be used as a scan flip-flop in addition to being used as an error detecting and correcting flop. Successful operation of the XTFF2 requires implementation of certain constraints on the timing of signals *Sync1* and *Sync2*. The setup and hold times of these latches, in conjunction with delays of multiplexers that feed the data inputs to them, need to be considered when determining these constraints. If $T_{setup(PH2)}$, $T_{setup(LA)}$ and $T_{setup(LB)}$ represent setup times for latches PH2, LA and LB and $T_{cd(LA)}$ and $T_{cd(LB)}$ represent

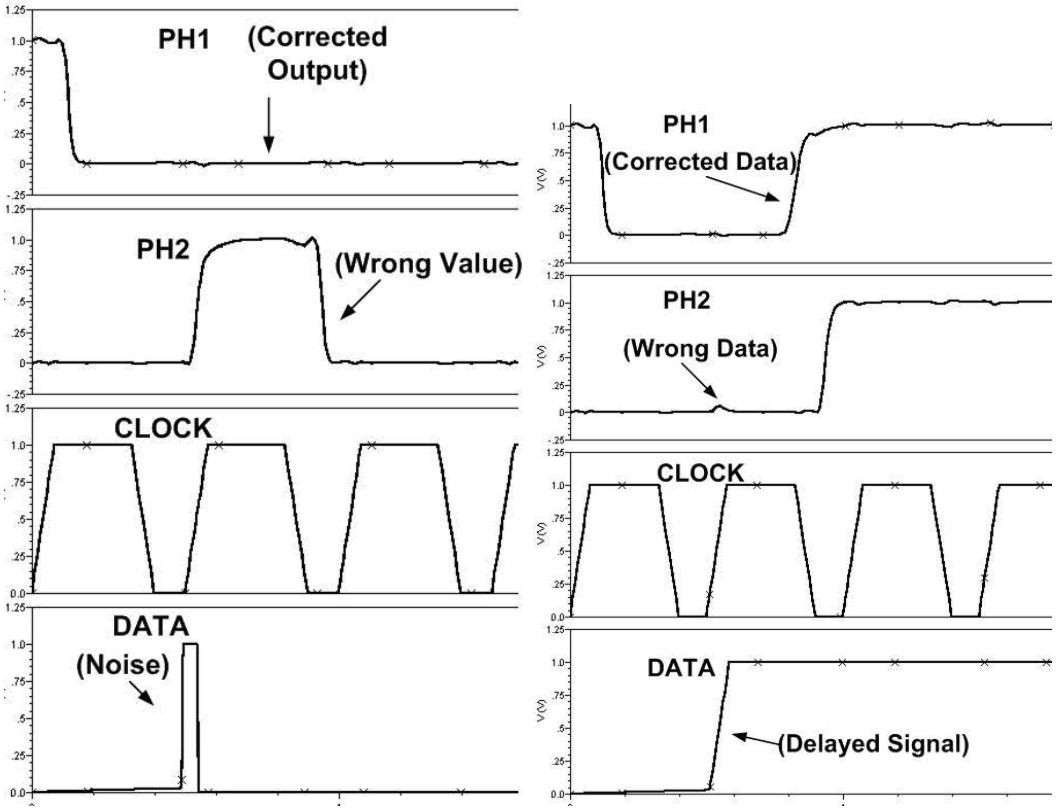
delays of multiplexers that lead to the data input of latches LA and LB, respectively, then the maximum width of a noise pulse (T_{Xpulse}) that can be detected and corrected by the XTFF2 (worst-case scenario) during the clock cycle can be given by Equation 3.4.

$$\begin{aligned} T_{Xpulse} &\leq [T2 - (T_{setup(LB)} + max(T_{cd(LB,R)}, T_{cd(LB,F)}) \\ &\quad -(T1 - (T_{setup(LA)} + max(T_{cd(LA,R)}, T_{cd(LA,F)})))] \qquad (3.4) \\ \Delta_{max} &\leq [T1 - (T_{setup(LA)} + max(T_{cd(LA,R)}, T_{cd(LA,F)}))] \\ &\quad -[T0 - T_{setup(PH2)}] \qquad (3.5) \end{aligned}$$

Here, $T_{cd(LA,R)}$, $T_{cd(LB,R)}$ and $T_{cd(LA,F)}$, $T_{cd(LB,F)}$ refer to the combinational delays for rising and falling transitions at the inputs of latch LA and LB, respectively. Time instants T1 and T2 are shown in Figure 3.9. Hence, from the equation we can observe that by controlling these time instants we can vary the tolerance of the flip-flop. If Δ_{max} is the desired value for maximum tolerable delay (beyond the setup time of latch PH2) of the XTFF2, then Equation 3.5 describes the constraints that need to be observed. Waveforms below illustrate the performance of the XTFF2 under various fault conditions. A noise pulse of 50ps duration and a peak voltage of 1 Volt was allowed to corrupt the data in the master latch (PH2) of the XTFF2. The output of the slave latch, latch PH1, was observed after the arrival of the active clock edge. Figure 3.13(a) illustrates the handling of a noise pulse by the XTFF2, while Figure 3.13(b) illustrates the effect of a delayed signal on the flop.

3.3 The Crosstalk and SEU Tolerant Flip-Flop (XSEUFF2)

Designs described so far are immune only to SETs and crosstalk effects. However, a particle hit on an internal node of a latch in the *Basic Scan Flip-Flop* (BSFF) may result in an SEU disrupting its operation. The designs XTFF and XTFF2 are not capable of handling such an effect. In order to overcome this shortcoming, a more tolerant design is proposed. The *XSEUFF2* provides immunity to the circuit from the following:



(a) Noise (b) Delay

Figure 3.13: Response of XTFF2 to SETs and Crosstalk Delay.

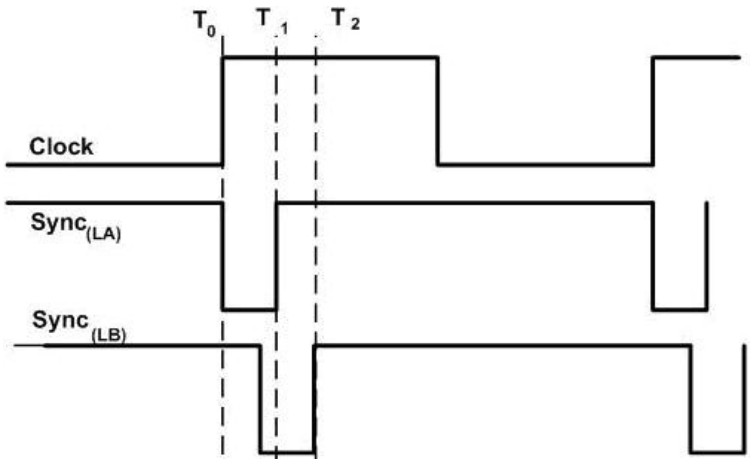


Figure 3.14: Timing of $Sync_{LA}$ and $Sync_{LB}$ Relative to System Clock.

1. SEUs occuring due to particle strikes on the internal nodes of latches.

2. Incoming noise pulses irrespective of their original source.

3. Signal Delays arising as a result of interconnect crosstalk.

3.3.1 Principle of Operation

Consider the design of a BSFF (Figure 2.1) and the WoV of a latch (Figure 1.6). Protection against an SET/Noise pulse requires that temporal sampling be carried out around the active clock edge (in this case, the *rising edge*). This can be achieved by generating additional sampling signals. These additional values can be stored in the scan portion of the BSFF. Correction can now be carried out by using a majority voter and this value then passed on to the slave latch. During this half of the clock cycle (i.e., when the slave is transparent), the flip-flop output is driven by the majority voter. Once the next part of the cycle begins, the master latch goes transparent and the slave becomes opaque. During this period, the latches in the scan portion can vote in conjunction with the system-slave latch to prevent disruption of the flip-flop output due to an SEU.

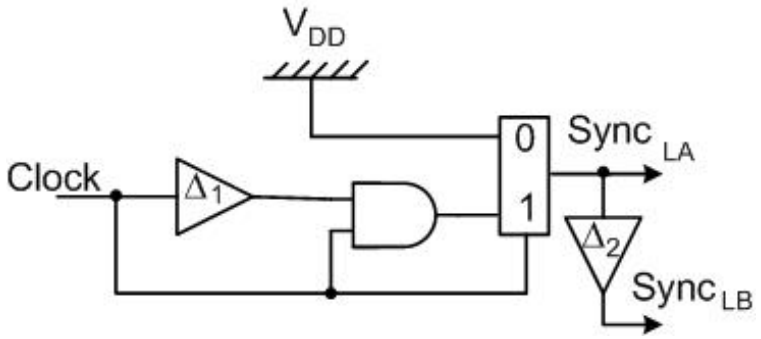


Figure 3.15: Schematic of Signal Generator.

3.3.2 Architecture of the XSEUFF2

Latches LA and LB are gated by the signals $Sync_{LA}$ and $Sync_{LB}$, respectively, during regular operation of the flip-flop. Figure 3.14 illustrates the timing of these signals relative to the system clock. The circuit generator for these signals is shown in Figure 3.15. Delay Δ_1 determines the width of $Sync_{LA}$ and Δ_2 determines the skew between $Sync_{LA}$ and $Sync_{LB}$. One such circuit can be used to generate signals for an entire bank of flip-flops thus reducing the overall transistor/area overhead of this scheme.

Figure 3.16 illustrates the schematic of the XSEUFF2. The temporal sampling required at the beginning of the WoV is carried out by loading multiple samples in latches PH1, LA and LB. The majority voter, which is embedded into a single structure with latch PH1, votes on the samples latched in and sends the correct output to the *System-Out* pin. PH2 continues to act as an input to the voter until it turns transparent. During the next half cycle, when PH1 turns opaque and PH2 goes transparent, values stored in LA, LB and PH1 are fed to the majority voter. This continuous voting is carried out by the scan latches and the system slave and helps ensure the integrity of data during the remainder of the WoV. Thus, alternate use of time and space redundancies can be made to make the design completely immune to soft-errors.

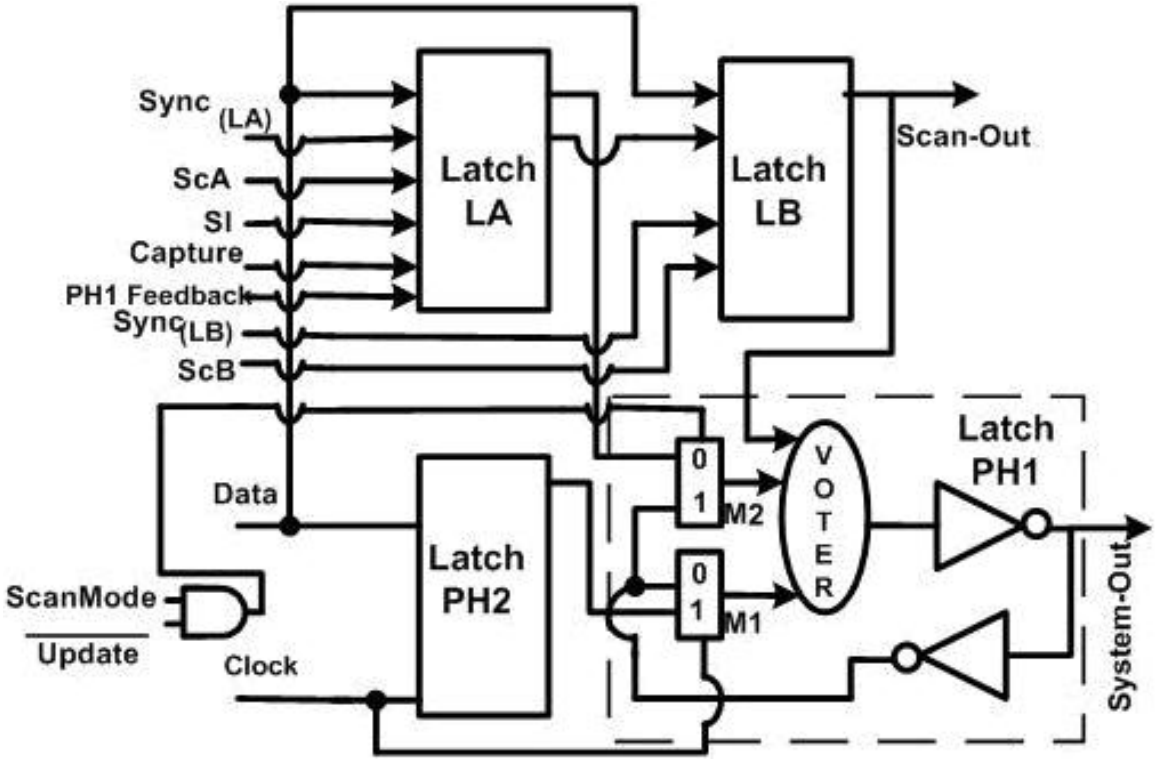


Figure 3.16: Schematic of the XSEUFF2.

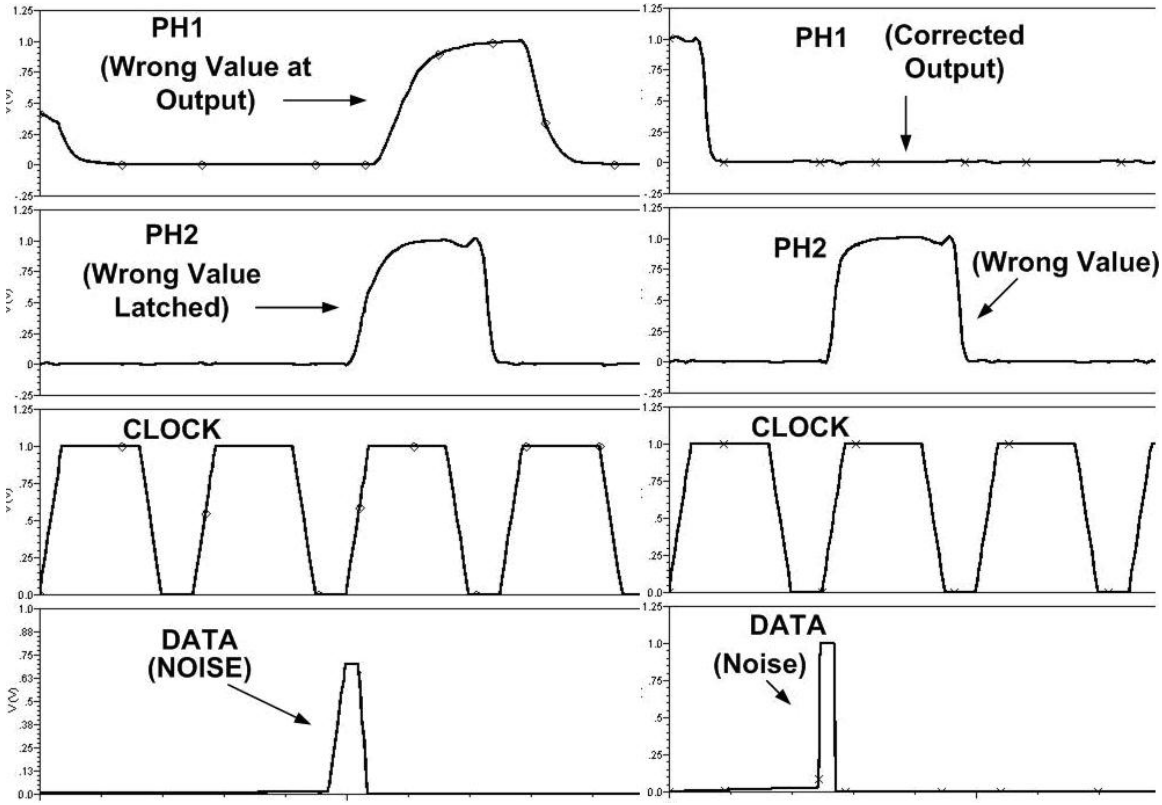


Figure 3.17: Response of BSFF *vs.* Response of XSEUFF2 to Noise Pulse.

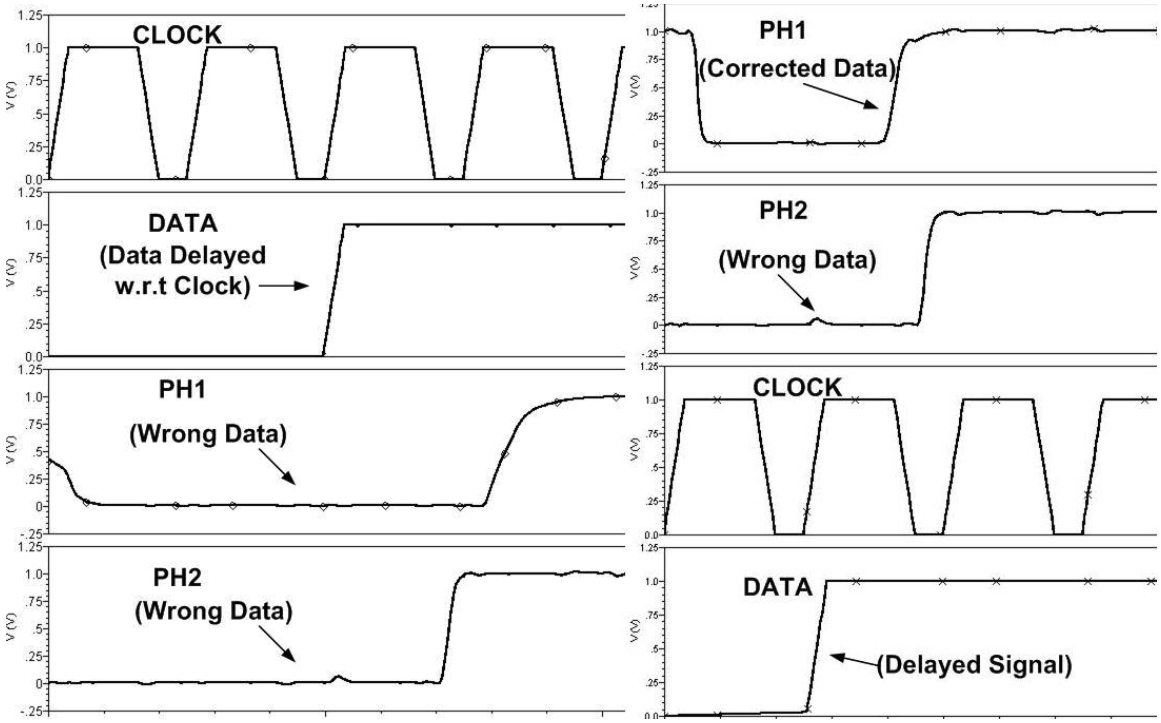


Figure 3.18: BSFF *vs.* XSEUFF2 w.r.t. Signal Delays.

3.3.3 Functional Analysis of the XSEUFF2

The design was simulated and its operation verified. During simulations, the XSEUFF2 was subjected to noise pulses and signal delays of varying magnitudes. Its response to a worst case transient pulse is shown in Figure 3.17. In Figure 3.17, we compare the behavior of the BSFF and the XSEUFF2. Latch PH1, in the BSFF, receives incorrect data from PH2 after the arrival of active clock edge. In the case of XSEUFF2, time-redundant samples are evaluated by the majority voter and PH1 receives corrected data at the end of the clock cycle. Figure 3.18 shows the response of the BSFF and the XSEUFF2 to signal delays. In Figure 3.18, although PH2 latched in a wrong value, PH1 received the corrected value from the majority voter circuit. Although the XSEUFF is able to transmit corrected data to PH1 in both cases, its tolerance is limited and depends on several circuit parameters. We define the limits for the width of a noise pulse and magnitude of signal delay that this FF can tolerate. If T_{XPulse} is the maximum tolerable width of a single event transient, then Equation 3.6 gives the relationship between the various sampling instants, the setup time of the FF, t_{setup},

the hold-time, t_{hold}, and the delays of the multiplexers at the input of latches LB and PH1. For instance $t_{cd(LB),r}$ represents the combinational delay of a rising transition at the input of LB. Similarly $t_{cd(LB),f}$ represents the combinational delay of a falling transition at the input of LB. Equation 3.7 gives the maximum tolerable delay (Δ_{max}) of the XSEUFF2.

$$T_{XPulse} = \{T_2 - t_{setup} + max(t_{cd(LB),r}, t_{cd(LB),f})\} - \{T_0 + t_{hold} - min(t_{cd(PH2),r}, t_{cd(PH2),f})\} \quad (3.6)$$

$$\Delta_{max} = \{T_1 - t_{setup} + max(t_{cd(LA),r}, t_{cd(LA),f})\} - \{T_0 - t_{setup}\} \quad (3.7)$$

3.3.4 Test Mode Operation

Here, operation of the XSEUFF2 during test mode is discussed. During Scan operation ($Scanmode = 1$), LA and LB become part of a large scan-chain. LA gets data from *SI*, which is connected to the output of a previous FF or the tester pin. *SCA* and *SCB* control data transfer between LA and LB during the test mode. *UPDATE* and *CAPTURE* are used to apply the test vector and capture the circuit response respectively. The XSEUFF2 can be tested for stuck-at-0 and stuck-at-1 faults on all its internal nodes by enabling the scan mode of operation and applying the appropriate vector. A stuck-at fault present on any of the input nodes of the majority voter can be tested by scanning in opposite values in latches LA and LB and disabling the signal generator, so that the value latched in PH2 can be used to sensitize this fault. The *Update* signal used in conjunction with the *Scanmode* signal keeps the value in PH1 stable while the response of the circuit is scanned out through LA and LB.

Chapter 4
Results

Layout level schematics for the *XTFF*, *XTFF2*, *XSEUFF2* were studied for area and timing overheads with respect to that for the *BSFF*. They were also compared with other designs, the *Error-Blocking Scan Flip-Flop* (EBSFF)[41], the *Error-Trapping Scan Flip-Flop* (ETSFF)[41] and the *Error-Blocking Scan Hold Flip-Flop*(EBSHFF) [20].

4.1 Transistor Overhead Comparison

Table 4.1: Transistor Overhead Comparison with Respect to the BSFF.

Flip-Flop	Normalized Transistor Count	Tolerant to SSEs	Tolerant to CSEs	Recover from Signal delays
BSFF	1.00	No	No	No
EBSFF	1.13	Yes	No	No
ETSFF	1.15	Yes	No	No
EBSHFF	0.91	Limited	No	No
XSEUFF1	1.54	Yes	Yes	No
XTFF	2.00	No	Yes	Yes
XTFF2	1.30	No	Yes	Yes
XSEUFF2	1.37	Yes	Yes	Yes

Table 4.1 shows the ratio of transistor counts present in various designs and also the extent to which each design is resilient to soft-errors. We observe that the XSEUFF2 is the most resilient and also has a higher hardware overhead as compared to the ETSFF and EBSFF. The design with a significantly high overhead is the XTFF2. This is due to the fact that most of the decision making hardware

in the *XEDCU* (refer to Figure 3.3) is realized through primitive gates that make optimization of the design difficult.

4.2 Propagation Delay and Power Overhead Comparisons

Here, we compare the Clock-Q delay of the proposed flip-flops. Clock-Q delay is defined as the time taken for a signal to reach the output of the flip-flop/latch after the arrival of the active clock edge/logic when required data is available on the input data line.

Table 4.2: Comparisons of Speed and Power.

Flip-Flop	Clock-to-Q Delay	Power Overhead
BSFF	1.00	1.00
EBSFF	1.23	2.13
ETSFF	1.05	2.26
EBSHFF	1.25	1.72
XSEUFF1	1.00	2.70
XTFF	1.40	3.10
XTFF2	1.21	1.75
XSEUFF2	1.30	2.50

Table 4.2 shows the ratios of propagation delays and power consumption for the various designs discussed in the previous chapters. The total power (static and dynamic) dissipation for XTFF is the highest due to the high number of synchronous signals used by it. The lowest is exhibited by the EBSHFF due to its lower transistor count and minimal use of synchronous signals. Table 4.3 indicates the magnitude of area overheads for certain ISCAS benchmarks.

Table 4.3: ISCAS '89 Benchmark Overheads for Proposed Flip-Flop Designs.

ISCAS Benchmark	XTFF Overhead (%)	XTFF2 Overhead (%)	XSEUFF2 Overhead (%)
s5378	53	16	20
s9234	40	12	15
s13207	58	17	22
s15850	49	14	19
s38417	54	16	21
s38584	49	15	19
s35932	57	17	22
Average	52	16	20

Chapter 5
Conclusion

We have proposed a design, the *Crosstalk and Soft–Error Tolerant Flip-Flop 2* (XSEUFF2), with novel combinations of space/time redundancies that immunize flip-flops from soft-errors and crosstalk effects. We have also discussed its limitations in terms of the maximum tolerable pulse width and difficulty of generating the required sampling signals. When compared with existing soft-error tolerant flip-flop designs, the *XSEUFF2* is the only one that is immune to both types of soft-errors (i.e., combinational soft-errors and sequential soft errors) and also to crosstalk effects (noise pulses and signal delays). This is because the *XSEUFF2* is the only flip-flop to use both spatial and time redundancy, in order to handle SEUs during the entire *Window of Vulnerability* (WoV). We compare our design, XSEUFF2, to the *Basic Scan Flip-Flop* (BSFF) of Mitra *et al.*[41], the *Error Blocking Scan Flip-Flop* (EBSFF) of Mitra *et al.*[41], the *Error Trapping Scan Flip-Flop* (ETSFF)[41] of Mitra *et al.*[41], and the *Error Blocking Scan Hold Flip-Flop* (EBSHFF) of Roy *et al.*[20]. Mitra's and Roy's designs do not protect the data in the WoV immediately around the clock edge. The area overhead of the *XSEUFF2* is 37% while that of the *EBSFF* and the *ETSFF* is 13% and 15%, respectively. The area overhead of the *EBSHFF* is 9% lower than that of the *BSFF*. The timing penalty of the *XSEUFF2* is 30% delay while that of the *EBSFF*, the *ETSFF* and the *EBSHFF* is 23%, 5% and 25%, respectively. The power overhead comparisons for the four designs are 250%, 213%, 226% and 172%. The average area overhead of the *XSEUFF2* on ISCAS benchmark circuits is 20%. These values indicate an increase in area, timing and power penalties, compared to earlier designs, which is greatly outweighed by the significant increase in overall circuit reliability, a critical issue for emerging technologies, particularly at 45 nm and

smaller feature sizes.

The other proposed designs, the *Crosstalk Tolerant Flip-Flop* (XTFF) and the *Crosstalk Tolerant Flip-Flop-2* (XTFF2) are tolerant to *Single-Event Transients* (SETs) that may arrive at the data input of the flip-flops around the active clock edge but not to those generated internally to the flip-flop. Area, timing and power penalties of the *XTFF2* are 20%, 21% and 75% greater than those of a *BSFF*. These overheads are lower than those incurred by the *XSEUFF2* making the *XTFF2* flip-flop suitable for designs where total immunity can be sacrificed to achieve higher operational frequency.

5.1 Future Work

This solution, like any other, is not scalable in terms of number of bit upsets. As we shrink further, a single particle-hit may upset multiple nodes. Today's *Single-Bit Upset* model must be applied to many nodes simultaneously to calculate this increased vulnerability of a design. *Multiple-Bit Upsets* (MBU) for memories have been studied in a far greater depth than for combinational logic [8, 29]. Work towards understanding effect of MBUs on existing latch-level SEU tolerant designs, such as the DICE cell, has been studied by Seifert *et al.* [56]. MBUs are most likely to become part of a reliability bottleneck for future devices and, hence, we need to focus on developing efficient architectures at various hierarchical levels, either in conjunction with each other or stand-alone, to overcome them.

References

[1] X. Bai and S. Dey. A High-Level Crosstalk Defect Simulation Methodology. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 23(9):1355–1361, September 2004.

[2] A. Balasubramanian, B. L. Bhuva, J. D. Black, and L. W. Masengill. RHBD Techniques for Mitigating Effects of Single-Event Hits Using Guard-Gates. *IEEE Trans. on Nuclear Science*, 52(6):2531–2535, December 2005.

[3] R. Baumann. The Impact of Technology Scaling on Soft Error Rate Performance and Limits to the Efficacy of Error Correction. In *Proc. of the Int'l. Electron Devices Meeting (IEDM)*, pages 329–332, 2002.

[4] J. M. Benedetto, P. H. Eaton, D. G. Mavis, M. Gadlage, and T. Turflinger. Variation of Digital SET Pulse Widths and the Implications for Single Event Hardening of Advanced CMOS Processes. *IEEE Trans. on Nuclear Science*, 52(6):2114–2119, December 2005.

[5] J. M. Benedetto, P. H. Eaton, D. G. Mavis, M. Gadlage, and T. Turflinger. Variations of Digital SET Pulse Widths and the Implications for Single Event Hardening of Advanced CMOS Processes. *IEEE Trans. on Nuclear Science*, 52(6):2114–2119, Dec. 2005.

[6] D. Bessot and R. Velazco. Design of SEU Hardened CMOS Memory Cells: The HIT Cell. In *Proc. of the RADECS Conf.*, pages 563–570, December 1993.

[7] D. K. Bhattacharrya and S. Nandi. Theory and Design of SEC-DED-AUED Codes. *IEEE Proc. – Computers and Digital Techniques*, 145(2):121–126, March 1998.

[8] S. Buchner, A. B. Campbell, T. Meehan, K. A. Clark, D. McMorrow, C. Dyer, C. Sanderson, C. Comber, and S. Kuboyama. Investigation of Single-ion Multiple-bit Upsets in Memories on Board a Space Experiment. In *Proc. of the European Conf. on Radiation and its Effects on Circuits and Systems*, pages 558–564, December 1999.

[9] M. L. Bushnell and V. D. Agrawal. *Essentials of Electronic Testing For Digital Memory and Mixed-Signal VLSI Circuits*. Springer, Boston, first edition, 2000. 2002 Printing.

[10] T. Calin, M. Nicolaidis, and R. Velazco. Upset Hardened Memory Design for Submicron CMOS Technology. *IEEE Trans. on Nuclear Science*, 43(6):2874–2878, December 1996.

[11] H. Cha and J. H. Patel. Latch Design for Transient Pulse Tolerance. In *Proc. of the ACM Int'l. Conf. on Computer Design*, pages 385–388, October 1994.

[12] C. L. Chen et al. Fault-Tolerance Design of the IBM Enterprise System/9000 Type 9021 Processors. *IBM J. of Research and Development*, 46(4):765–779, July 1992.

[13] W. Chen, S. K. Gupta, and M. A. Breuer. Test Generation in VLSI Circuits for Crosstalk Noise. In *Proc. of the Int'l. Test Conf.*, pages 641–650, 1998.

[14] S. DasGupta, R. G. Walther, T. W. Williams, and E. B. Eichelberger. An Enhancement to LSSD and Some Applications of LSSD in Reliability, Availability and Serviceability. In *Proc. of the Int'l. Symp. on Fault Tolerant Computing*, page 289, June 1981.

[15] P. E. Dodd. Physics Based Simulations of Single-Event Effects. *IEEE Trans. on Device and Materials Reliability*, 5(3):343–357, Sept. 2005.

[16] A. J. Drake, A. J. KleinOsowski, and A. K. Martin. A Self-Correcting Soft Error Tolerant Flop-Flop. In *Proc. of the 12th NASA Symp. on VLSI Design*, October 2005.

[17] J. F. Ziegler et al. IBM Experiments in Soft Fails in Computer Electronics. *IBM J. of Research and Development*, 40(1):3–18, January 1996.

[18] L. L. Sivo et al. Cosmic Ray-Induced Soft Errors in Static Memory Cells. *IEEE Trans. on Nuclear Science*, NS-26:5042–5047, 1979.

[19] A. Faridpour and M. Hill. Performance Implications of Tolerating Cache Faults. *IEEE Trans. on Computers*, 42(3):257–267, 1993.

[20] A. Goel, S. Bhunia, H. Mahmoodi, and K. Roy. Low-overhead Design of Soft-error-tolerant Scan Flip-flops with Enhanced-scan Capability. In *Proc. of the Asia and South Pacific Design Automation Conf.*, January 2006. 6 pp.

[21] M. Y. Hsiao, W. C. Carter, J. W. Thomas, and W. R. Stringfellow. Reliability, Availability and Serviceability of IBM Computer Systems: A Quarter Century of Progress. *IBM J. of Research and Development*, 25(5):453–469, September 1981.

[22] H. Imai. *Essentials of Error-Control Coding Techniques*. Academic Press, San Diego, CA, 1990.

[23] Y. I. Ismail, E. G. Friedman, and J. L. Neves. Figures of Merit to Characterize the Importance of On-chip Inductance. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 7(4):442–449, 1999.

[24] A. Jagirdar, R. Oliveira, and T. J. Chakraborty. A Novel Fault Recovery Method for Transient Pulses and Signal Delays. In *Proc. of the North Atlantic Test Workshop*, pages 112–119, 2006.

[25] A. Jagirdar, R. Oliveira, and T. J. Chakraborty. Efficient Flip-Flop Designs for SET/SEU Mitigation with Tolerance to Crosstalk Induced Signal Delays. In *Proc. of the Workshop on Silicon Errors in Logic-System Effects (SELSE 3)*, pages 123–128, 2007.

[26] M. Kamon, M. J.Tsuk, and J. K. White. FASTHENRY – A Multipole Accelerated 3-D Inductance Extraction Program. *IEEE Trans. on Microwave Techniques*, 42(9):1750–1758, September 1994.

[27] T. Karnik and P. Hazucha. Characterization of Soft-Errors Caused by Single Event Upsets in CMOS Processes. *IEEE Trans. on Dependable and Secure Computing*, 1(2):128–143, 2004.

[28] S. Kim and A. Somani. Area Efficient Architectures for Information Integrity Checking in the Cache Memories. In *Proc. of the Int'l. Symp. on Computer Architecture*, pages 246–256, 1999.

[29] R. Koga, S. D. Pinkerton, T. J. Lie, and K. B. Crawford. Single-word Multiple-bit Upsets in Static Random Access Devices. *IEEE Trans. on Nuclear Science*, 40(6):1941–1946, December 1993.

[30] S. Krishnamohan and N. R. Mahapatra. Combining Error Masking and Error Detection Plus Recovery to Combat Soft Errors in Static CMOS Circuits. In *Proc. of the Int'l. Conf. on Dependable Systems and Networks*, pages 40–49, July 2005.

[31] M. Lajolo. Bus Guardians: An Effective Solution for Online Detection and Correction of Faults Affecting System-On-Chip Buse. *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 9(6):974–982, December 2001.

[32] G. G. Langdon and C. K. Tang. Concurrent Error Detection for Group Look-ahead Binary Adders. *IBM J. of Research and Development*, pages 563–573, September 1970.

[33] L. W. Massengill, E. V. Kerns, S. E. Kerns, and M. L. Alles. Single-event Charge Enhancement in SOI Devices. *IEEE Electron Device Letters*, 11(2):98–99, February 1990.

[34] W. Massengill, M. Alles, and S. Kerns. SEU Error Rates in Advanced Digital CMOS. In *Proc. of the Second European Conf. on Radiation and its Effects on Components and Systems*, pages 546–553, Sept. 1993.

[35] D. Mavis and P. Eaton. Soft Error Rate Mitigation Techniques for Modern Microcircuits. In *Proc. of the Int'l. Reliability Physics Symp.*, pages 216–225, 2002.

[36] P. Mazumder. An On-Chip ECC Circuit for Correcting Soft-Errors in DRAMs with Trench Capacitors. *IEEE J. of Solid-State Circuits*, 24(11):1397–1403, October 1989.

[37] C. Metra, M. Favalli, and B. Ricco. Self-Checking Detection and Diagnosis of Transient, Delay and Crosstalk Faults Affecting Bus Lines. *IEEE Trans. on Computers*, 49(6):560–574, June 2000.

[38] S. Mitra and E. J. McCluskey. Which Error Detection Scheme to Choose? *Proc. of the Int'l. Test Conf.*, pages 985–994, October 2000.

[39] S. Mitra and E. J. McCluskey. Word-voter: A New Voter Design for Triple Modular Redundant Systems. In *Proc. of the 18th IEEE VLSI Test Symp.*, pages 465–470, May 2000.

[40] S. Mitra, N. Seifert, and P. Sanda. Soft Errors: Trends, System Effects and Protection Techniques. In *Proc. of the Int'l. Test Conf.*, pages 1–99, 2006.

[41] S. Mitra, M. Zhang, N. Seifert, Q. Shi, and K. S. Kim. Robust System Design with Built-In Soft-Error Resilience. *Computer*, 38(2):43–52, February 2005.

[42] R. Naseer and J. Draper. DF-DICE: A Scalable Solution for Soft-Error Tolerant Circuit Design. In *Proc. of the Int'l. Symp. on Circuits and Systems*, May 2006. 4 pp.

[43] M. Nicolaidis. Time Redundancy Based Soft-error Tolerance to Rescue Nanometer Technologies. In *Proc. of the VLSI Test Symp.*, pages 86–94, April 1999.

[44] T. J. O'Gorman, J. M. Ross, A. H. Taber, J. F. Ziegler, H. P. Muhlfeld, C. J. Montrose, H. W. Curtis, and J. L. Walsh. Field Testing for Cosmic Ray Soft Errors in Semiconductor Memories. *IBM J. of Research and Development*, 40(1):41–50, January 1996.

[45] P. Oldiges, K. Bemstein, D. Heidel, B. Klaasen, E. Cannon, R. Dennard, H. Tang, M. Ieong, and H. S. P. Wong. Soft Error Rate Scaling for Emerging SO1 Technology Options. In *Digest of Technical Papers, Symp. on VLSl Technology*, pages 46–47, June 2002.

[46] R. Oliveira. Novel Fault-Tolerant Scan Register Designs for Mitigation of Soft-Errors in Deep-SubMicron Technologies. Master's thesis, ECE Dept., Rutgers University, May 2007.

[47] R. Oliveira, A. Jagirdar, and T. J. Chakraborty. A Soft Error Tolerant, Low Overhead TMR Design for Flip-Flops. In *Proc. of the North Atlantic Test Workshop*, pages 97–104, 2006.

[48] R. Oliveira, A. Jagirdar, and T. J. Chakraborty. A TMR Scheme for SEU Mitigation in Scan Flip-Flops. In *Proc. of the Int'l. Symp. on Quality Electronic Design*, pages 905–910, 2006.

[49] M. Omana, D. Rossi, and C. Metra. Novel Transient Fault Hardened Latch. In *Proc. of the Int'l. Test Conf.*, pages 886–892, 2003.

[50] L. Penzo, D. Sciuto, and C. Silvano. VLSI Design of Systematic Odd-weight-column Byte Error Detecting SEC-DED Codes. In *Proc. of the 8th Int'l. Conf. on VLSI Design*, pages 156–160, January 1995.

[51] J. C. Pickel and J. T. Blandford. Cosmic Ray Induced Errors in MOS Memory Cells. *IEEE Trans. on Nuclear Science*, NS-25:1166–1171, December 1978.

[52] D. Pradhan. *Fault-Tolerant Computer System Design.* Prentice Hall, Englewood Cliffs, NJ, 1986.

[53] R. Reed. Heavy Ion and Proton Induced Single Event Multiple Upsets. *Proc. of the IEEE Nuclear and Space Radiation Effects Conf.*, pages 2224–2229, 1997.

[54] A. D. Sathe, M. L. Bushnell, and V. D. Agrawal. Analog Macromodeling of Capacitive Coupling Faults in Digital Circuit Interconnects. In *Proc. of the Int'l. Test Conf.*, pages 375–383, 2002.

[55] J. R. Schwank, V. Ferlet-Cavrois, M. R. Shaneyfelt, P. Paillet, and P. E. Dodd. Radiation Effects in SOI Technologies. *IEEE Trans. on Nuclear Science*, 50(3):522–538, June 2003.

[56] N. Seifert, B. Gill, V. Zia, and V. Ambrose M. Zhang. Assessing the Impact of Scaling on the Efficacy of Spatial Redundancy Based SER Mitigation Schemes. In *Proc. of the Workshop on Silicon Errors in Logic-System Effects (SELSE 3)*, pages 13–18, 2007.

[57] H. Shin. Modeling of Alpha-particle-induced Soft Error Rate in DRAM. *IEEE Trans. on Electron Devices*, 38(9):2465–2471, November 1991.

[58] P. Shirvani and E. J. McCluskey. PADded Cache: A New Fault-Tolerance Technique for Cache Memories. In *Proc. of the 17th IEEE VLSI Test Symp.*, pages 440–445, 1999.

[59] P. Shivkumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi. Modeling the Effect of Technology Trends on Soft-Error Rate of Combinational Logic. In *Proc. of the Int'l. Conf. on Dependable Systems and Networks*, pages 389–398, 2002.

[60] A. Sinha, S. K. Gupta, and M. A. Breuer. Validation and Test Issues Related to Noise Induced by Parasitic Inductances of VLSI Interconnects. *IEEE Trans. on Advanced Packaging*, 25(3):329–339, August 2002.

[61] L. Spainhower and T. A. Gregg. S/390 Parallel Enterprise Server G5 Fault Tolerance. *IBM J. of Research and Development*, 43:863–873, September-November 1999.

[62] H. H. K. Tang. Nuclear Physics of Cosmic Ray Interaction with Semiconductor Materials: Particle-induced Soft Errors from a Physicists Perspective. *IBM J. of Research and Development*, 40(1):91–108, January 1996.

[63] J. von Neumann. Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components. In C. Shannon and J. McCarthy, editors, *Automata Studies – Annals of Mathematical Studies*, number 34, pages 43–98. Princeton University Press, 1956.

[64] F. Wang, Y. Xie, R. Rajaraman, and B. Vaidyanathan. Soft Error Rate for Combinational Logic Using an Accurate Electrical Masking Model. In *Proc. of the Int'l. Conf. on VLSI Design*, pages 165–170, 2007.

[65] C. F. Webb and J. S. Liptay. A High Frequency Custom S/390 Microprocessor. *IBM J. of Research and Development*, 41(4/5):463–474, 1997.

[66] N. H. E. Weste and D. Harris. *CMOS VLSI Design, A Circuits and Systems Perspective*. Addison-Wesley, Boston, third edition, 2005.

[67] M. A. Xapsos. Applicability of LET to Single Events in Microelectronic Structures. *IEEE Trans. on Nuclear Science*, 36(9):1613–1621, December 1992.

[68] J. Xiong and L. He. Full Chip Routing Optimization with RLC Crosstalk Budgeting. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 23(3):1036–1049, March 2004.

[69] Y. Zhao, L. Chen, and S. Dey. On-line Testing of Multi-source Noise-induced Errors on the Interconnects and Buses of Systems-on-Chips. In *Proc. of the Int'l. Test Conf.*, pages 491–499, 2002.

[70] Y. Zhao and S. Dey. Analysis of Interconnect Crosstalk Defect Coverage of Test Sets. In *Proc. of the Int'l. Test Conf.*, pages 492–501, 2000.

[71] J. F. Ziegler. Terrestrial Cosmic Rays. *IBM J. of Research and Development*, 40(1):19–39, Jan. 1996.

[72] J. F. Ziegler et al. IBM Experiments in Soft Fails in Computer Electronics (1978-1994). *IBM J. of Research and Development*, 40(1):3–18, 1996.