UNIVERSITY OF CALIFORNIA, SAN DIEGO

Analysis and Design of Reliable Nanometer Circuits

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Computer Engineering)

by

Chong Zhao

Committee in Charge:

Professor Sujit Dey, Chair
Professor Chung-Kuan Cheng
Professor Rajesh Gupta
Professor Bill Lin
Professor Curt Schurgers

2007

The dissertation of Chong Zhao is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2007

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Prof. Sujit Dey for being an excellent research advisor. Thank you for guiding me to this fascinating research topic, for teaching me how to be an independent researcher as well as a constructive team player, for introducing me to the best in the field, and for being the support I can always count on. I would also like to thank my thesis committee members, Professor Chung-Kuan Cheng, Professor Rajesh Gupta, Professor Bill Lin, Professor Curt Schurgers, for providing valuable feedbacks to this work.

I am fortunate to be a member of the Embedded System Design, Test, and Automation Lab (ESDAT) at UC San Diego. Graduate school is a unique experience that enriches one's life. My friends at ESDAT have made this experience fun and memorable. Thank you, Dr. Yi Zhao, Dr. Li Chen, Dr. Xiaoliang Bai, Dr. Kanishka Lahiri, Dr. Dong-Gi Lee, Dr. Kanishka Lahiri, Dr. Krishna Sekar, Dr. Clark Taylor, Shoubhik Mukhopadhyay, Debashis Panigrahi, Naomi Ramos, Saumya Chandra, Myank Tiwari, for many discussions and laughter, and for making me feel the warmth of a big family. Special thanks to Cathy MacHutchin, for being a wonderful friend and for being amazingly organized and efficient in administrative work.

I would like to acknowledge the Gigascale Silicon Research Center (GSRC) for providing financial support for part of this work.

The text of the following chapters, in part or in full, is based on material that has been published in conference proceedings or journals, or is pending publication in

conference proceedings or journals. In particular, Chapter 2 is based on material in the published paper: Chong Zhao, Xiaoliang Bai, Sujit Dey, "Soft Spot Analysis: A Scalable Methodology Targeting Compound Noise Effects in Nano-meter Circuits", IEEE Design & Test of Computers, Volume 22, Issue 4, July-Aug. 2005, pp. 362-375, and material in the published paper: Chong Zhao, Xiaoliang Bai, Sujit Dey, "A Scalable Soft Spot Analysis Methodology for Compound Noise Effects in Nano-meter Circuits," in Proceedings of 41st Design Automation Conference (DAC), pp. 894-899, June 2004, San Diego, California, USA. Chapter 3 is based on material in the published paper: Chong Zhao, Xiaoliang Bai, Sujit Dey, "A Static Noise Impact Analysis Methodology for Evaluating Transient Error Effects in Digital VLSI Circuits", in Proceedings of International Test Conference 2005 (ITC), pp. 40.2, October, 2005, Austin, Texas, USA, and material in the published paper: Chong Zhao, Xiaoliang Bai, Sujit Dey, “Evaluating transient error effects in digital nanometer circuits”, IEEE TRANSACTIONS ON RELIABILITY, VOL. 56, NO. 3, SEPTEMBER 2007, pp. 381-391. Chapter 4 is based on material in the published paper: Chong Zhao, Yi Zhao, Sujit Dey, "Constraint-Aware Robustness Insertion for Optimal Noise-Tolerance Enhancement in VLSI Circuits," in Proceedings of 42nd Design Automation Conference (DAC), pp. 190-195, June 2005, Anaheim, California, USA, and material in the paper accepted by IEEE Transactions on Very Large Scale Integration Systems (TVLSI): Chong Zhao, Yi Zhao, Sujit Dey, “An Intelligent Robustness Insertion Methodology for Optimal Transient Error Tolerance”. Chapter 5 is based on material in the published paper: Chong Zhao, Sujit Dey, "Improving Transient Error Tolerance of Digital VLSI Circuits Using RObustness COmpiler (ROCO)", in Proceedings of 7th International Symposium on Quality

Electronic Design (ISQED), pp. 133-138, March 2006, San Jose, California, USA. Chapter 6 is based on material to be published in the Proceedings of International Conference on Computer Design (ICCD), 2007, Lake Tahoe, California, USA: Chong Zhao, Sujit Dey, "Modeling Soft Error Effects Considering Process Variations". I was the primary researcher and author of each of the above publications, and the co-authors listed in these publications collaborated on, or supervised the research which forms the basis for these chapters.

VITA

EDUCATION

1990 – 1995	B.S., Department of Physics, Peking University, Beijing, P. R. China
1996 – 1997	M.S., Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA
1995 – 1997	M.A., Department of Physics and Astronomy, University of Southern California, Los Angeles, CA, USA
2003 – 2007	Ph.D., University of California at San Diego, La Jolla, CA, USA

WORK EXPERIENCE

1997 – 1998	Sony Electronic, Rancho Bernardo, CA, USA
1998 – 1999	Conexant Systems Inc. San Diego, CA, USA
1999 – 2002	Mindspeed Technologies, Inc. San Diego, CA, USA
June 2003 – Sept. 2003	Intern, Intel Corporation, Santa Clara, CA, USA

JOURNAL PUBLICATIONS

1. Chong Zhao, Xiaoliang Bai, Sujit Dey, "Soft Spot Analysis: A Scalable Methodology Targeting Compound Noise Effects in Nano-meter Circuits", IEEE Design & Test of Computers, Volume 22, Issue 4, July-Aug. 2005, pp. 362-375.

2. Chong Zhao, Xiaoliang Bai, Sujit Dey, “Evaluating transient error effects in digital nanometer circuits”, IEEE TRANSACTIONS ON RELIABILITY, VOL. 56, NO. 3, SEPTEMBER 2007, pp. 381-391.

3. Chong Zhao, Yi Zhao, Sujit Dey, “An Intelligent Robustness Insertion Methodology for Optimal Transient Error Tolerance”, accepted by IEEE Transactions on Very Large Scale Integration Systems.

CONFERENCE PUBLICATIONS

4. Chong Zhao, Xiaoliang Bai, Sujit Dey, "A Scalable Soft Spot Analysis Methodology for Compound Noise Effects in Nano-meter Circuits," in Proceedings of 41st Design Automation Conference (DAC), pp. 894-899, June 2004, San Diego, California, USA.

5. Chong Zhao, Yi Zhao, Sujit Dey, "Constraint-Aware Robustness Insertion for Optimal Noise-Tolerance Enhancement in VLSI Circuits," in Proceedings of 42nd Design Automation Conference (DAC), pp. 190-195, June 2005, Anaheim, California, USA.

6. Chong Zhao, Xiaoliang Bai, Sujit Dey, "A Static Noise Impact Analysis Methodology for Evaluating Transient Error Effects in Digital VLSI Circuits", in Proceedings of International Test Conference 2005 (ITC), pp. 40.2, October, 2005, Austin, Texas, USA.

7. Chong Zhao, Sujit Dey, "Improving Transient Error Tolerance of Digital VLSI Circuits Using RObustness COmpiler (ROCO)", in Proceedings of 7th International Symposium on Quality Electronic Design (ISQED), pp. 133-138, March 2006, San Jose, California, USA.

8. Chong Zhao, Sujit Dey, “Evaluating and Improving Transient Error Tolerance of CMOS Digital VLSI Circuits,” in Proceedings of the International Test Conference 2006 (ITC), pp. 29.1, October 2006, Santa Clara, California, USA.

9. Chong Zhao, Sujit Dey, “Modeling Soft Error Effects Considering Process Variations”, to be presented on the International Conference on Computer Design (ICCD), October 2007, Lake Tahoe, California, USA.

ABSTRACT OF THE DISSERTATION

Analysis and Design of Reliable Nanometer Circuits

by

Chong Zhao

Doctor of Philosophy in Electrical Engineering (Computer Engineering)

University of California, San Diego, 2007

Professor Sujit Dey, Chair

As CMOS technology advances to the nanometer scale, semiconductor industry is enjoying the ever-increasing capability of integrating more and more devices and elements on a single die. Meanwhile, the reliability of the integrated circuit (IC) product is being severely challenged, as many previously negligible noise effects are becoming more prominent, causing significant performance and reliability degradations of nanometer integrated circuits.

In particular, radiation-induced transient error is quickly evolving to a serious limiting factor in the circuit reliability. Unfortunately, it has not been sufficiently and successfully addressed in previous technology generations, especially in cost-sensitive mainstream applications. Due to tight design constraint, budget and application

requirement, traditional redundancy-based techniques that have been exploited in space and mission-critical applications are no longer applicable. There is an urgent need for cost-effective techniques, methodologies and flows to facilitate the development of reliable IC products.

This dissertation is dedicated to the quest for solutions in the analysis, design and optimization of highly error-tolerant nanometer circuit systems. As will be elaborated and demonstrated throughout the entire dissertation, all developed techniques and methodologies share distinguished characteristics of being novel, accurate, economical, practical and scalable, as compared to other existing works. Together they form a unified and automated reliability optimization framework that will enrich the legacy of the IC design industry.

Chapter 1. Introduction

Silicon technology has advanced relentlessly following the Moore's law (i.e., doubling of the chip density every 1.5 years) [1] for the past four decades. This trend is likely to continue for the next decade in spite of the enormous investment needed in the manufacturing facilities and great difficulties anticipated in extending the CMOS scaling to its ultimate limits. As CMOS technology evolves into the nanometer regime, advanced manufacturing technology and comprehensive computer-aided design (CAD) techniques enable multi-million transistors to be manufactured on a single die, and multiple components to be integrated onto a single chip to become a "System-on-Chip (SoC)". Meanwhile, the semiconductor industry has to cope with two major challenges: the ever-increasing design complexity and complicated physical effects inherent from the nanoscale technology. Many previously negligible effects are becoming prominent and the reliability of cutting-edge nanometer circuits are being severely endangered.

1.1 Reliability Concerns in Nanometer Circuits

In digital circuits, information is encoded and processed as signals in the form of logic 1 and 0 (corresponding to different voltage levels), which are transmitted through logic gates and interconnects. During this process, disturbances in the forms of current and voltage variations, or "*noise*", may potentially damage the original signal, resulting in distorted or erroneous data to be generated or transmitted. Fortunately, these signals

possess a certain level of resilience to various noise sources, so that they may be safely decoded back to the intended data even in a noisy environment. However, if signal distortions exceed the device noise margin, the correct values might not be able to be restored. In this dissertation, "*reliability*" is a term used to describe the tendency of the circuit to restore the distorted signals and its ability to operate properly with the presence of noise interferences.

Complementary Metal Oxide Semiconductor (CMOS) digital circuits are believed to have high reliability due to their exceptional capability of restoring distorted signals (as compared to their analog counterparts). For this very reason, noise had always been considered as secondary effects in the previous technology generations and designers have been relying on this inherent tolerance of the circuit to self-protect or self-rescue from various noise interferences. Consequently, the objective of chip design and implementation has been primarily set on making the products capable of performing expected functionalities under certain design constraints including speed, area and power, whereas reliability have not been included in the design metric.

With the feature size shrinks to nanometer scale, reliability degradation has become a serious design concern that may cause significant yield loss and performance impact if not properly addressed. Several factors collectively contribute to the reliability degradation. First, as the device dimension becomes comparable to the atoms in the semiconductor material, many noise sources become significantly more prominent than ever. Second, as the supply voltage reaches sub-volt range, noise margin of the semiconductor devices has been greatly reduced, which means the required noise level to

cause irreversible distortions is much lower. Third, the number of devices on a single die is increasing exponentially with the integration capability, and they are operating at a muck faster rate as the operating frequency reaches multi-GHz range. This results in stronger and more frequent interactions among adjacent devices that lead to stronger error effects and higher failure rate.

Due to these technology trends, the reliable functioning of VLSI circuits is being greatly threatened. If the potential vulnerabilities are not properly addressed during the design phase, the manufactured chip may be highly unreliable. Therefore, more comprehensive and improved reliability criteria should be implemented into the design flow via different levels of abstraction. In their 2005 edition, the International Technology Roadmap for Semiconductors (ITRS) [2] predicts that "Design-for-Reliability" (DFR) will become an important practice to achieve a high level of error tolerance; and reliability-aware design will become indispensable in the current and next technology generation.

Failures experienced by nanometer circuits can be either permanent or transient [19]. A permanent failure causes irreversible damage and malfunction of the chip. Examples of noise sources that cause permanent failures include time-dependent dielectric breakdown (TDDB), hot carrier injection (HCI), negative bias temperature instability (NBTI) in transistors and electromigration in interconnects. In contrast to permanent failures, transient failures occur occasionally because of temporary environmental conditions and last only for a short period of time. Examples of noise sources that cause transient failures include power supply and interconnect noise,

electromagnetic interference, electrostatic discharge and radiation-induced soft errors. Among them, the radiation-induced soft error has become one of the most serious transient failures in nanometer circuits. It is caused by highly energetic particles striking the sensitive regions in semiconductor devices and is often referred to as a "single-event-upset (SEU)".

SEU will not cause permanent chip damage and its error effects will not persist as the noise source disappears. Because of the unpredictable nature of the particle activities (in cosmic rays or semiconductor materials), it is extremely difficult to detect, diagnose and prevent such transient error effects. Due to its low occurrences, until recent technology generation, SEU has been only a concern in space and mission-critical applications, where reliability is the most important performance requirement and cost is not a limiting factor. The most commonly adopted SEU mitigation techniques involve using redundancy at various abstraction levels to provide extra protections. As the technology reaches nanometer scale, as was mentioned above and will be elaborated in detail later, SEU has become a non-negligible error source even in mainstream applications (such as consumer electronics), whose cost-sensitive nature prohibits the extensive use of redundancy. Furthermore, as the circuit complexity and level of integration significantly increase, efficient yet economical SEU detection and prevention pose great challenges to the semiconductor industry.

1.2 Radiation-Induced Single-Event-Upset (SEU)

The remaining part of this chapter gives an overview of SEU, including the basic mechanism of particle strikes in semiconductor devices, a brief history of SEU-related

research work, and its impacts on the functionality of VLSI circuits. It also reviews the contributions and limitations of previous research efforts on SEU detection and prevention.

1.2.1 Basic Mechanism of Particle Strikes on Semiconductor Devices

When a cosmic particle enters a semiconductor material, it deposits charges by freeing electron-hole pairs along its path as it loses energy. The particle of particular interest is the heavy neutron, which constitutes ~92% of all particles in the terrestrial environment [37]. The energy level considered is in the range of 1-10 MeV, because particle strikes with higher energy occur with significantly low probability and they usually cause permanent failure instead of transient upsets. The *linear energy transfer* (LET) is frequently used to relate the energy of the incident particle to the charge deposition in a particular type of material. LET is defined as the energy loss of the particle per unit path length in the material and has the unit of $MeV\text{-}cm^2/mg$. In bulk silicon, a typical charge collection depth λ_c is ~2μm for an LET of 1 MeV-cm^2/mg, and an ionizing particle deposits q_d =10.8fC of charge along each micron of its track. Thus a particle with LET of $1 MeV\text{-}cm^2/mg$ deposits ~21.6fC of charge [46].

The most sensitive regions to collect the deposited charge in the struck device are the reverse-biased p/n junctions. The high field present in a reverse-biased junction depletion region can efficiently collect the particle-induced charge through drift processes, leading to a transient current at the junction contact [50]. This transient current may potentially damage signals generated and propagated in a circuits, and eventually

lead to erroneous circuit behavior and observable error effects. The exact error impact depends on the type of the struck device as well as the nature of the affected circuit and will be discussed in more detail in section 1.3.

Natural particle activities has strong timing, geographical and altitude dependencies: for a specific orbit during a certain period of time, it is characterized by the *LET spectrum Φ(LET)*, defined as the number of particles detected on a unit area per unit time as a function of the particle's LET [49]. In the atmosphere, particle flux is typically in the range of 10^{-5}-10^{1}cm^{-2}day^{-1}, and there is a sharp knee in the natural LET spectrum at an LET ~30 and particles with LET above 30 are exceedingly rare [9].

1.2.2 A Brief History

The first SEU-related work was published in 1962 [51], forecasting the eventual occurrence of SEU in microelectronics due to cosmic rays. The first confirmed report of cosmic-ray-induced upsets in space was presented at the NSREC in 1975 [52], which reported four observed upsets in 17 years of operation in a communication satellite. Because the numbers of errors observed was so small, it had been several years before the importance of SEU was fully recognized. In the late 1970s, evidence continued to mount that cosmic-ray-induced upsets were indeed responsible for errors observed in satellite memory subsystems, and the first models for predicting system error rates were formulated [54]. In the 1980s, research on SEU continued to increase and methods for hardening ICs to SEU were widely developed [55] [56]. Studies of SEU in random logics appeared in the late1980s, but were often overshadowed by the volume of work

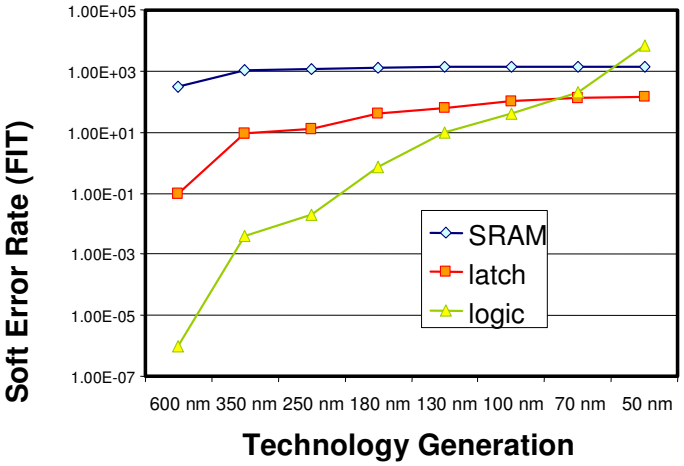
addressing memory upset [57] [58]. In the 1990s, a surge of interest in random logic SEU emerged, fueled by several factors including: 1) a perception that the memory soft error was controllable with advanced technologies and error detection and correction (EDAC) techniques [59]; 2) a growing concern that technology scaling could lead to an inversion between the relative significance of memory and logic on observed soft error rates [60]; and 3) observations that clock speeds were driving up core-logic error rates [61].

As the semiconductor industry enters the 21st century, the SEU sensitivity of IC products is expected to continuingly increasing. SEU vulnerability has become a mainstream product reliability metric of IC industry, as outlined by the SEMATECH National Industry Association Roadmap [62]. The feasibility of traditional SEU-hardening techniques is becoming questionable; while circuit designs that are inherently radiation resistant, known as "hardening by design" (HBD), are receiving considerable attention [63]. Methodologies of cost-effective SEU resilient circuit and system design have drawn tremendous research interest recently in the nanometer scale IC products.

1.2.3 Technology Trends

With the technology scaling, soft error rate (SER) in VLSI circuits drastically increases. In particular, random logic SER increases at a higher rate than memory SER. In [27], it has been concluded that when the feature length decreases from 0.6μm to 0.1μm, soft error rate (SER) in memory chips (in the unit of FIT/bit, where FIT="Failure-In-Time", gives the number of failure in 10^9 hours) remains constant but the number of bits on a die increases quadratically over each technology generation so the number of failures

experienced by a single die increases moderately. Meanwhile, random logic SER has increased by a factor of 10^7, and is expected to surpass SER in unprotected memories in the year of 2011. Furthermore, memory devices can be effectively protected by various schemes, such as error correction code (ECC) [64]. Therefore, for products that use ECC to protect the on-chip memories, logic will quickly become the dominant error source.



Source: P. Shivakumar et. al., ICDSN'02 (IBM & UTA)

Figure 1-1 SEU Technology Trend

1.3 SEU in CMOS Digital Circuits

In COMS digital circuits, when a particle penetrate both the source and drain of the struck transistor in a circuit elements, it results in a significant (but short-lived) source-drain current that mimics the "ON" state of the transistor and causes a temporary voltage swing around the struck node. There are two types of elements in static CMOS digital circuits: sequential elements, such as flip-flops (FFs), have the capability to store (or "memorize") logic values; whereas combinational elements can only alter the logic

level of the incoming signals as they propagate through. The particle-induced voltage swing may lead to different error behavior in different types of elements.

1.3.1 SEU in Combinational Logics

Figure 1-2(a) shows the situation when a particle strikes the PMOS transistor in a CMOS inverter and the corresponding linear RC modeling. R_D is the equivalent resistor of the NMOS transistor network. The transient current from the source to the drain of the PMOS transistor caused by the strike has a double-exponential form [42]:

$$I(t) = \frac{Q}{(\tau_\alpha - \tau_\beta)}(e^{-t/\tau_\alpha} - e^{-t/\tau_\beta}) \quad \textbf{(1.1)}$$

where $Q = LET^* \lambda_c * q_d$, is the total deposited charge; τ_a is called the "collection time constant"; and τ_β is called the "ion track establishment time constant". In the current technology node, typical values are ~1.64x10^{-10} sec for τ_α and ~5x10^{-11} sec for τ_β. This transient current can be realized as a voltage controlled current source, and the independent voltage source provides the double-exponential term.

When the input to the inverter is "1", the nominal output is "0", and the PMOS transistor is in its "OFF" state. However, as the result of this current flow, the PMOS transistor will be temporarily shorted and a transient voltage pulse *V(t)* (called a "Single-Event-Transient", or a "SET") will appear at the output. In [48], an approximate closed-form expression of *V(t)* was obtained:

$$V(t) = \frac{Q}{C_n \tau_\alpha} e^{-t/\tau_n} \left(\frac{e^{t/\tau_n} \cdot e^{-t/\tau_\alpha} - 1}{1/\tau_n - 1/\tau_\alpha}\right) \quad \textbf{(1.2)}$$

where $\tau_n = C_n * R_D$; C_n is the total capacitance (sum of the output capacitance C_o and the load capacitance C_l). In the above equation, the contribution of τ_β is ignored as it is relatively small compared to that of τ_α.

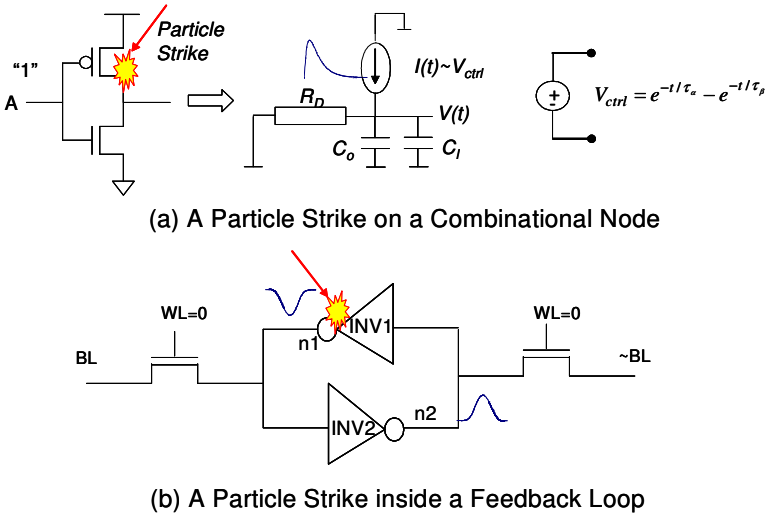


Figure 1-2 Particle Strikes in Digital Circuits

Figure 1-3 shows the transient waveform $V(t)$ measured at the output of the inverter in Figure 1-2(a). It is usually modeled as a square shaped glitch $g(w,h)$ with certain duration w and amplitude h. The duration w is measured at the 50% of supply voltage V_{dd}; whereas the amplitude h is measured as the maximum deviation from the nominal voltage level. This glitch caused by strikes on a combinational node is often referred to as a "single-event-transient (SET)".

1.3.2 SEU in Sequential Elements

If a particle strikes a transistor in a sequential element with a regenerative feedback (e.g. a flip-flop), the generated transient voltage can strengthen itself through the

propagation in the feedback loop and eventually stabilize as a static error. Figure 1-2(b) shows an example of a strike on a typical cell with a feedback loop: when the word line (WL) is low, the cell is holding its stored data. If a particle strike on INV1 causes node n1 to transition, the disturbance may propagate forward through INV2 and cause a transition on node n2. The feedback loop will cause both nodes to flip and the memory cell will reverse its state to a wrong value. Once the cell flips, it will not be recovered unless it is rewritten via the bitlines. Furthermore, when the WL is high, an external transient may be able to reach the feedback look and become a stable error. Therefore, sequential elements play a crucial role in SEU error rate in digital circuits.

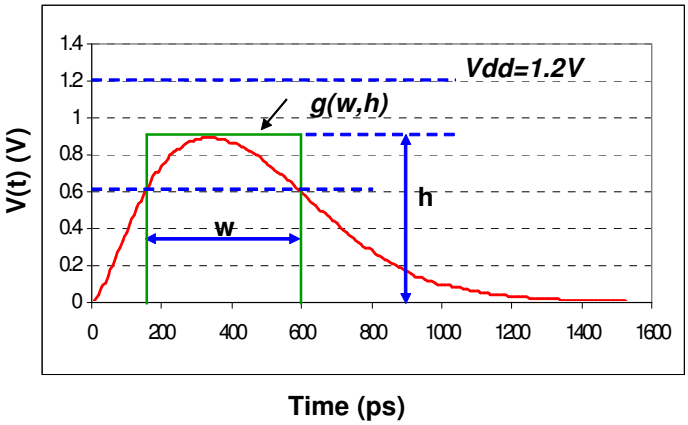


Figure 1-3 Transient Voltage and The Square-Glitch Model *g(w,h)*

1.3.3 Inherent SEU Immunity of Digital Circuits – Masking Effects

Compared to direct particle strikes in the sequential elements, transient errors caused by strikes in combinational gates are much more difficult to analyze. First, the strength of the incurred voltage swing depends on not only the incident energy, but also the property and dimension of the struck transistor, as well as the supply voltage and the

loading condition. Second, a SET will not become a stable error unless it is later captured by a FF. There exists three "masking effects" (electrical masking, latching-window masking and logic masking) that can prevent a transient from being captured. Therefore, the observable error rate depends on the electrical, timing and structural properties of the entire combinational logic.

The example circuit shown in Figure 1-4 illustrates the three masking effects: a particle strikes the PMOS transistor in the inverter G1 and incurs a positive transient glitch *g[w,h]* at its output node.

1) Electrical masking: the amplitude of the generated transient has to be larger than the input noise margin of a subsequent gate in order to continue its propagation as a legitimate digital pulse. In the example, the glitch can possibly propagate to the output of G3 only if its amplitude *h* is higher than the input noise margin of G3.

2) Logic masking: the generated transient has to be located on a sensitized logic path to reach the endpoint FF. If it reaches a logic gate whose output value is completely decided by controlling value of the gate on the side inputs, it will cease to further propagate. In the example, if IN1=1, the glitch will not reach the output of G3; even when IN1=0, if both IN2 and IN3 equal to 1, the input B to the OR gate G5 will be 1, so the glitch at input A of G5 will still be logically masked. It is easily found that the only occasions that the glitch can reach the FF are {IN2, IN3, IN4}= 101, 110, or 100.

3) Latching-window masking: the generated transient has to arrive at the input of a FF within a timing window ("sampling window") to be captured because a FF

is insensitive to any signal arrives outside the sampling window. The sampling window is bounded by the setup time (t_{su}) and the hold time (t_h). Since the glitch will be phase delayed as it propagates through the intervening gates en route to the DFF, to arrive at the DFF within its sampling window, the glitch at the original struck gate has to meet certain timing requirement.

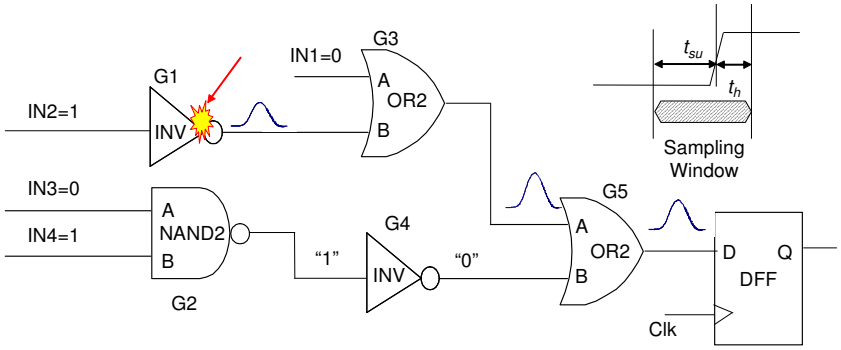


Figure 1-4 Masking Effects in Digital Circuits

The strengths of the three masking effects are purely determined by the electrical, logic and timing structure of the circuit and independent of the external particle activities. Therefore, digital circuit is considered to have inherent tolerance to single event transient.

1.4 Previous SEU-Related Research Work

Error effects caused by the particle-induced transient current are non-persistent – they will not be able to be detected when the noise source disappears. This makes SEU detection, diagnosis and correction extremely difficult. Ever since the discovery of the SEU effect, researchers in semiconductor companies and universities have spent tremendous efforts and resources on the methodologies and techniques to prevent it from causing damage to electronic products.

1.4.1 SEU Modeling and Analysis

The SEU effects can be studied at various levels and through either dynamic fault simulation or static analysis.

At the device-level, the most commonly used formalism in simulation is the drift-diffusion models [69] [70], where the semiconductor device equations are derived from the Boltzmann transport equation using numerous approximations. The equations to be solved are the Poisson equation and the current continuity equations. Although device-level simulations provide the most accurate modeling, they are extremely time-consuming and computationally intensive so they can not be used directly as the SEU vulnerability metric without higher-level abstraction.

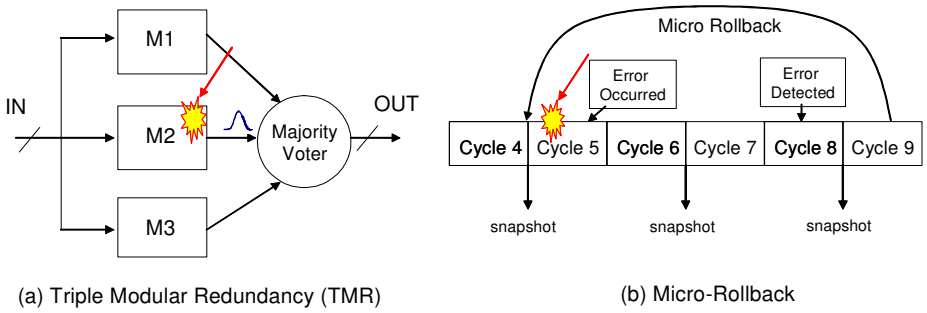
Stepping up in the hierarchical view, these models can be incorporated into macro-models of the devices interconnected in a sub-circuit, where the charge collection in individual device junctions is related to changes in the circuit currents and voltages. A common circuit model for the charge collection at a junction due to direct funneling or diffusion is the double-exponential, time-dependent current pulse [42] [72]. Deterministic circuit simulation of logic circuitry has been effectively performed in the circuit domain using industry standard tools such as Synopsys HSPICE, Orcad PSPICE, Cadence Spectre. Methods to track radiation vulnerability at the circuit level have emerged primarily in the realm of random logic. For example, [71] presented a probabilistic description of single-event fault generation, propagation, and logic error events using a high-level HDL circuit description.

Although simulation-based analysis is generally accurate and easy to implement, as circuits grow exponentially in density and complexity, comprehensive circuit simulation is becoming impractical; and method based on dynamic simulations inevitably faces the scalability challenge. Naturally, static method of SEU analysis has become the target of many research works. [73] [74] both have presented analytical descriptions of core logic soft error vulnerability based on the "window of vulnerability" of in-data-path static and dynamic latch elements, the synchronous clock, and other deterministic elements. Method for the identification of SEU vulnerabilities have also been developed [36] [68].

1.4.2 SEU Mitigation

Traditionally, SEU has been only a concern in space applications because the density of cosmic particles in outer space is exceedingly higher than at the ground level so the soft error rate observed in ground applications is negligibly small. For example, in 0.6μm technology, the soft error rate in typical unprotected SRAM at ground level is in the order of 10^2-10^4 FIT/Mbit [7]. For a typical 64Kbit SRAM, one failure may occur at most every $1.7x10^2$ years. However, reliability is the most critical requirement in space applications and even a low transient error rate is unacceptable. In contrast, cost is at most a secondary concern to system reliability. Therefore, significant amount of resources can be spent to achieve a high level of SEU tolerance. Many early SEU mitigation techniques adopted in space applications are at high level of design abstraction such as the module level or even system level because they are easy to implement and more flexible.

Triple modular redundancy (TMR) [38] has been a popular module-level scheme to detect and correct soft errors. As shown in Figure 1-5(a), the module under protection has three identical instances (M1, M2 and M3) that perform the same functionality simultaneously. If M2 endures a particle strike, its output may differ from the other two, but the correctness of the module will be guaranteed by a "Majority Voter". TMR will almost ensure the correctness of the protected module because the probability of two modules suffering a strike at the same time is infinitesimal. An area overhead of 300% is the protection cost of TMR. In extremely critical components, the protected module can be as large as an entire microprocessor chip.



(a) Triple Modular Redundancy (TMR) (b) Micro-Rollback

Figure 1-5 Traditional SEU Mitigation Techniques

Micro-rollback is a system-level soft error mitigation technique [65] that is able to correct the transient error effect by bringing the system back a state prior to the occurrence of the error. In order to be able to perform such an operation, it is necessary to save "snapshots" of some states of the system. As shown in Figure 1-5(b), in case of an earlier error (during cycle 5) being detected (during cycle 8), the error can be corrected by overwriting the current state (cycle 9) with a snapshot (cycle 4) taken in the past. Micro-rollback not only has high spatial overhead due to the additional storage needed to save

the snapshots as well as extra circuitry to detect the error and to execute the rollback, but also needs temporal overhead due to the halt and rewind in the instruction sequence.

Although high-level protection schemes have significant protection penalty in area, timing and power, they have been extensively adopted in space applications, as well as other mission-critical applications, such as life-saving device (pace makers, etc.) and safety-proof equipment (automobile breaking systems, etc).

With the technology scaling, soft error in mainstream applications at ground level is gradually becoming a serious concern. For mainstream applications, low-cost is the major objective while intermittent occurrence of transient errors is not necessarily disastrous. Hence, techniques such as TMR and micro-rollback are most probably not applicable due to their extremely high spatial and/or temporal overheads. The objective of SEU mitigation shifts from ensuring 100% SEU tolerance regardless of cost to achieving a desirable level of tolerance with reasonable cost.

The most fundamental method for hardening against SEU is to reduce charge collection at sensitive nodes. This can be accomplished in the material or device level by introducing extra doping layers to limit substrate charge collection [66], or using silicon-on-insulator (SOI) substrates to reduce the sensitive area of the device [67]. However, their invasive nature requires fundamental changes in the manufacturing process so the cost of their implementation is prohibitively high. On the contrary, techniques at the circuit level provide acceptable tradeoff between the implementation cost and the level of protection.

In cell-based ASIC design flow, standard cells are the basic circuit-building block. Many previous works have been focused on designing SEU-hardened standard cells by inserting spatial and/or temporal redundancies. For example, [10] proposed design of SEU-hardened latches and FFs based on a "Separate-Dual-Transistor (SDT)" structure that uses both temporal and spatial redundancies to increase its transient error tolerance. The detailed operation of an SDT DFF and its application will be further explored in Chapter 4. Designs of hardening cells using similar concepts and approaches can also be found in many other published works [12] [13] [14].

SEU-hardened standard cells provide the basic building block of SEU-tolerant digital circuits. In reality, the tight design constraints usually prohibit global application of these specially designed cells due to the associated timing/area overhead. Most realistically, SEU-hardened cells can be only used at selected locations. Technologies focusing on the optimization of hardening cell insertions can be found in many research works [32] [68].

SEU mitigation can also be achieved during the chip synthesis [33] since the logic implementation and gate sizing can significantly affect the generation and capturing of the transient glitches in the combinational netlist. The advantages of these gate-level techniques include: (1) the optimization does not necessarily incur design overhead; and (2) the application of the techniques can be merged into the existing chip physical design flow. However, similar metrics of SEU vulnerability have also to be developed to guide the SEU-aware synthesis process.

1.4.3 Limitations of the Previous Works

As the objective of SEU mitigation shifts from ensuring 100% SEU tolerance regardless of cost to achieving a desirable level of tolerance with reasonable cost, existing State-of-the-art EDA tools and previous design/testing methodologies are not geared towards handling transient interference in an economic and efficient way. Chip designers are left no choice but relying on empirical guidelines and manual efforts, resulting in lengthened design cycle and increased non-recurring engineering (NRE) cost; and the products are potentially highly susceptible to particle strikes during their field operation. This section lists some limitations of the previous works introduced above from a high-level viewpoint:

- Traditionally, attacking the reliability problem starts with separately analyzing each individual noise source. As chips become more complex and take on more functionality, one of the biggest challenges lies not in modeling the behavior of the chip itself, but rather in modeling the behavior of the noise sources. Many noise sources co-exist and interact with each other to aggravate their error effects. In addition, various types of process and environmental variations will add extra uncertainties to the chip reliability. Consequently, techniques that target on individual noise sources inevitably face great difficulties because the amount of work involved in analyzing all noise sources; further more, they can not accurately and quickly consider the compound effects, leading to overly optimistic analysis result.

- Traditional design, verification and test methodologies are predominantly deterministic in nature, such as vector-based simulation, static timing analysis, formal equivalency check and automatic test pattern generation. Ensuring circuit correct functionality and certain level of reliability primarily count on exercising different corner cases. However, SEU is causes by strikes of cosmic particles whose activities are totally random and unpredictable and can not be modeled deterministically. As result, traditional techniques are not capable of analyzing SEU effects and improving circuit SEU-tolerance.

- Traditional SEU mitigation techniques in mission critical applications have been relying on costly redundancy-based approaches because cost has been of secondary concern. As SEU become a reliability concern in cost-sensitive mainstream applications, existing techniques can not be directly applied due to the tight design constraint and budget.

- The majority of SEU analysis methodologies have been relying on dynamic simulation or intensive computation. Due to increasing chip complexity and tightened design requirements, they are no longer scalable or viable solutions.

- SEU analysis and mitigation have existed mainly as a secondary task and isolated from the design flow. The shortcomings are two-sided: on one hand, necessary SEU analysis can not easily executed in the early design phase to identify the potential vulnerability; on the other hand, the analysis results can not be directly utilized to guide the SEU mitigation efforts. The consequence is not only insufficient SEU protection to be built in the design, but also unnecessary

design resource and engineering cost.

- As has been widely accepted, circuit-level redundancy insertion is one promising solution to SEU mitigation. However, its applicability may be severely limited in low-power, high-speed and cost-effective IC product due to the associated penalty in timing, area and power. In order to achieve high reliability with acceptable cost, the redundancy insertion has to be judiciously applied. This requires accurate and convenient reliability metric as guidelines, which, unfortunately, has not been successfully developed in existing works.

1.5 Overview of Research Contributions

Facing the challenges stated above, it is imperative that revolutionary methodologies, techniques and flows be developed to facilitate the SEU analysis and mitigation, and the design and optimization of highly reliable nanometer circuit systems. This constitutes the primary objective of the research works presented in this dissertation.

1.5.1 Scope and Applicability

SEU-related research is an extremely widespread field so it is impractical to cover the entire space. Based on the previous discussions on the challenges and technology trend, the research work presented in this dissertation will be focused on SEU analysis and mitigation of random logics in products targeting the mainstream applications. The applicability of the methodologies and techniques is limited to static digital CMOS circuits.

The selection of this particular scope is due to three reasons. First, the severity of the transient error effects in this scope is becoming dominantly significant. Second, the quest for proper solutions in this scope is particularly challenging. Third, the need to successfully address the problem is becoming increasingly urgent. In this scope, viable solutions have to be low-cost, static and constraint-aware. In addition, they have to be compatible and integratible with the existing design flow.

Within the specified scope, "reliability" is exclusively used to refer to the "SEU tolerance" of a static digital CMOS circuit, i.e. the ability of the circuit to resist the transient errors caused by cosmic particle strikes and to maintain its functional correctness. Another term, "robustness", is used interchangeably with "reliability".

1.5.2 Basic Approaches

In the previous technology generation, reliability did not draw enough attention and was not included in the design sign-off metric. As a result, the manufactured VLSI circuits might have certain level of reliability defects, or "vulnerabilities", in the circuit structure that might lead to chip malfunction if attacked by particle strikes. As reliability has become a serious design concern, these vulnerabilities have to be identified and strengthened during the chip design phase. In order to improve the productivity and reduce the cost, these efforts have to be automated in an integrated design flow. This section overviews the ideas and approaches presented in the dissertation.

Fundamentally different from the existing approaches that set their viewpoints on one or more of the noise sources, the proposed approaches start the efforts by focusing on

the VLSI circuit itself. The inherent transient error tolerance of CMOS digital circuits is solely determined by their logical, structural and electrical properties and independent of the external and/or internal noise interferences. Hence, it is much more realistic to analyze the transient error tolerance and identify the most vulnerable regions during the early design phase, when the behaviors of the external noise sources are largely unknown.

Once these vulnerabilities have been identified, the circuit behavior in the presence of noise interferences can be accurately and quickly determined. Instead of estimating the impact of the potential noise source on the circuit behavior, a better approach is studying the impact of the inherent circuit tolerance on the behaviors of potential noise sources. Conceptually, the circuit is viewed as a transfer function that transforms any given noise description to a corresponding error rate. Once the transfer function is obtained, it can be applied to arbitrary noise sources to evaluate its specific error effect.

The next step is to utilize these analysis results to guide the efforts of improving the circuit reliability. Under these guidelines, circuit hardening techniques can be applied most efficiently and it is possible to achieve maximum improvement while minimizing the penalty. In digital circuits, reliability enhancing techniques need to be developed for both the combinational and sequential logics.

These three steps together form a complete reliability optimization flow. In order to maximize the efficiency, it is necessary to integrate the reliability analysis and enhancement efforts in a single framework. In addition, it is desirable to merge these efforts into the existing chip design flow, not only to avoid redundant works, but also to

maintain the level of required performance and cost.

In summary, *accuracy*, *cost*, *scalability* and *integration* are the key words that best describe the requirement for a viable and promising reliability optimization solution. These aspects motivate all the research works presented in the following chapters.

Taking one step forward, knowing that the existence of other types of variations and noise will unavoidably aggravate the transient error effects, this compound effect will be studied using two examples. The first one is the circuit vulnerability analysis that considers the interactions between transient error and signal integrity problems (such as crosstalk). The second is the impact of process variations on this particular type of environment variations. These efforts on the compound effects add another dimension to the unified reliability optimization framework.

1.5.3 Dissertation Outline

The rest of this dissertation is organized as follows:

Chapter 2 presents a “soft spot analysis” methodology. It is a static technique to identify the most vulnerable regions in the combinational circuits. The analysis is purely based on the electrical, timing and logic structure of the circuit without the necessity of being aware of the external noise disturbances. It discovers that a small portion in the design have significantly high vulnerability, i.e. noise at these so called “soft spots” will potentially cause high functional impact when suffering external attack. The result can be used to guide cost-effective reliability optimization of the combinational circuits.

Chapter 3 presents a “noise impact analysis” methodology. It is a static technique to identify the most vulnerable sequential elements in the circuits. The analysis is based on not only the circuit property, but also the knowledge of external noise information. It discovers that only a small percentage of all sequential elements have high probability of being affected by transient errors. The result can be used to guide the judicious insertion of SEU-hardened sequential elements.

Chapter 4 presents a “constraint-aware robustness insertion” technology. Based on the noise impact analysis result, using a cost-effective SEU-hardened standard cell, it is an optimization algorithm that automatically find the optimal scheme to protect the sequential elements under the given design constraints and budgets.

Chapter 5 presents a robustness enhancement technique. Based on the soft spot analysis results, it selects the most vulnerable combinational circuit nodes as the target of applying various circuit-tuning techniques to suppress the generation, propagation and capture of transient errors in the combinational circuits. As a result, the overall observable error rate can be significantly reduced.

Chapter 6 presents the modeling of transient error effects considering process variations. This work collectively considers two types of variability in nanometer digital circuits and demonstrates that the presence of process variation can greatly aggravate the environment variations; as a result, ignoring the process variation may lead to large analysis inaccuracy. It developed a statistical modeling of the generation and propagation of transient errors in the combinational circuits that can be used to accurately and quickly evaluate the transient error behavior.

Chapter 7 summarizes the major contributions of research works presented in this dissertation. By integrating all pieces of individual work in a unified and self-contained framework, a promising flow that realizes cost-effective and intelligent reliability optimization will be depicted. This chapter also provides the directions of future research.

Chapter 2. Soft Spot Analysis

Nanometer circuits are becoming increasingly vulnerable to interferences from multiple noise sources, including the radiation-induced soft errors. A desirable approach to ensure reliable functioning of chips is to first identify regions in the circuit that are most susceptible to multiple noise sources (called "soft spots"), and then to "harden" these soft spots using various techniques. This chapter presents a scalable technique to evaluate the circuit vulnerability. A "*softness*" of the circuit is defined as an important vulnerability measurement. Several key factors affecting the *softness* value are examined and an efficient Automatic Soft Spot Analyzer (ASSA) is developed to evaluate the *softness*. The proposed methodology provides guidelines not only to reduction of severe noise effects caused by aggressive design in the pre-manufacturing phase, but also to selective insertion of protection schemes to achieve high degree of on-line robustness. The quality of the proposed soft-spot analysis technique is validated by HSPICE simulation, and its scalability is demonstrated on a commercial embedded processor.

2.1 Introduction

As the feature size shrinking to nanometer scale, clock frequency reaching multi-GHz and supply voltage reducing to sub-voltage range, effects of various noise sources are becoming stronger than ever at the same time as the noise margin of the semiconductor device is significantly reduced. As a result, nanometer circuits are

becoming more vulnerable to various noise interferences. To make matters significantly worse, different noise sources and failure mechanisms tend to interact with each other to create compound effects. These compound noise effects exacerbate the difficulty in analysis and design of reliable digital circuit system.

Extensive research and technology developments have been devoted to study and ensure the reliability of nanometer chips. Analysis and optimization techniques for signal integrity issues such as crosstalk [3], IR-drop [4], ground bounce [5] and substrate noise [6] in System-on-Chips (SoCs) have been widely investigated. In particular, single-event-upsets (SEUs) caused by cosmic particles [7] severely impact field-level product reliabilities. Unfortunately, there have been few attempts to develop analysis techniques for the compound noise effects. Due to the unpredictable and transient nature of these noise effects, on-line detection and protection schemes become inevitable. However, blindly applying hardening techniques to the entire design incurs unacceptable design overhead. As the circuit size drastically increases, approaches based on dynamic simulations will not be able to be completed within reasonable time. In summary, a static technique that evaluates the impact of compound noise effects and identifies the regions of vulnerabilities is essential to the design of highly robust nanometer circuits.

This chapter introduces the “soft spot analysis”, an efficient methodology to address the challenges discussed above. Fundamentally different from traditional approaches that are focused on the behaviors of the random and transient noise interferences, ours sets its viewpoint on the design’s noise immunity. It is an intrinsic characteristic of the circuit that is not dependent on the external noise interferences, but

closely related to the timing, logic and electrical features of the design that can be conveniently analyzed during early design phase. Instead of trying to predict how and when different noise interferences are going to occur without enough information of the unpredictable sources, the proposed methodology evaluates how likely noise occurring at different nodes in the circuit will cause system malfunctioning. The methodology produces an overall vulnerability distribution in a design and discovers that the vulnerabilities of different regions greatly vary. This allows the designers to further investigate these most vulnerable nodes through focused noise analysis, to eliminate potential noise effects by limited design modifications, and to selectively apply on-line hardening techniques to prevent noise at these vulnerable nodes from causing severe functional impacts. As a result, the cost and effort of designing highly robust circuits can be dramatically reduced. The methodology is a static approach that does not require dynamic simulation or intensive computation so it can handle large complex circuits.

The rest of this chapter is organized as follows. Section 2.2 studies the compound noise effects caused by the simultaneous presence of multiple noise sources. Section 2.3 describes the static soft spot analysis in detail. Section 2.4 presents experimental results. Section 2.5 concludes this chapter.

2.2 Multiple Noise Source and Their Compound Effects

Multiple noise sources co-exist in a nanometer circuit due to various physical mechanisms. Examples include crosstalk noise caused by switching coupled wires, IR-drop caused by excessive simultaneous current draw from the resistive/inductive power grid, substrate coupling noise and environmental variations during the chip's operation

such as particle-strike-induced soft errors, etc. Each of these effects can individually cause circuit failures, and furthermore, different noise sources can also combine to magnify their effects, greatly increasing the possibility of errors.

In the example circuit shown in Figure 2-1, due to coupling capacitances Cx1 and Cx2 between the victim net (in the middle) and the two aggressor nets, when the input of INV1 remains at 0, a 0→1 transition at the input of INV3 or INV5 will result in a negative crosstalk glitch on the victim net, which will propagate to the input of a D-type flip-flop (DFFV) through INV2.

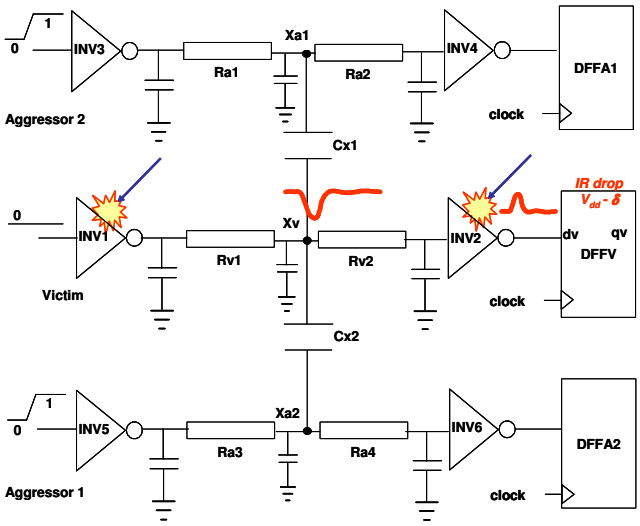


Figure 2-1 Compound Noise Effects: Example Circuit

However, as shown by the INV2 input (xv) and output (dv) voltages in Figure 2-2(a), the output glitch of INV2 is not strong enough to be captured as a stable logic error in DFFV even in the worst case when both aggressors switch simultaneously in the same direction, so the value in DFFV remains at the correct value "0" ("qv" in Figure 2-2(a)). However, if a particle strikes on the sensitive region of *either* INV1 (Figure 2-2

(b)) or INV2 (Figure 2-2 (c)) at a proper time, with all the other conditions unchanged, a "1" is latched in DFFV. In another case, the same crosstalk glitch that was not strong enough to change the state of DFFV is turned into a latched error by an IR-drop on the power line of DFFV (Figure 2-2 (d)). In all the failure cases, the error effects are the same – an observable error captured by DFFV.

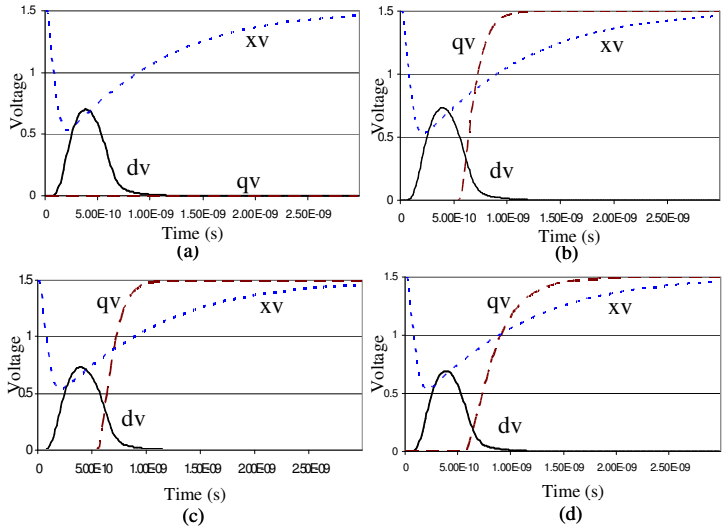


Figure 2-2 Compound Noise Effects: HSPICE Simulation

These experiments show that although the essential physical mechanisms of these noise effects are different, they can affect circuit behavior in a combined manner. A single noise source that is not strong enough might be potentially intensified by others. Therefore, a system exposed to multiple noise sources becomes more vulnerable. Furthermore, by simply examining the erroneous behavior, it is not possible to determine the failure mechanism. Therefore, methodologies trying to address effect of one single noise source while ignoring others may be not only overly optimistic but also inefficient.

2.3 Soft Spot Analysis

The random occurrences and complex physical mechanisms of different noise sources and their interactions depend on many factors that cannot be precisely determined until the product is manufactured and operating in a real environment. As a result, the behaviors and aggregated effects of various noise sources are extremely difficult to predict during the chip design. On the other hand, accurate information about the design itself, such as timing delay, logic paths and layout-extracted electrical characteristics, can be effectively analyzed during the design phase. These factors combine together to influence the design's noise-immunity, an intrinsic characteristic that is independent of the external noise disturbances. By studying these design features, it is possible to estimate the ability of different regions in a design to resist potential noise interferences, to predict the severity of functional impact if a noise were to occur, and to improve the circuit robustness by strengthening the potential vulnerable spots. The soft spot analysis has been developed based on these thoughts:

For each node N in a given digital circuit, a "*softness*" S_N is defined to measure its tendency to allow noise to propagate through it with enough strength and proper timing to eventually cause observable errors. An "observable" error refers to one that is captured by a memory element to become a stable erroneous logic value. The objective of the soft spot analysis is therefore to determine the magnitude of S_N for all circuit nodes and to identify a collection of "soft spots" as the nodes with high softness values. Obviously, S_N should reflect the collective contributions of the three masking effects discussed in Chapter 1: timing masking, electrical masking and logic masking.

2.3.1 Timing Masking

Timing masking means that noise can cause an observable error only if it is captured by a memory element. In order to be captured, it must arrive at the input of the memory element within a sampling window. For a DFF, the sampling window is bounded by the setup time (t_{su}) and the hold time (t_h) around the active clock edge, as shown on the right-hand side of Figure 2-3. An *effective noise window* TW_{eff}^{N} is used to measure the required time interval for noise at node N to reach a DFF within its sampling window – if a noise originates or arrives at node N before the start (after the end) of TW_{eff}^{N}, it will reach all DFFs before the start (after the end) of their sampling window, and will not be captured. As shown in Figure 2-3, TW_{eff}^{N} of a specific path p is bounded by a start time (t_{start}^{Np}) and an end time (t_{end}^{Np}), decided by the worst-case longest delay $(\Delta T^p)_{max}$ and best-case shortest delay $(\Delta T^p)_{min}$ from node N to the DFF through path p, respectively. If the clock period is T, it is easy to see that:

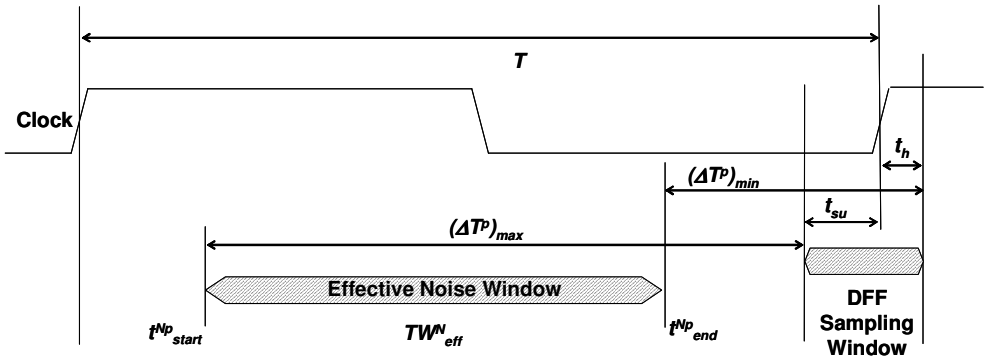


Figure 2-3 Calculating Effective Noise Window

$$t_{start}^{Np} = T - t_{su} - (\Delta T^{p})_{\max} \tag{2.1}$$

$$t_{end}^{Np} = T + t_{h} - (\Delta T^{p})_{\min} \tag{2.2}$$

Considering the fact that usually there are more than one DFF reachable from node *N* through many logic paths, the maximum (latest) end time and the minimum (earliest) start time among all paths are used to calculate TW_{eff}^{N}. Let *P* be the collection of all possible paths through node *N*, the effective noise window can be calculated as:

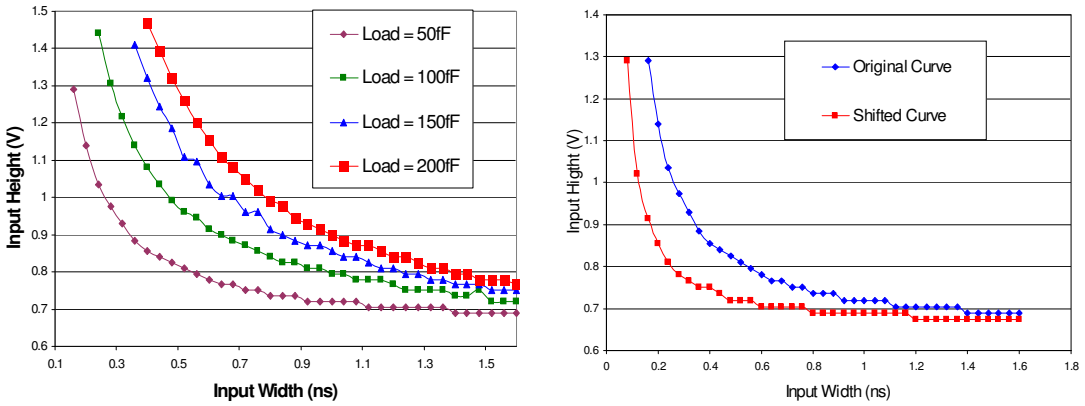
$$TW_{eff}^{N} = \max_{p \in P}\{t_{end}^{Np}\} - \min_{p \in P}\{t_{start}^{Np}\} \tag{2.3}$$

The TW_{eff}^{N} provides a measurement of the strength of the timing masking effect at node N – the larger the timing window, the more likely noise at this node will be able to overcome the timing masking effect.

2.3.2 Electrical Masking

Electrical masking means that noise must have enough duration and amplitude to propagate through multiple logic gates before being captured by a sequential element. The strength of the electrical masking effect of an individual gate can be described by its *noise rejection curves (NRCs)*. Figure 2-4(a) shows an example of noise rejection curves of an 0.18μm inverter driving different capacitive loads. The X and Y axes are the width and height of the input noise, respectively. The curve is drawn such that a glitch can propagate through the gate with enough strength (characterized by a pre-defined threshold voltage) only if its shape is in the region above the curve.

Noise rejection curves can be viewed as a representation of the ability of a gate to filter noise caused by any source. In reality, the nature of certain noise sources, like the radiation-effects, may be completely unknown until field operation, while the effects of some other noise sources, such as crosstalk, can be conveniently estimated based on the RC characteristics of the circuit. If the information of those "analyzable" noise sources can be built into the noise rejection curve, the curves reflect the "remaining" noise margin only to those unpredictable noise sources. This idea is called "curve shift".



(a) Noise Rejection Curve of an Inverter (b) Curve Shift due to Crosstalk

Figure 2-4 Noise Rejection Curve and Curve Shift

The example circuit shown in Figure 2-5 illustrates how crosstalk effect is built into the NRC by curve shift: gate G1 and G4 are identical inverters with the same load capacitance of 50fF, including the input capacitance of pin B of a two-input OR gate G3 (C_{inB}) and the wire capacitance (C_w), so they should be both represented by the NRC of 50fF shown in Figure 2-4(a). However, the existence of coupling capacitance (Cx) between the input of G1 and G2 will result in certain crosstalk effects, making G1 less resistant than G4 to noise of other type at its input. This indicates that using the same

NRC to describe the noise-tolerance of G1 and G4 is not appropriate: some glitches at the input of G1 which are below the original NRC may be able to propagate through and therefore should be relocated to the region above the original NRC. In another point of view, the NRC of G1 deviates from the original position and "shifts" toward the axes. If the severity of the crosstalk can be estimated in terms of its worst-case magnitude and duration, the original NRC is then correspondingly shifted, to obtain the "shifted curve", as shown in Figure 2-4(b).

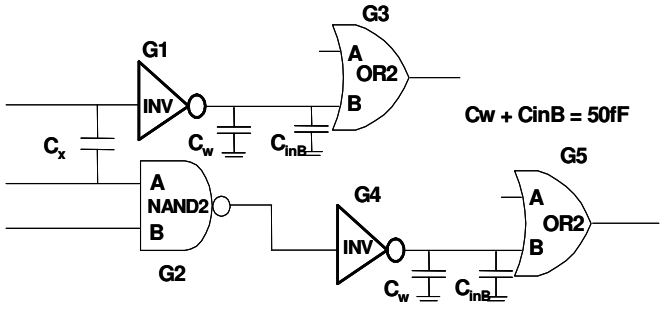


Figure 2-5 Circuit Example - Curve Shift

In the noise rejection curve graph, define the *noise propagation ratio* R_e^N as:

$$R_e^N = \left(\frac{Area_{sensitive}}{Area_{immune}}\right)_{NRC} \tag{2.4}$$

where $Area_{sensitive}$ is the area of the region above the curve (sensitive region) and $Area_{immune}$ is the area of the region under the curve (immune region). When calculating the areas, the upper bound on the Y-axis is set to be the maximum possible input glitch height, which is the power supply voltage and the upper bound is set to be the maximum possible input glitch width, which can be assumed to be the clock period. R_e^N is used to

measure the strength of electrical masking effect: the higher ratio R_e^N, the more easily a glitch can overcome the electrical masking effect to propagate through the gate.

For cell based designs, the standard cell library can be pre-calibrated using off-line HSPICE to create NRC database of all gates with different capacitive loads. Then for a given design, the load capacitance of each node is first obtained from layout-extracted RC information to retrieve the proper curve from the database. This retrieved curve may be then "shifted" according to preliminary analysis results of certain noise sources. For example, crosstalk effects may be estimated using existing techniques such as the extended 2-π model [16], given the detailed RC parasitic and coupling capacitances. Finally R_e^N is computed as the area ratio in the modified curve. This process is illustrated in Figure 2-6**.** Curve shifts caused by other mechanisms can be similarly considered, given specific design and process information.

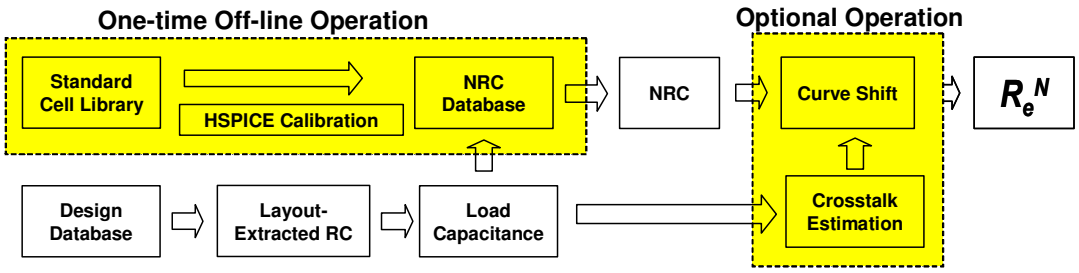


Figure 2-6 Calculating R_e^N in Cell-Based Designs

An obvious advantage in the above flow is that cell calibration is done only once for a given library and can be used on all circuits implemented in the same library. As a result, one can fully make use of the accuracy of HSPICE without repeatedly suffering from its time-consuming nature.

2.3.3 Logic Masking

Logic masking refers to the effect that noise ceases to propagate through a gate whose output is solely determined by its other inputs. Depending on the logic structure, the chances of noise occurrences at different nodes to survive multiple levels of logic gates are different. Complete determination of the logic masking effect requires exhaustive exploration of the entire input vector space and prohibitively long dynamic simulation time. An efficient logic path tracing technique has been developed as an alternative to estimate the *propagation probability* P_{prop}^{N}, the probability of a glitch propagating from node *N* to all reachable DFFs through legitimate logic paths.

The algorithm consists of two steps, each using the breadth-first search (BFS) algorithm [17] to go through the gate-level netlist of the design.

The first step uses a forward BFS, starting from the primary inputs (PIs), to derive for each node *N* the *logic probability*, the probability of being logic "1" (or "0"), denoted by $Pr^{N}(1)$, ($Pr^{N}(0) = 1\text{-}Pr^{N}(1)$). The example shown in Figure 2-7 best illustrates how to calculate the logic probabilities of all nodes. Assuming the logic probabilities for all inputs ($Pr^{A}(1)$, $Pr^{B}(1)$, $Pr^{C}(1)$, and $Pr^{D}(1)$) are 1/2, $Pr^{E}(1)$, $Pr^{G}(1)$, $Pr^{F}(1)$, and $Pr^{H}(1)$ are calculated in order as the netlist is searched, and listed above the circuit. To start the process, the logic probabilities at the PIs should be known. It can be obtained by recording the statistics of logic zeros and ones applied to the PIs during the functional verification. If no such information is available, a good approximation is to assume that logic "1" and "0" have equal probabilities at all PIs.

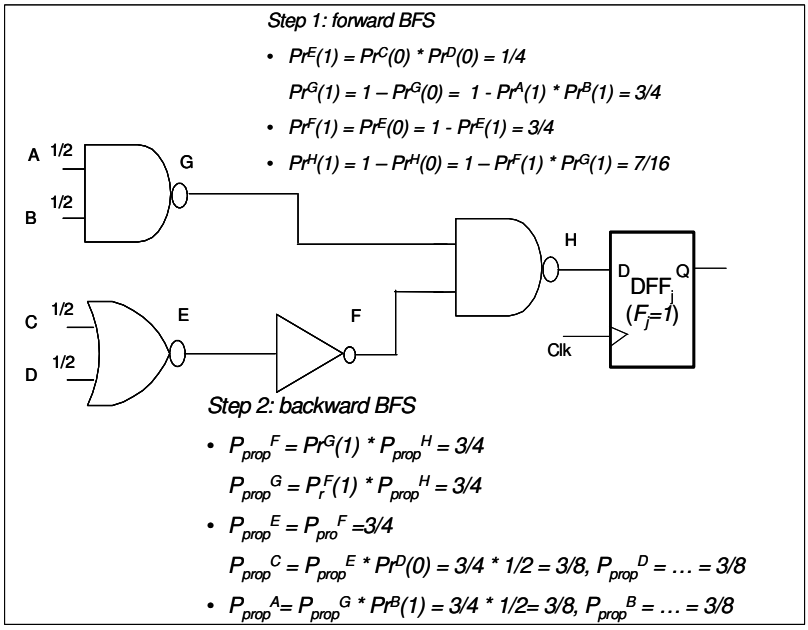


Figure 2-7 Example Circuit: Calculating Logic Masking Factor

The second step uses a backward BFS, starting from the input nodes of the DFFs, to calculate the propagation probability P_{prop}^N at each node. As the netlist is searched backward, the value at an input of a gate M (a descendant node) is calculated from the value at the output node of gate M (the parent node) and the probability of all the side inputs carrying non-controlling values of gate M that is determined from the logic probabilities at the side inputs obtained during the forward BFS. In the same example shown in Figure 2-7, P_{prop}^H is set to 1 because node H is the input of a single DFF. Next, node F and G are reached during the BFS. P_{prop}^F is calculated as P_{prop}^H multiplied by $P_r^G(1)$, as the non-controlling value of a NAND gate is "1". P_{prop}^G is similarly calculated at the same time. As the search continues, values at node E, A, B and then node C, D are calculated in turn. The calculations and results are listed under the circuit. If a node has

multiple parents (multiple fan-outs), the values derived from all parents are added together to give the cumulative propagation probability at the node.

In reality, all DFFs in a design may not be of equal functional significance. The designers may assign the j^{th} DFF a functional weighting factor F_j based on design-specific knowledge. For example, a DFF that stores a crucial control signal such as global reset, clock gating enable or interrupt status, should be assigned a higher weight, indicating an observable error latched in this DFF will have higher functional impact. As a result, the propagation probability of the input to the j^{th} DFF will be equal to F_j and the backward BFS will start from different DFFs with different weights.

2.3.4 Evaluating *Softness* and Identifying *Soft Spots*

The softness S_N should be a function of the timing factor (TW_{eff}^{N}), electrical factor (R_e^{N}) and logic factor ($P_{prop}{}^{N}$). While S_N may have many possible analytical forms, if TW_{eff}^{N}, R_e^{N} and $P_{prop}{}^{N}$ are considered to contribute independently, S_N can be expressed as:

$$S_N = W_N * \left(TW_{eff}^{N} * R_e^{N} * P_{prop}{}^{N} \right) \tag{2.5}$$

In equation (2.5), W_N is an optional application-specific weighting factor at node N, default to be 1, for the designers to convey design-related knowledge. This weighting factor provides additional flexibility and controllability to the proposed methodology.

An automated flow called “Automatic Soft Spot Analyzer” (ASSA) has been developed, as shown in Figure 2-8, to implement computation of softness of all circuit nodes. To execute, ASSA first uses a “Library Calibration Engine” to generate a noise

rejection curve database for the cell library. Note that this step only needs to be performed once for each library, after which the database may be read in from storage when analyzing a given design. Next, timing, electrical and logic factors are evaluated using information in the design database (gate-level netlist, timing, physical layout, extracted RC, *etc*). ASSA then calculates S_N and provides a *softness* distribution as its output. From this distribution, a set of "soft spots" can be identified as nodes with high softness values. In addition, optional inputs including PI logic probabilities, DFF functional weighting factors and overall weighting factors may also be provided to improve the accuracy and validity of the results.

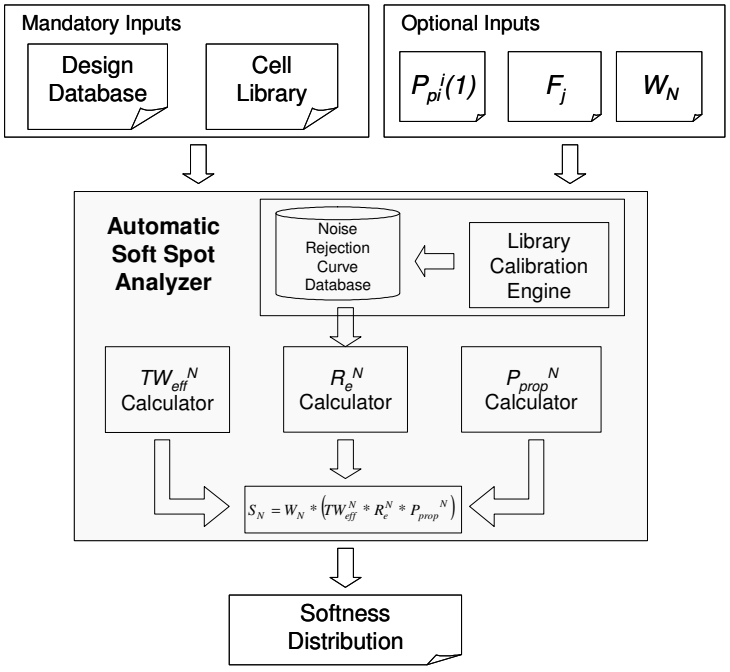


Figure 2-8 Automatic Soft Spot Analyzer (ASSA)

The selection of soft spots according to the softness distribution is closely related to the affordable design cost. A cost metric is needed to decide a threshold of softness

value S_{th}: the lower the threshold, the more nodes are identified as soft, the more cost must be paid to analyze, revise and protect these soft spots and the higher level of robustness will be achieved accordingly. For mission-critical applications, S_{th} should be set low such that a larger portion of nodes will be marked as soft spots and will be made highly noise immune at higher cost; for cost-sensitive mainstream applications, in order to save cost, S_{th} should be set higher so only the nodes that are most likely to cause the largest functional impart if being affected will be considered for further robustness optimization. The determination of the cost metric and robustness optimization are topics of ongoing research that is beyond the scope of this work.

2.4 Experimental Results

This section will demonstrate that the proposed methodology is not only able to accurately captures the most vulnerable nodes in a circuit, but also able to be applied to large systems as its runtime is almost linear to the number of circuit nodes.

2.4.1 Accuracy and Efficiency

The proposed method was applied to four circuits to evaluate its quality and speed by comparing with accurate SPICE fault simulation. Due to speed limitation of SPICE, only small circuits can be used. Two of the four circuits are basic blocks in many digital designs (ADDER: 4-bit adder and DEC: 4-bit decoder), the other two (Xt1 and Xt2) are random logics extracted from a commercial processor (Xtensa™ from Tensilica [18]). All circuits are combinational blocks with registered primary outputs and are synthesized by Synopsys DesignCompiler™ using a 0.18μm cell library; Cadence SiliconEnsemble™ is

used for generating physical layout; Mentor Graphics Xcalibre™ is used to extract RC networks; and Synopsys PrimeTime™ is used for static timing analysis.

In each experiment, the SPICE netlist is extracted from physical layout and contains RC information, including coupling capacitances, so crosstalk effects may be observed. Preliminary SPICE simulation results show that although crosstalk effects exist at many nodes, none of them is strong enough to cause functional errors. In addition, transient glitches of random shape and timing are injected into the circuit. This transient glitch can be used to model various noise effects such as erroneous logic switch due to particle strikes on the transistor's sensitive region.

Since the goal is to study the vulnerability of individual circuit node, the SPICE simulation is focused on a specific node at a time – a number of input vectors are applied to the circuit while transient glitches are injected on a single node. The number of observable errors caused by the injection on each node is counted separately and used as a measurement of the "simulated softness". These results were compared with the softness values computed using ASSA engine. During the computation, the input logic probability is obtained from the actual statistics of SPICE simulation input vectors; the other two optional inputs (F_j and W_N) are set to 1 for the lack of application-specific knowledge.

Table 2-1 shows some statistics about the experiments. Row 1 is the area and row 2 is the internal node count of the sample circuits. Row 3 and 4 show the number of nodes on which transient faults have been injected and the number of input vectors used in simulation. Row 5 compares the runtime of ASSA to that of SPICE in terms of the average evaluation time per node. It can be seen that speedup factors (shown in row 6) at

the order of 10^3 were achieved by ASSA over HSPICE. Furthermore, as the circuit complexity increases, HSPICE simulation speed decreased drastically so tradeoffs had to be made between precision and runtime. For example, the size of DEC is only 2.3 times of the size of ADDER, but the number of chosen nodes and simulated vectors had to be reduced to 1/2 and 1/4, respectively, in order to finish simulation in comparable time. On the contrary, ASSA shows constant analyzing time for circuits of similar complexity (ADDER, Xt1 and Xt2) while exhibits performance improvement as the circuit size increases (DEC).

Table 2-1 Sample Circuits and Simulation Time

		ADDER	DEC	Xt1	Xt2
1. Area (Library Unit)		1107	2488	865	995
2. Node Count		89	210	74	59
3. No. Simulated Nodes		42	20	22	37
4. No. Input Vectors		512	128	512	256
5. Runtime (sec/node)	**HSPICE**	12,062	19,097	9,548	5,230
	ASSA	4.65	2.45	4.14	4.88
6. Speedup Factor		$2.6\text{x}10^3$	$7.8\text{x}10^3$	$2.3\text{x}10^3$	$1.1\text{x}10^3$

The calculated softness was compared with the simulated softness node by node, as shown in Figure 2-9. The indices of the simulated nodes are shown on the x-axis and the normalized softness values of selected nodes are shown on the y-axis, where the "ASSA" and "HSPICE" curves show the ASSA-calculated and the SPICE-simulated values, respectively. It can be seen that ASSA not only correctly captured the most vulnerable nodes but also provided a distribution of softness among all simulated nodes that matched simulation results very well: the nodes with high calculated softness values indeed cause more functional errors when being noise injected.

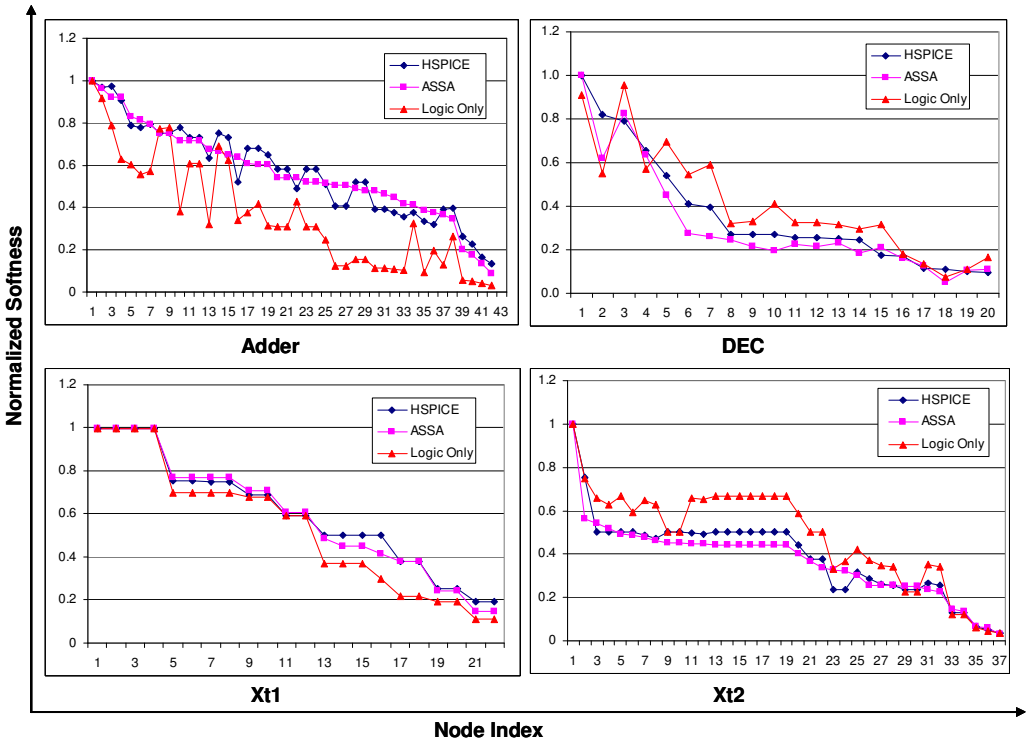


Figure 2-9 ASSA Results Compared with HSPICE Simulation

2.4.2 Improved Accuracy via Curve Shift

The ASSA calculation results shown in Figure 2-9 were obtained without considering curve shift caused by crosstalk. As discussed, curve shift can be utilized to improve the accuracy by building analyzable noise information into the noise rejection curves. Circuit DEC was chosen to demonstrate the effect of curve shift because it is relatively large and strong crosstalk effects can be observed at some of the internal nodes. In Figure 2-10, the third curve marked as "ASSA With Xtalk" is the softness values calculated by ASSA after considering curve shift caused by crosstalk. The discrepancies between the simulation and calculation results diminish as the calculated values increase

for most of the nodes. Especially for node 2, where the worst case crosstalk is estimated to be 0.25V, the calculated softness improves form 0.62 to 0.84, which is very close to the simulated value (0.82). However, results of some nodes (such as node 3 and 4) indicate that considering crosstalk effect make the calculation more pessimistic. This is due to the fact that the crosstalk estimation only gives the worst case values, which might not occur during simulation.

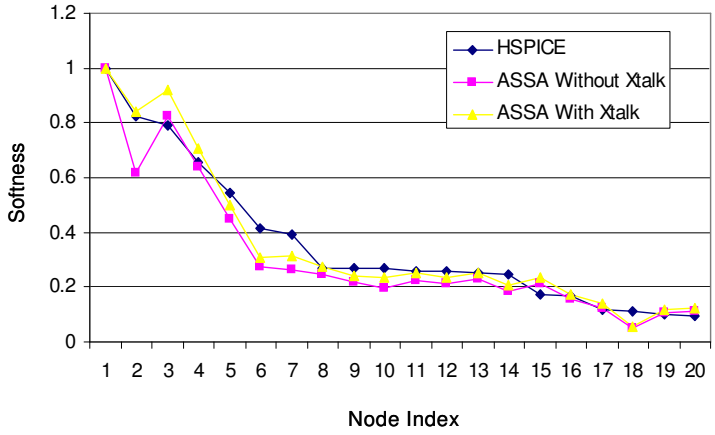


Figure 2-10 Effect of Considering Curve Shift Caused By Crosstalk

2.4.3 Scalability and Softness Distribution

To demonstrate its scalability, ASSA was applied on the Xtensa™ processor [18], a commercial state-of-the-art configurable and extensible RISC processor. The experiments were conducted on a large logic module EX with 97 registered output ports, 338 input ports and 3156 internal nodes. EX is particularly challenging because of its large number of inputs, unbalanced logic paths and strong crosstalk effects due to aggressive layout scheme. It is essential to identify and fix potential vulnerable spots and

provide low-cost on-line protection schemes in early design phase. Simulation-based methods are not applicable because of the design complexity. Soft spot analysis, on the other hand, promptly was able to finish within reasonable time. Table 2-2 shows a breakdown of time taken in each step. Note that the processing time of each node (2.55s) is comparable to the design DEC (2.45s), indicating that the methodology is able to scale approximately linearly as the design complexity increases.

Table 2-2 ASSA Runtime on Circuit EX

Operation	Time Taken
Calculating Electrical Factor R_e^N	52min*+
Calculating Logic Factor P_{prop}^N	2min
Calculating Timing Factor TW_{eff}^N	78min++
Calculating *Softness* S_N	2min
Total Time	134min
Process Time Per Node	2.55 sec
**: Library calibration time not included.* *+: Including layout and RC-extraction time.* *++: Including static timing analysis time.*	

It is worth mentioning that the majority of long processing time for calculating R_e^N and TW_{eff}^N is due to layout, RC extraction and static timing analysis. As all these operations are inevitable steps in any VLSI design flow, it will not require much additional time/effort when integrated with standard design flow.

It has also been validated that the vulnerabilities of different nodes vary greatly. Figure 2-11(a) shows the *softness* distribution of all nodes: the x-axis is the normalized *softness* in logarithm scale (normalized to 10,000 for convenience of depiction) and the y-axis is the number of nodes with different *softness* values. It clearly demonstrates the non-uniform *softness* distribution among all nodes: only about 0.7% (22) has *softness* larger

than 10% of the maximum value, and another 18.6% (587) has *softness* larger than 1%, whereas the *softness* values of the other 80.7% of all nodes are at least 2 orders of magnitude lower than the maximum value. Figure 2-11(b) shows the actual softness distribution among all nodes. The selection of soft spots is also illustrated in this example: if S_{th} is set to be 10% of the maximum softness value (S_{max}), only 22 spots are categorized as soft spots, whereas if S_{th} is set to be 1% of S_{max}, a total of 609 nodes are categorized as soft spots.

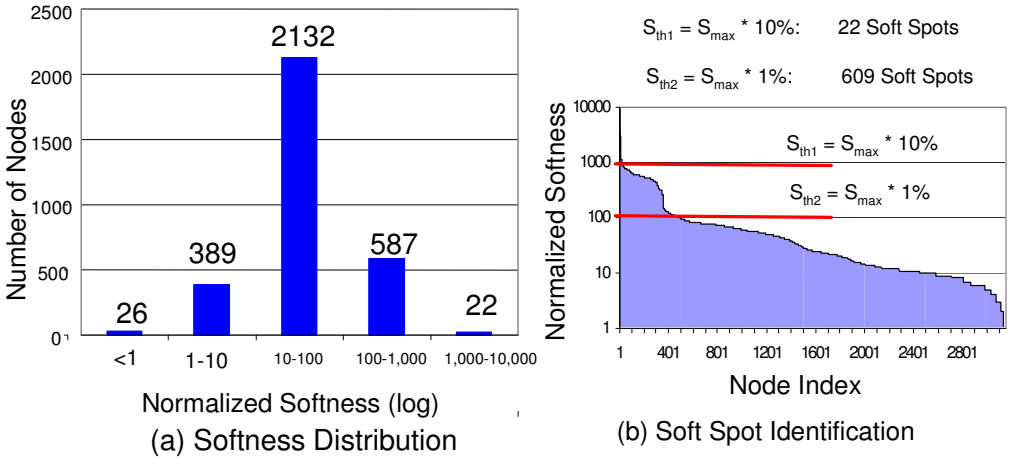


Figure 2-11 Softness Distribution and Soft Spot Identification in Design EX

This unbalanced softness distribution is crucial in efficient, low-cost robust circuit design. During the pre-manufacturing design phase, accurate but time consuming analysis can be subsequently performed on the identified most vulnerable spots. If the potential severe noise effects are caused by aggressive design, design modifications can be done. Furthermore, this methodology provides a guideline to selectively apply on-line error detection and protection scheme to the spots that are most likely affected by transient

errors during its lifetime, so a high degree of on-line robustness can be achieved with low design overhead.

2.5 Conclusion

In this chapter, a high-quality, efficient soft spot analysis methodology is proposed to identify the most vulnerable spots in a design exposed to multiple noise sources by evaluating the intrinsic noise immunity of a circuit. The quality of analysis result has been validated with extensive SPICE simulations. The soft spot analysis methodology was also applied to a large block in a commercial configurable processor to demonstrate its efficiency and scalability in analyzing complex designs. The proposed methodology is the key first step toward design of low-cost, highly robust nanometer circuit systems.

Soft spot analysis is targeted at locating the vulnerabilities in the combinational circuits. The next chapter will introduce a technique of evaluating the vulnerabilities of sequential elements.

2.6 Acknowledgement

Chapter 2 is based on material in the published paper: Chong Zhao, Xiaoliang Bai, Sujit Dey, "Soft Spot Analysis: A Scalable Methodology Targeting Compound Noise Effects in Nano-meter Circuits", IEEE Design & Test of Computers, Volume 22, Issue 4, July-Aug. 2005, pp. 362-375, and material in the published paper: Chong Zhao, Xiaoliang Bai, Sujit Dey, "A Scalable Soft Spot Analysis Methodology for Compound Noise Effects in Nano-meter Circuits," in Proceedings of 41st Design Automation Conference (DAC),

pp. 894-899, June 2004, San Diego, California, USA. The dissertation author was the primary investigator and author of this paper.

Chapter 3. Noise Impact Analysis

In the previous chapter, a static "soft spot analysis" technique to evaluate the vulnerability of the combinational circuit has been presented. In digital circuit, sequential elements such as flip flops (FFs) play crucial role in circuit transient error tolerance. In this chapter, a novel "noise impact analysis" technique will be presented to evaluate the vulnerability of the sequential elements. Different from the soft spot analysis, which does not consider any information of the external environment, the noise impact analysis also considers the probabilities for a transient glitch to occur at different circuit locations. With both the circuit and the transient noise abstracted in the format of matrices, the circuit-noise interaction is modeled by a series of matrix transformations. During the transformation, factors affecting the propagation and capture of transient errors are modeled as matrix operations. As the end of the transformation, the probability of a FF capturing transient noise originated inside the combinational logics is computed, called the "noise capture ratio" of the FF. The noise capture ratio of all FFs forms a reliability metric to measure the observable error rate and gauge the circuit's tolerance to radiation-induced transient errors.

3.1 Introduction

The need for cost-effective robust circuit design mandates the development of efficient reliability metrics that include SEU analysis and avoidance. Redundancy

insertions at various levels have been traditionally adopted to ensure a high degree of SEU tolerance in space or mission critical applications [23], where cost and design efforts are only the secondary concern compared to system reliability. Unfortunately, the associated penalties (200%~300%) are unacceptable for cost-sensitive mainstream applications. It has been proposed [14] to use partial redundancy insertion to reduce protection cost. In order for the partial redundancy insertion to be efficient and economical, guidelines are needed to identify the location where the insertions should be made (i.e. the most vulnerable circuit elements) and could be made (i.e. overhead allowed by the design constraints).

In digital circuit, the D-type flip-flop (DFF) plays a crucial role in the circuit transient error tolerance because only transient errors latched in the DFFs will actually cause stable error to affect the circuit functionality. In this chapter, a static technique of evaluating the transient error tolerance of all DFFs will be presented. It targets one particular type of SEU, which is the single-event-transient (SET), a transient glitch caused by a particle strike on a combinational node. As is discussed in Chapter 1, a single-event-transient (SET) has to propagate to a DFF with enough strength within a specific timing window in order to be captured so the error rate is determined by the likelihood of SETs originated in the combinational circuits becoming observable errors, which depends on several factors, including its generation, propagation and capture.

1) "SET generation" refers to the likelihood that a particle strikes a logic gate and strength of the result transient pulse. The occurrence of this event depends on the cosmic particle activities as well as the effective cross-section of the sensitive

region. The pulse amplitude and duration also depends on the electrical characteristics of the affected gate.

2) "SET propagation" refers to the behavior of the resulting transient traveling within the combinational circuit toward the DFFs, which is affected by the three masking effects. Their existence and strength are independent of the external environment but are closely related to the circuit structure, and therefore, can be statically estimated to some degree of accuracy.

3) "SET capture" refers to the capability of the destination DFFs latching the incoming transient pulses from the combinational logic.

The technique presented in this chapter is called the "noise impact analysis". As shown in Figure 3-1, it consists of four components. First, in the "Transient Noise Modeling", a matrix representation, "Noise Probability Density Function (NPDF)", is used to probabilistically describe the SET generation. Second, the "Circuit Noise-Immunity Analysis" quantifies the three masking effects. Third, a "NPDF transformation" process is used to model the impacts of the circuit noise immunity on SET distribution and propagation. During the transformation, the circuit is represented as a netlist of matrices and the masking effects are modeled as operations on the NPDF matrices. Finally, when the transformation proceeds to the input of a DFF, a "Noise Capture Ratio" is computed from the incoming NPDF. As will be discussed, this noise capture ratio properly measures the SET error rate. It is discovered that the SET capture probabilities of different DFFs greatly vary. This is crucial in search for efficient yet economical solutions to improving the circuit SET tolerance. Since the proposed methodology does

not require dynamic simulation or intensive computation, it is fast and can handle large complex designs.

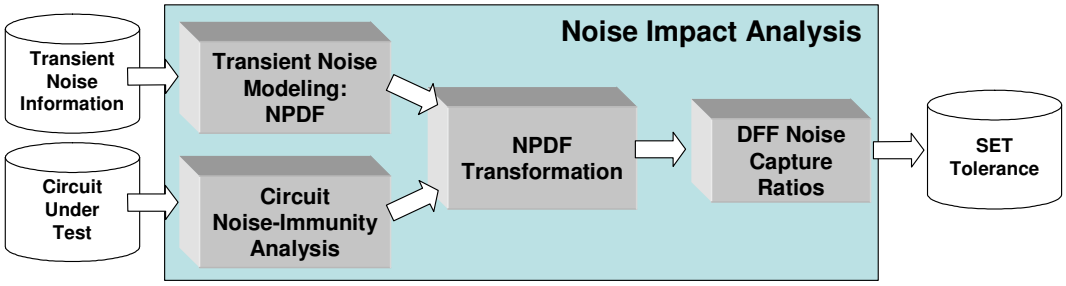


Figure 3-1 Noise Impact Analysis Framework

The rest of the chapter is organized as follows: section 0 models the random SET generations using the NPDF matrices; section 3.3 investigates the intrinsic noise immunity of digital circuits; section 3.4 describes the NPDF transformation and the noise capture ratio calculation; section 3.5 presents experimental results and proposes potential applications; and section 3.6 concludes the chapter.

3.2 Single-Event-Transient in Static CMOS Digital Circuits

As introduced in Chapter 1, a particle strike on a combinational gate may cause a transient voltage pulse at the gate output. This transient pulse is usually modeled as a square-shaped glitch *[w,h]* (Figure 1-3). The height h ($0<h\leq V_{dd}$, where V_{dd} is the supply voltage) is the maximum deviation from the nominal voltage level; the width w ($0<w\leq T$, where T is the clock period) is measured at a pre-defined threshold voltage.

Due to the random nature of the cosmic particle activities, multiple occurrences of the transient glitches can be best described probabilistically. This randomness is modeled

using a "Noise Probability Density Function (NPDF)": for each circuit node *N*, *NPDF(N,t,w,h)* is defined as the probability for a glitch *[w,h]* to occur at time *t* ($0 \le t \le T$). This distribution depends on the particle activities in the environment, which exhibit strong temporal, geographical and altitude dependence. It also depends on the incident angle, contact point and the sensitive volume of the material [24]. Theoretically, if the energy spectrum of the particles and the shape of the resulting transient pulses as a function of the particle energy are known, it is possible to determine the shape distribution.

W \ H, P_{ij}(‰)	0	1	2	3	4	5	6	7
0	22	16	17	15	13	17	12	18
1	21	18	15	12	19	14	15	17
2	11	14	14	13	20	20	17	19
3	11	12	16	9	18	10	16	16
4	16	19	14	13	15	8	13	16
5	18	15	14	12	18	18	20	11
6	18	13	11	14	18	16	13	14
7	13	23	21	25	16	20	14	14

(a) NPDF Example

W \ H	0	1	2	3	4	5	6	7
0	0,0	0,0	1,1	1,4	1,7	1,7	2,7	2,7
1	0,0	0,0	1,2	1,6	2,7	2,7	3,7	3,7
2	0,0	0,0	1,2	2,7	3,7	3,7	4,7	4,7
3	0,0	0,0	2,3	3,7	3,7	4,7	5,7	5,7
4	0,0	0,0	2,3	4,7	4,7	5,7	6,7	6,7
5	0,0	0,0	2,4	4,7	5,7	6,7	7,7	7,7
6	0,0	0,0	3,4	4,7	6,7	7,7	7,7	7,7
7	0,0	0,0	3,4	5,7	6,7	7,7	7,7	7,7

(b) NPM Example

Figure 3-2 Examples of NPDF and NPM

Assuming SETs observe uniform temporal distribution, i.e. they may happen any time within a clock cycle with equal probability, *NPDF(N,t,w,h)* is reduced to *NPDF(N,w,h)*. A given *NPDF(N,w,h)* is constructed in a matrix format as follows. First the shape of an SET *[w,h]* is quantized to *[i,j]* with respect to the clock period (*T*) and the supply voltage (V_{dd}), where $i=floor(w/T*M)$, $j=floor(h/V_{dd}*M)$, and *M* is a designated quantizer. Then P_{ij}, the probability of a glitch *[i,j]* occurring on node *N*, is recorded in the

NPDF matrix. Figure 3-2(a) shows an example of an NPDF matrix with $M=8$. The unit of P_{ij} is 1/1000 ("‰") and the matrix is normalized: $\sum_{[i,j]\in NPDF} P_{ij} = 1$.

As an example, P_{35} = 10(‰) means 10 out of 1000 glitches at this node have a width w ($3/8*T \le w < 4/8*T$) and a height h ($5/8*V_{dd} \le h < 6/8*V_{dd}$). The resolution can be improved at the cost of more memory usage.

3.3 Transient Error Tolerance of Digital Circuits

The three masking effects mentioned in section 1.3.1 (timing masking, electrical masking and logic masking) determines the inherent transient error tolerance of digital circuits. This section discusses how to numerically measure the strength of the three masking effects. Although the basic principles used here are similar to that presented in the previous chapter, the measurement detail is different in order to achieve higher accuracy and to facilitate the execution of the noise impact analysis.

3.3.1 Timing Masking

To become an observable error, a glitch needs to arrive at a DFF within a sampling as shown in Figure 3-3(a). This imposes a timing requirement for a glitch on individual circuit node, or the "effective sensitive window" derived in the previous chapter: for a specific path p that goes through node N, the sensitive window has a start time ($t_{start}^{Np} = T - t_{su} - (d_p)_{\max}$) and an end time ($t_{end}^{Np} = T + t_h - (d_p)_{\min}$), as shown in Figure 3-3 (a), where T is the clock period. In general, node N is located on multiple logic paths, so the sensitive window is bounded by the latest end time and the

earliest start time among the collection of all timing paths *P*: $t^N_{start} = \min_{p\in P}\{^{Np}_{start}\}, t^N_{end} = \max_{p\in P}\{^{Np}_{end}\}$.

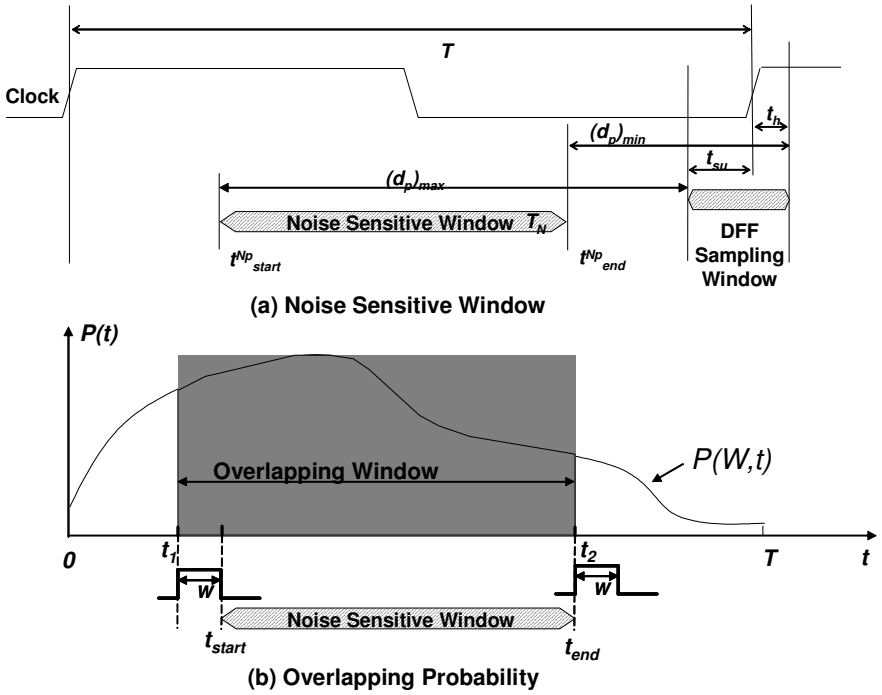


Figure 3-3 Sensitive Window and Overlapping Probability

Obviously, the probability for an SET at node *N* with width *w* to be sampled by DFFs is equal to the probability that it overlaps with the sensitive window, $P_t^N(w)$. As shown in Figure 3-3(b), for an overlap to occur, the starting time of the glitch must be earlier than t_{end} and later than t^N_{start}-*w* (0 if $t^N_{start} < w$). During any clock period *[0,T]*, if the probability for an SET of width *w* to start at time t is *P(w,t)*, the overlapping probability can be calculated as:

$$P_t^N(w) = \int_{t_1}^{t_2} P(w,t)dt \tag{3.1}$$

where $t_1=max\{t^N_{start}-w,0\}$, $t_2=t^N_{end}$. Since a uniform temporal distribution is assumed, from the normalization condition for SET of width w: $\int_0^T P(w,t)dt = 1$, which gives $P(w,t) = 1/T$, therefore:

$$P_t^N(w) = \begin{cases} t^N_{end}/T, & \text{if } w \geq t^N_{start} \\ (T_N + w)/T, & \text{if } w < t^N_{start} \end{cases} \tag{3.2}$$

where T_N is the size of the sensitive window: $T^N = t^N_{end} - t^N_{start}$

The overlapping probability $P_t^N(w)$ converts the sampling window of the destination DFFs into a local timing constraint on the transient glitch at node N.

3.3.2 Electrical Masking

A logic gate can change the shape of SETs presented at its input and filter out those without enough duration and amplitude. The shape of output glitch ($[w_{out}, h_{out}]$) is a function of the shape of the input glitch $[w_{in}, h_{in}]$ as well as the gate type ("*GateType*"), affected input pin ("*GatePin*"), the glitch transition type ("*TransType*", either positive or negative), and loading condition ("C_{load}"):

$$[w_{out}, h_{out}] = f(w_{in}, h_{in}, GateType, GatePin, TransType, C_{load}) \tag{3.3}$$

For standard cell based design flow, this function for all library cells can be calibrated via accurate off-line transistor-level. In this work, after the library cells are pre-calibrated, they are further quantized and saved in a matrix format called "Noise Propagation Matrix (NPM)", constructed in the same manner as the NPDF. Figure 3-2(b)

shows an NPM for positive-transitioned glitches at the input of an inverter driving a load of 0.02pf. The quantizer M is also 8. The entries are 2-tuples $[w_i,h_j]$ representing the width and height of the output glitch when the input glitch has quantized shape $[i,j]$. Using NPMs, when the shape of a glitch at a particular input pin of a gate is given, the shape of the output glitch can be retrieved from the library.

Multiple NPMs are needed to completely characterize one gate. For example, a 2-input AND gate needs 4 NPMs for every load condition: positive and negative transition at A when B=1; positive and negative noise transition at B when A=1. The notation *NPM(GateType, GatePin, TransType,* C_{load}*)* is used to distinguish multiple NPMs of the same gate, so *NPM(AND, A, 0→1, 0.02pf)* refers to the NPM for a positive glitch at pin A of an AND gate driving a loading of 0.02pf. In the analysis, combinational gates need to be represented by several of its NPMs according to various conditions and model the entire circuit as a netlist of NPM matrices.

3.3.3 Logic Masking

Glitches at one input of a gate can not reach the output if the output is determined by the controlling values on other inputs of the gate. Therefore, only glitches on a "sensitized path" (defined as a logic path from a node to an endpoint DFF on which all side inputs of all logic gates have *non-controlling* values) will eventually be able to logically reach a DFF. For each input node N of a gate in the circuit, a "Surviving Probability" P_l^N is defined as the probability of no side input carrying controlling values. To compute P_l^N, it is necessary to know the probability of logic 0 and 1 on all circuit nodes. Using similar logic tracing technique presented in section 2.3.3, the circuit can be

traced by breadth-first-search (BFS) and iteratively compute the logic probability at each encountered node. The surviving probability P_l^N will be used in the analysis to evaluate the logic masking effect of an individual circuit node.

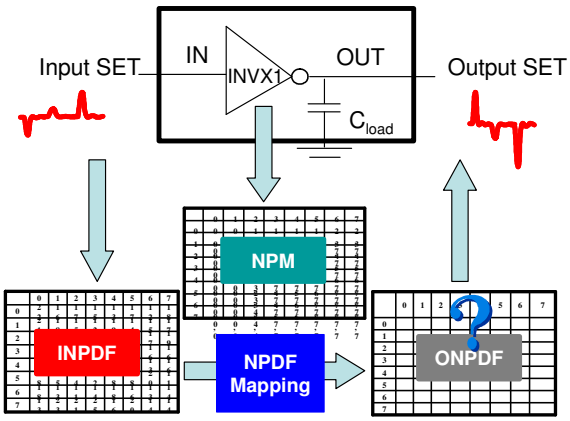
3.4 Noise Impact Analysis Using NPDF Transformation

This section presents the "noise impact analysis" in detail. The core technology is the static "NPDF transformation" process whose basic idea is to eliminate SETs that can not potentially cause observable errors as they propagate in the circuit. As both the SETs and the circuit are represented in matrices (NPDFs and NPMs), their interactions can be described by matrix operations. Specifically, when glitches propagate through a gate, the change of glitch shape distribution are described by several operations on NPDF matrices: (1) The "mapping" operation (3.4.1) maps the NPDF at the input of a gate to the output using proper NPMs; (2) The "reshaping" operation (3.4.2) realizes the probability redistribution due to timing and logic masking; (3) The "superposition" operation (3.4.3) combines NPDFs from different paths at a single node; (4) The NPDF "transformation" (3.4.3) consists of repetitive usage of these operations when propagating the NPDFs inside the circuit. As the transformation proceeds to an endpoint DFF, a "cumulative NPDF" is obtained, from which probability of captured SETs can be determined (3.4.4).

3.4.1 NPDF Mapping

The shape of a transient glitch will change as it propagates from the input to the output of a logic gate. Once the shape of the input glitch is known, the shape of the output glitch can be determined using a proper NPM of the gate. The shape distribution

described by NPDF will change accordingly. As shown in Figure 3-4(a), the NPDF at the output (*ONPDF*) can be viewed as the NPDF at the input (*INPDF*) being redistributed by a proper NPM. This redistribution procedure is defined as "NPDF mapping", described using the pseudo-code in Figure 3-4 (b): each *INPDF[i,j]* is mapped to a certain location *[k,l]* in the *ONPDF*, where the destination *[k,l]=NPM[i,j]*. Note that this is a multi-to-1 mapping so when multiple locations in *INPDF* are mapped to the same location in *ONPDF*, the values accumulate. To better understand the mapping operation, one can view the NPM as the transfer function of an optical lens that maps an object (*INPDF*) to an image (*ONPDF*) of a different shape. Figure 3-5 shows the mapping of the NPDF in Figure 3-2(a) using the NPM in Figure 3-2(b). The *INPDF* and the NPM are re-drawn in (a) and (b). For example, *INPDF[2,4]=20* and *NPM[2,4]=[3,7]*, so *ONPDF[3,7]* is set to "20" (c). After mapping every *INPDF[i,j]*, the final ONPDF is shown in (d).



(a) Model Abstraction

```
NPDF_Mapping (INPDF, NPM)
N ← The number of rows in the tables
M ← The number of columns in the tables.
/* Initialization of the ONPDF table */
for i = 0 to N-1
    for j = 0 to M-1
        ONPDF[i,j] ← 0
/* NPDF Mapping */
for i = 0 to N-1
    for j = 0 to M-1
        P ← INPDF[i,j]
        [k,l] ← NPM[i,j]
/* If multiple location in INPDF is mapped to the
   same location, the result accumulates */
        ONPDF[k,l] ← ONPDF[k,l] + P
```

(b) Pseudocode

Figure 3-4 NPDF Mapping

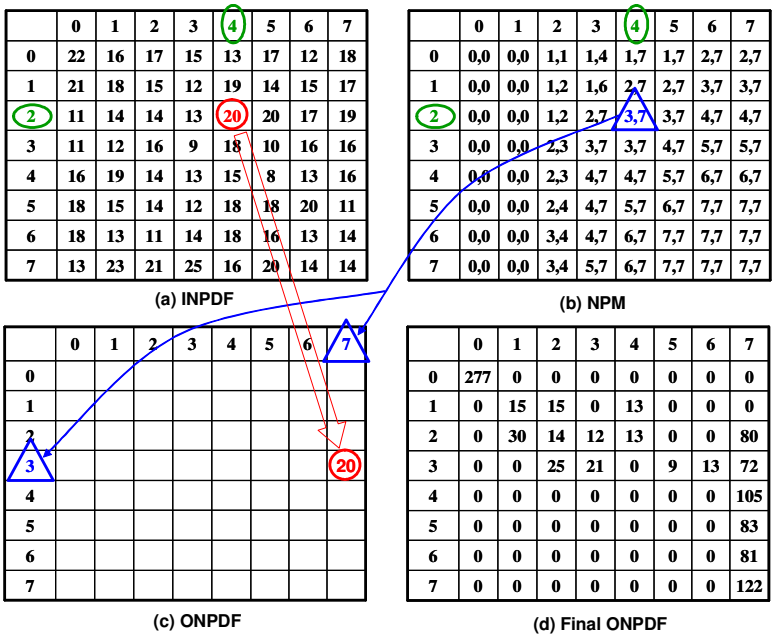


Figure 3-5 NPDF Mapping Example

3.4.2 NPDF Reshaping

Given an NPDF and a reshaping matrix R ($0 \le R[i,j] \le 1$ for all $[i,j]$) of the same size, reshaping operation $NPDF' = NPDF ** R$ is defined as shown in (Eq.4). It scales every $NPDF[i,j]$ by a fraction ($R[i,j]$) and moves the rest $(1-R[i,j])$ to $NPDF[0,0]$.

$$NPDF'[i,j] = \begin{cases} R[i,j] * NPDF[i,j], & [i,j] \neq [0,0] \\ NPDF[0,0] + \sum_{[k,l] \neq [0,0]} (1 - R[k,l]) * NPDF[k,l], & [i,j] = [0,0] \end{cases} \quad (3.4)$$

The effects of logic and timing masking can be realized by reshaping an NPDF with proper reshaping matrices. The logic reshaping matrix is defined as $R_l^N[i,j] = P_l^N$ for all i's and j's, where P_l^N is the surviving probability. It reflects the fact that SETs of all shapes at one input of a gate will cease to propagate if any of the other inputs carries

controlling value so they should be eliminated from further propagation. The timing reshaping matrix R_t^N is defined as: $R_t^N[i,j]=P_t^N(i)$ for all j's, where $P_t^N(i)$ is the overlapping probability when the quantized glitch width w equals to i. Note that unlike logic reshaping, the content of the timing reshaping matrix is different for different rows because the overlapping window is a function of the glitch width. It reflects the fact that only a portion $P_t^N(i)$ of SETs can reach a DFF within its sampling window so the rest should be eliminated from further propagation.

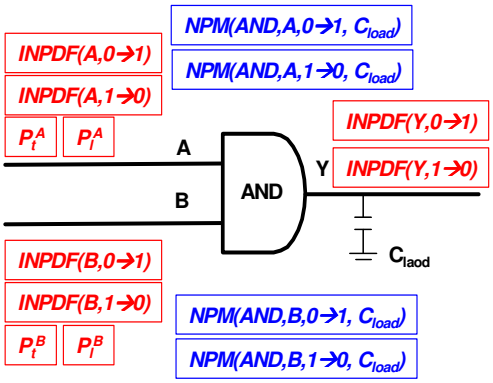
3.4.3 NPDF Transformation

The NPDF transformation consists of a series of mapping and reshaping operations at individual gates as well as superposition of NPDFs propagated from different paths. A 2-input *AND* gate (Figure 3-6 (a)) is first used as an example to describe the NPDF transformation at the single gate.

First, *INPDF*'s at all nodes *A*, *B* and *Y* are timing-reshaped to eliminate SETs originated at all nodes that will not be able to reach any DFF within the sampling window. For example, the positive-transition NPDF at input *A* is reshaped by R_t^A: $INPDF_t(A,0\rightarrow1)= INPDF(A,0\rightarrow1)**R_t^A$.

Second, the timing-reshaped $NPDF_t$'s at *A* and *B* are logic-reshaped to eliminate SETs at the inputs that will be logic-masked from propagation. For example, $INPDF_t(A,Y,0\rightarrow1)$ is reshaped by R_l^A: $NPDF_l(A,Y,0\rightarrow1)=INPDF_t(A,Y,0\rightarrow1)**R_l^A$, where R_l^A equals to the probability of *B* having non-controlling value ("1" for an AND gate).

Third, the logic-reshaped $INPDF_l$'s at *A* and *B* are mapped to Y is to determine how the SET distribution (that "survives" the timing and logic masking effects) is changed by electrical masking. For example, $INPDF_l(A,0\rightarrow1)$ is mapped to $ONPDF(A,Y,0\rightarrow1)$ by $NPM(AND,A,0\rightarrow1,C_{load})$.



(a) AND Gate Example

```
/* Transformation of the 0→1 NPDF from A to Y of a 2-input AND gate*/
NPDF_tranformation(INPDF(A,0→1), P_t^A, P_l^A,
NPM(AND,A,0→1, C_load))
begin
    N ←Number of rows; M ←Number of columns
    for i = 0 to N-1
        for j = 0 to M-1
            R_t^A[i,j] = P_t^A(i) /* Construct the timing re-shaping matrix*/
            R_l^A[i,j] = P_l^A    /* Construct the logic re-shaping matrix*/
    /* 1.Reshaping the input NPDF by the timing reshaping R_t^A */
    INPDF_t(A,0→1) = INPDF(A,0→1)**R_t
    /* 2.Reshaping the result NPDF by the logic reshaping matrix R_l^A*/
    INPDF_l(A,Y,0→1)= INPDF_t(A,0→1) **R_l
    /* 3.Mapping the reshaped INPDF to the output using the proper NPM*/
    ONPDF(A,Y,0→1)= NPDF_Mapping(INPDF_t(A,0→1),
NPM(AND,A,0→1,C_load))
end
```

(b) Pseudo-code

Figure 3-6 NPDF Transformation

Figure 3-6 (b) shows the procedure of transforming an INPDF at input *A* to output *Y* for a positive glitch. Since logic gates usually have different response to positive and negative transition noise (as characterized by different NPMs), transformations of both type transitions need to be done. As a result, four *ONPDF*'s are generated at output *Y*.

Finally, *ONPDF*'s from pin *A* and *B* of the same transition types are merged with INPDF of the same transition type at *Y* by "NPDF superposition" (realized as matrix addition) for the "*cumulative NPDF (C_NPDF)*":

$$C\_NPDF(Y,0\rightarrow 1)[i,j] = INPDF_t(Y,0\rightarrow 1)[i,j] + ONPDF(A,Y,0\rightarrow 1)[i,j] + ONPDF(B,Y,0\rightarrow 1)[i,j] \quad (3.5)$$

$$C\_NPDF(Y,1\rightarrow 0)[i,j] = INPDF_t(Y,1\rightarrow 0)[i,j] + ONPDF(A,Y,1\rightarrow 0)[i,j] + ONPDF(B,Y,1\rightarrow 0)[i,j] \quad (3.6)$$

The *C_NPDF*'s contain information about the SET distribution at node *Y*. It includes not only those that may be originated at *Y*, but also those propagated from *A* and *B*. Note that for a negating gate, the polarity of the *ONPDF*'s propagated from the inputs needs to be inverted before merging with the INPDF at the output to obtain the cumulative NPDF.

For a circuit consisting of multiple gates and many levels of logics, the logic-reshaping, mapping and superposition operations need to be performed repetitively as the NPDF travels in the circuit netlist from PIs to POs. In order to achieve optimal processing speed, an efficient netlist tracing algorithm has been developed based on breadth-first-search (BFS) [27]: The circuit is first converted to a directed graph where the logic gates are represented by vertices and wires by the edges going from the driving gate to the driven gate(s). Then starting from PIs (roots), the frontier is expanded between discovered and undiscovered vertices uniformly across the breadth of the frontier. Transformations of NPDFs from the predecessors are performed at each vertex on the frontier, and are repeated until the BFS reaches all POs (leaves).

3.4.4 Calculating DFF Noise Capture Ratio

When a cumulative NPDF encounters a DFF, it shows a shape distribution of SETs that can reach the DFF within its sampling window. In general, DFFs are only

sensitive to glitches with enough strength – only SETs in certain region of the NPDFs may be captured. This region is defined as the "dangerous zone" as show in Figure 3-7.

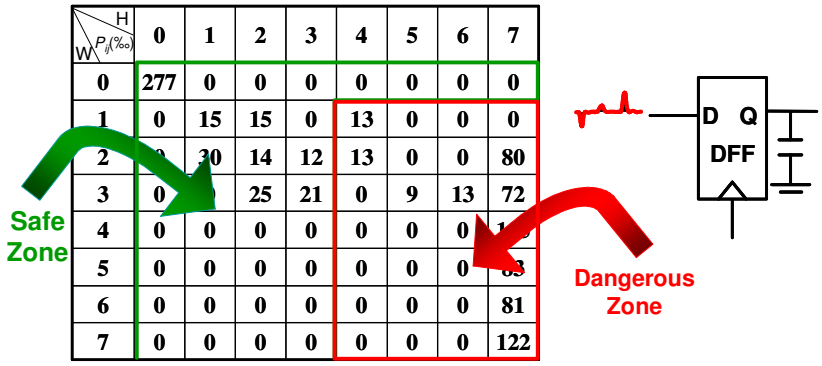


Figure 3-7 DFF Dangerous Zone in NPDF

Projecting the dangerous zone of a DFF onto the incoming cumulative NPDFs, a "noise capture ratio" R_c can be calculated as:

$$R_c = P^D(0) * \sum_{[i,j] \in DZ} C\_NPDF_{pos[i,j]} + P^D(1) * \sum_{[i,j] \in DZ} C\_NPDF_{neg}[i,j] \qquad (3.7)$$

where $P^D(0)$ and $P^D(1)$ are the logic probabilities for logic 0 and 1 at the DFF input pin D, respectively; the summations are taken over all entries in the dangerous zone ("*DZ*") in the positive- and negative-transition NPDFs. A weighted sum of the two different polarities is needed because at the DFF input, a positive glitch can appear only if the nominal value is 0 and a negative glitch can appear only if the nominal value is 1.

The noise capture ratios of all DFFs in a circuit are the key result of the noise impact analysis which accurately measures the SET effects in digital circuits. The analysis derives a transfer function of the circuit that converts the SET distribution

(represented in NPDFs) to the SET capture probability (measured by the DFF noise capture ratios). In analogy to an optical lens system, whose transfer function is the intrinsic property of the system and does not depend on the shape of the object placed in front of it, the validity of the transfer function of the circuit does not depend on the input SET distribution.

3.5 Experimental Results

In the first two experiments, results from the noise impact analysis are compared with SPICE Monte Carlo simulation. In the third experiment, the methodology is applied to a large circuit to demonstrate its efficiency and scalability.

All experiments were performed on a 2.5GHz Pentium4 processor with 512MB RAM running Linux Redhat7.0. The circuit-under-tests were synthesized to a 0.18μm cell library using Synopsys DesignCompiler; PrimeTime was used for static timing analysis; Cadence SiliconEnsemble was used for physical layout; Mentor Graphics Xcalibre was used for RC extraction. The NPDF transformation was implemented in C++ and the entire flow was linked in TCL scripting language for a smooth interface with any TCL-based CAD tools.

3.5.1 Experiment I: One Simple Circuit

The first experiment was intended to validate the NPDF transformation technique via SPICE simulation on a very simple circuit CUT0 (Figure 3-8). During the simulation, each of all eight input combinations (“000” – “111”) was applied to the circuit for 5000

consecutive clock cycles; during each cycle, a glitch with random shape and timing was injected on one of the five nodes (IN1, IN2, IN3, E and F), so each node was disturbed 1000 times for each input vector. These 1000 glitches observed a uniform temporal distribution and their shape distribution (INPDFs) followed a 2-D normal distribution. The glitches measured at node OUT were categorized according to their shapes to generate the simulated cumulative NPDFs. It was then compared with the cumulative NPDF calculated by NPDF transformation.

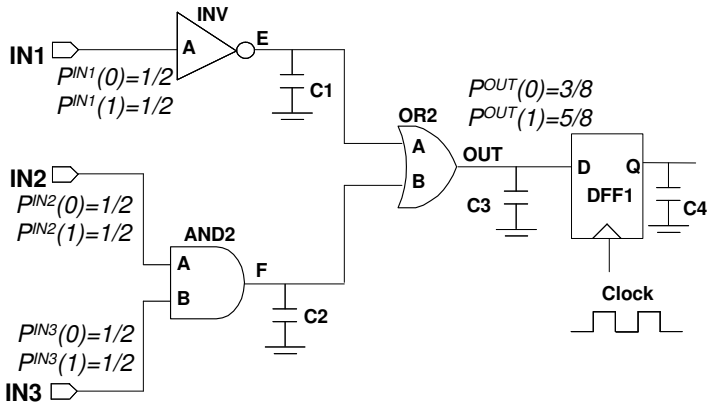


Figure 3-8 Simple Circuit used in Experiment I

Figure 3-9 shows the simulated and calculated cumulative NPDFs at node OUT. The calculated noise capture ratio R_c are listed under the tables for comparison. The DFF dangerous zones are outlined in the NPDFs as well (notice that the DFF has different dangerous zones for positive and negative glitches). NPDFs produced by NPDF transformation matched the simulation result well – the calculated and simulated R_c values are within 7% of each other. The discrepancy is partly due to accuracy loss during

the matrix quantization. Increasing the quantizer *M* will produce more accurate result, but at the cost of more memory and speed.

W \ H	0	1	2	3	4	5	6	7
0	656.95	2.3	3.75	0.9	3.05	3.45	8.7	23.3
1	0	0	0	0	0	0	0	61.85
2	0	0	0	0	0	0	0	57.2
3	0	0	0	0	0	0	0	71.5
4	0	0	0	0	0	0	0	48.6
5	0	0	0	0	0	0	0	28.95
6	0	0	0	0	0	0	0	13.9
7	0	0	0	0	0	0	0	15.6

(a) Calculated: Positive NPDF

W \ H	0	1	2	3	4	5	6	7
0	619.09	3.14	6.27	1.73	4.68	3.64	5.45	31
1	0	0	0	0	0	0	0	81.27
2	0	0	0	0	0	0	0	69.23
3	0	0	0	0	0	0	0	69.09
4	0	0	0	0	0	0	0	45.59
5	0	0	0	0	0	0	0	31.68
6	0	0	0	0	0	0	0	15.45
7	0	0	0	0	0	0	0	12.68

(b) Simulated: Positive NPDF

W \ H	0	1	2	3	4	5	6	7
0	725.05	1.6	3.9	2.7	0	0	0	33.45
1	0	0	0	0	0	0	0	67.95
2	0	0	0	0	0	0	0	68.2
3	0	0	0	0	0	0	0	36.55
4	0	0	0	0	0	0	0	41.6
5	0	0	0	0	0	0	0	10.75
6	0	0	0	0	0	0	0	7.85
7	0	0	0	0	0	0	0	0.4

(c) Calculated: Negative NPDF

W \ H	0	1	2	3	4	5	6	7
0	763.17	0.89	4.44	0	0	1.5	0	44.56
1	0	0	0	0	0	0	0	70.17
2	0	0	0	0	0	0	0	56.5
3	0	0	0	0	0	0	0	22.17
4	0	0	0	0	0	0	0	25.22
5	0	0	0	0	0	0	0	7.06
6	0	0	0	0	0	0	0	3.89
7	0	0	0	0	0	0	0	0.44

(d) Simulated: Negative NPDF

$(R_c)_{Cal}$ =0.2574

$(R_c)_{Sim}$ = 0.2378

Figure 3-9 Experiment I Results

3.5.2 Experiment II: Two Larger Circuits

The second experiment is to validate the DFF noise capture ratio using two larger circuits: CUT1 has 4 PIs, 7 internal nodes, 15 combinational gates and 8 DFFs; CUT2 has 4 PIs, 12 nodes, 21 gates and 8 DFFs. In both cases, simulations were run with all 16 input combinations (“0000” – “1111”). For each input vector, 1000 glitches with uniform temporal distribution and normal shape distribution were injected on each internal node. The number of errors observed in each DFF was counted. If the total number of injected

glitches is M_{total} and m_i errors are detected in the i^{th} DFF, then the simulated noise capture ratio $(R_c^i)_{sim} = m_i/M_{total}$. The noise impact analysis was performed on both circuits using the same glitch distribution used in the simulation. NPDFs at the input of each DFF were obtained from NPDF transformation; $(R_c^i)_{cal}$ was calculated using equation (4.10).

Table 3-1 CUT1: R_c Comparison with SPICE

DFF	DZ(pos)	DZ(neg)	Pr(1)	Cal. *Rc*	Sim. *Rc*	|Err|
1	0.0480	0.0276	0.75	0.0327	0.0317	3.0%
2	0.1242	0.0744	0.5625	0.0962	0.0940	2.3%
3	0.0983	0.0905	0.625	0.0934	0.0906	3.0%
4	0.1222	0.0938	0.5625	0.1062	0.0997	6.2%
5	0.0749	0.0817	0.5	0.0783	0.0828	5.8%
6	0.1296	0.0594	0.375	0.1033	0.0915	11%
7	0.0945	0.0510	0.25	0.0836	0.0735	12%
8	0.0587	0.0311	0.125	0.0552	0.0501	9.3%

Table 3-1 shows the numerical results for all DFFs in CUT1: "DZ(pos)" and "DZ(neg)" are the sum of entries in the dangerous zones in the positive and negative NPDF at DFF inputs, respectively; next column lists the logic "1" probabilities at the DFF inputs; the calculated and simulated R_c's are in the next two columns; and the last column lists the relative error between the calculated and simulated results: the discrepancy is within 10% for most of the DFFs. Figure 3-10 illustrates the calculated and simulated R_c in both circuits, from where a high degree of correlation can be observed.

Due to memory and capacity limitation, SPICE simulation had to be partitioned into multiple runs. It took a total of ~5480 minutes to finish the entire simulation on CUT1 and ~6824 minutes for CUT2. In contrast, given all required input, the NPDF transformation took only ~2 seconds for CUT1 and ~3 seconds for CUT2. A 10^5 speed-up

is achieved over the simulation-based approach.

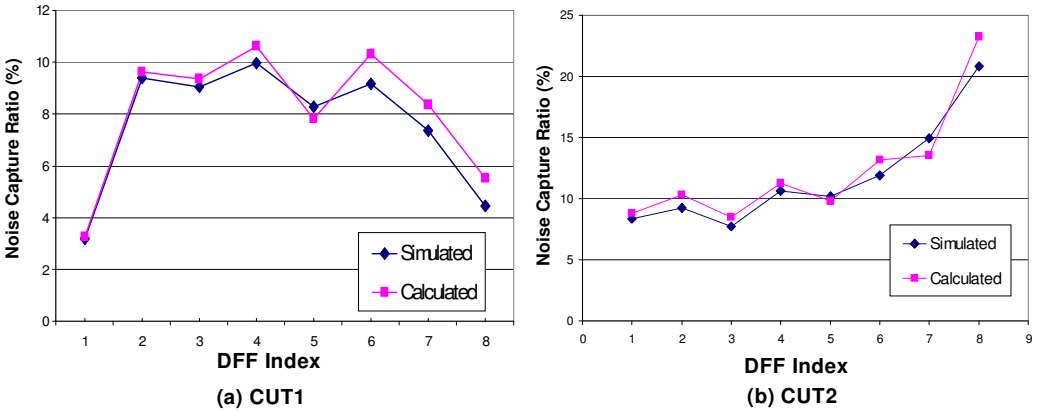


Figure 3-10 R_c Comparison with SPICE

3.5.3 Experiment III: Scalability

For circuit of larger size, SPICE simulation could not finish within reasonable time. However, the noise impact analysis is scalable to large circuits. This can be demonstrated by applying the methodology on the block EX in the Xtensa™ processor [18] used in the experiment in section 2.4.3. Given all required inputs, the R_c's for all DFFs were calculated within 16 seconds proving that the proposed technique is of high efficiency. In fact, even better performance can be expected for larger designs because the time for program setup, such as reading in the NPM database, is independent of the design size.

It is discovered that the noise capture ratios of different DFFs vary greatly. Figure 3-11(a) plotted the normalized R_c values of all DFFs: only 33 out of the 97 DFFs have very high values and about 68% have R_c values that are at least 3 orders of magnitudes

lower. This unbalanced distribution enables us to cost-effectively design highly reliable circuit – by inserting hardening cells only to the DFFs with high R_c values, the reliability of a circuit can be improved with limited design overhead.

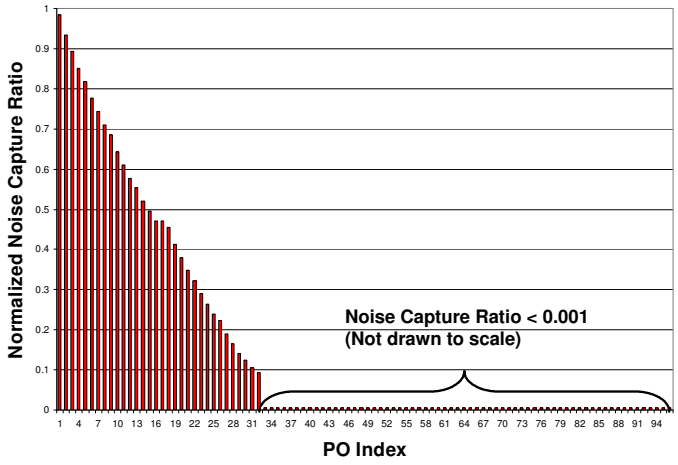


Figure 3-11 R_c Distribution in Design EX

3.6 Conclusion

In this chapter, a novel noise impact analysis methodology based on an efficient NPDF transformation technique is developed. Its accuracy, efficiency and scalability have been demonstrated through experiments. However, several issues have yet been addressed, which will direct the future research work.

First, although it is clear that the validity of the methodology does not depend on the *INPDF*'s, in order to produce realistic result, it is imperative to obtain accurate

description of SET occurrences, which is a non-trial task and requires the knowledge of the cosmic particle activities as well as the nature of the impact.

Second, the methodology only addresses one type of SEU, which is the single event transient (SET) caused by particle strikes on combinational nodes. In order to completely characterize the transient error effects in a circuit, the error effect due to direct hit on sequential elements should be incorporated in the framework.

Third, the logic tracing technique does not apply to re-converging logic paths. The logic probability at all nodes in a circuit with re-converging logic path has been proven to be NP-complete. Ignoring the re-convergence produces a pessimistic logic probability. Many works have been dedicated to finding approximation algorithms [29], which should be accommodated in the proposed methodology.

These two chapters introduce methodologies of identifying vulnerabilities in the combinational and sequential circuits. They can be used to guide the improvement of transient error tolerance of the entire circuit, which will be the topic of the next two chapters.

3.7 Acknowledgement

Chapter 3 is based on material in the published paper: Chong Zhao, Xiaoliang Bai, Sujit Dey, "A Static Noise Impact Analysis Methodology for Evaluating Transient Error Effects in Digital VLSI Circuits", in Proceedings of International Test Conference 2005 (ITC), pp. 40.2, October, 2005, Austin, Texas, USA, and material in the published paper: Chong Zhao, Xiaoliang Bai, Sujit Dey, “Evaluating transient error effects in digital

nanometer circuits", IEEE TRANSACTIONS ON RELIABILITY, VOL. 56, NO. 3, SEPTEMBER 2007, pp. 381-391. The dissertation author was the primary investigator and author of this paper.

Chapter 4. Intelligent Robustness Insertion

As nanometer circuits are becoming increasingly susceptible to radiation-induced single-event-upsets (SEUs), robustness insertion has been adopted to provide additional resiliency to the transient errors. Robustness insertion adds temporal and/or spatial redundancies into the circuit to provide capability to automatic transient error detection and correction. However, blind insertions of such redundancies may not only result in inefficient protection results, but also cause excessive design overhead. In order to provide optimal protection while keeping the associated overhead within an acceptable range, the most vulnerable circuit components need to be first identified and then protected. Hence, an accurate analysis of the overall transient error tolerance is the key. The "noise impact analysis" technique presented in the previous chapter provides such a gauging metric. In this chapter, a "constraint-aware robustness insertion" methodology is introduced to judiciously protect sequential elements in static CMOS digital circuits. It utilizes the noise impact analysis results to guide the robustness insertion. Based on a configurable hardening sequential cell design, an optimization algorithm is developed to search for the optimal protection scheme under given design constraints and budgets. An integrated framework is also constructed to automate the process.

4.1 Introduction

Concurrent error detections and corrections are becoming important techniques to improve the transient error tolerance of nanometer circuits. Circuit-level robustness insertion has been considered as one promising solution. Various hardening circuitries using spatial and/or temporal redundancies have been developed [30] [37] [36]. However, simply inserting these hardening cells into the design might not lead to the feasible or sufficient solution.

On one hand, the penalty of using redundancy to improve circuit reliability includes higher cost (in terms of larger area and higher power consumption) and lower performance (in terms of reduced clock frequency or bandwidth). The higher cost might not be acceptable for cost-sensitive mainstream applications; the reduced speed might not be allowed for high-performance products. Therefore, a valid redundancy insertion scheme has to be aware of the design constraints and specifications.

On the other hand, based on the circuit structure, noise condition and design specification, different parts of the circuit might have different levels of vulnerability and different requirements of reliability. If hardening cells are uniformly and randomly inserted without a guideline, some part of the circuit will be over-protected while some other parts will be left under-protected. Hence, intelligent use of redundancy is needed to achieve optimal protection results.

This chapter proposes an intelligent methodology – “constraint-aware robustness insertion”. By judiciously hardening the DFFs (D-type flip-flops), it provides the optimal protection scheme to improve the transient error tolerance of static CMOS digital circuits without violating design constraints and budgets. Protecting DFFs is essential in

increasing circuit reliability because a transient can become an observable error only if it is captured in a DFF. In reality, the tight design constraints may prevent sufficient hardening from being applied to all DFFs. Therefore, it is critical to know which DFFs should get high priority to receive protections and what level of protections is appropriate. A complete robustness insertion mechanism that can provide high level of protection requires three indispensable components:

1) Hardening cell: a low-cost, configurable noise-tolerant DFF design is the basic building block of the framework;

2) Robustness calibration: a method to evaluate the robustness of individual DFFs and the entire circuit is needed to provide guidelines for the insertion; and

3) Robustness optimization: an optimization algorithm is crucial in the search for the optimal protection schemes under constraints.

The constraint-aware robustness insertion methodology integrates these three factors in an automated framework. The hardening cell is developed from the low-cost "Separate-Dual-Transistor" DFF (SDT-DFF) introduced in section 1.4 by adding configurable error-resiliency to the original design. The robustness calibration is based on the result of the noise impact analysis technique introduced in the previous chapter. The robustness optimization is formulated as a multi-constrained 0-1 knapsack problem, which can be solved by a dynamic-programming-based algorithm. This is the first effort of selective hardening cell insertion under explicit guidelines of both robustness evaluation and design constraints.

The rest of the chapter is organized as follows. Section 4.2 reviews the SDT-DFF cell design with the focus set on its configurable error resiliency; section 4.3 reviews the DFF robustness calibration based on the noise impact analysis; section 4.4 describes the robustness optimization algorithm and the integrated implementation framework; section 4.5 includes experimental results and discussions; and section 4.6 concludes the chapter.

4.2 Error-Resilient Sequential Cell Design

The "Separate-Dual-Transistor (SDT)" mentioned in Chapter 1 is a low-cost error-resilient sequential cell design. By adding a reasonable amount of both spatial and temporal redundancy, it has high tolerance to transient errors caused by direct particle strike as well as single-event-transient caused by particle strike on combinational nodes. Furthermore, the level of error resiliency can be conveniently configured during the design phase to meet certain reliability requirement at different cost. In this section, the basic working mechanism of the cell design is first explained in full detail; then the error-resilience configurability is discussed and numerically measured.

4.2.1 The Robust Separate-Dual-Transistor (SDT) Latch Design

The basic structure of an SDT latch is shown again in Figure 4-1. It is similar to a traditional latch design but there are two differences. First, compared with a conventional latch, each transistor in the feedback loop is duplicated. Second, it has two input pins D1 and D2, coming from the normal input signal D but are differentiated by a preset delay value *dt* (Note that the delay element is not part of an SDT latch). An SDT latch functions as a normal latch if the input D is free of transient glitches, since D1 and D2 always carry

the same logic value (temporarily separated by the delay *dt*) and the pairing transistors are in the same states. However, an SDT latch has higher transient error tolerance than a normal latch.

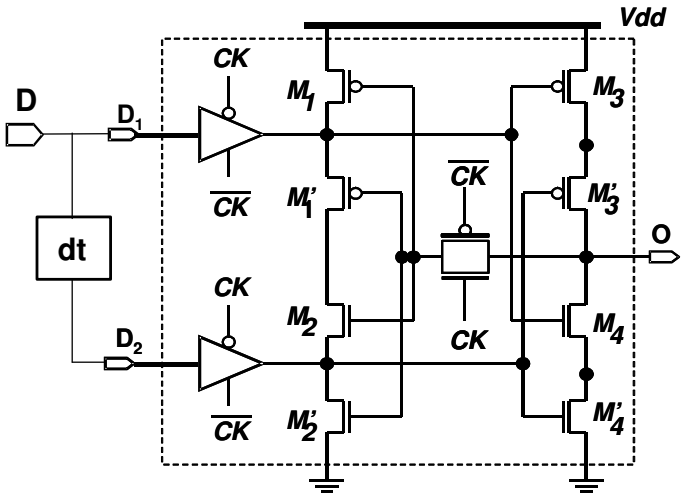


Figure 4-1 Separate-Dual-Transistor Latch Design

First, an SDT latch design is immune to a transient glitch presented at the input D whose width *w* is smaller than *dt*: during the transparent phase, if a transient glitch with duration *w* ($w \leq dt$) is present at D, at most one of D1 and D2 can carry the wrong logic value at any given time, so the pairing transistors will be in opposite states to block the charging or discharging path and the value stored in the latch is preserved until the glitch disappears when two differentiated signals become the same again. Second, an SDT latch is also immune to a sequential transient error caused by direct particle hit on any of single transistors (M1-M4, M1'-M4'). For example, if M1 endures a particle hit during its OFF state, it might be temporarily shorted and switch to its ON state; however, M1' remains in OFF state, locking the original logic value. Furthermore, in order to avoid the simultaneous upset of the paring transistors caused by a single hit ("size effect"), the two

transistors in a pair (such as M1 and M1') should be spatially separated by other transistors (such as M3 and M3') during the standard cell physical design.

4.2.2 Design and Characterization of SDT Flip-Flops (SDT-DFF)

The structure of an SDT-DFF is shown in Figure 4-2. It is composed of two identical SDT latches, and a delay element with delay *dt*. From the discuss above, it is obvious that an SDT-DFF constructed in this manner is immune to both sequential transient error and transient glitch narrower than *dt*.

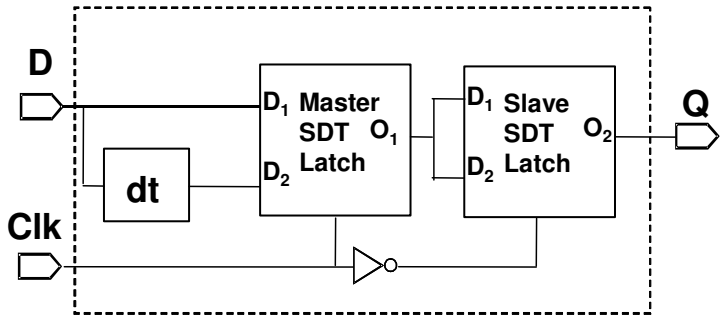


Figure 4-2 SDT Flip-Flop Design

A DFF captures data during its sampling window and is only susceptible to incoming glitches with enough strength, described by its "noise rejection curve (NRC)", as shown in Figure 4-3(a). The widths and heights of the incoming glitch are labeled on the x-axis and y-axis, the curve is obtained such that only noise with shapes in the region above it can be captured by the DFF. Te region above the curve is the "noise sensitive zone". Apparently, the smaller the sensitive zone, the wider incoming transient glitch the DFF can tolerate, and the higher transient error tolerance it has.

The higher error tolerance of an SDT-DFF as compared to a regular DFF can be reflected on the curve as its noise sensitive zone shrinks to above the horizontal line *w=dt* (Figure 4-3 (b))**.** The larger the delay *dt*, the smaller the sensitive zone and the more error-tolerant the SDT-DFF is. Hence, if a design uses SDT-DFFs instead of the conventional DFFs, glitches originated inside the combinational circuit will not become observable errors if their durations are less than a certain value. In this chapter, the operation of replacing an existing DFF in a design by a SDT-DFF will be referred to as "SDT insertion", or simply "insertion".

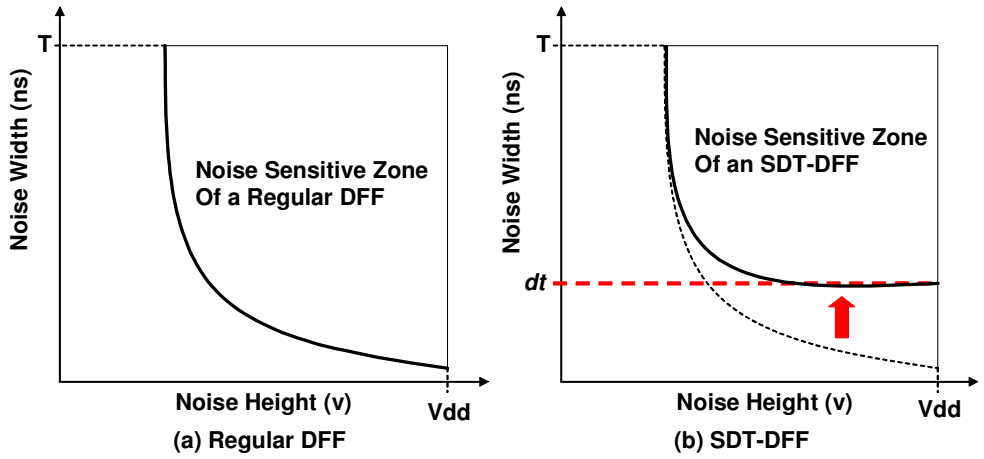


Figure 4-3 Noise Rejection Curve and Noise Sensitive Zone

In reality, an SDT-DFF can be constructed from a conventional DFF standard cell by transistor duplication and delay insertion. The delay cell can be implemented as a chain of identical buffers. The notation SDT-DFF(*n*) refers to an SDT-DFF with n buffers. Since an SDT-DFF only captures the correct data after the paring transistors stabilize to the same state, the clock-to-Q delay of an SDT-DFF is longer, which lengthens the timing path. In addition, an SDT-DFF has a larger area. The area and timing

cost as compared with the original DFF can be viewed as having two parts: the transistor duplication causes fixed area overhead δA_{dup} and timing overhead δT_{dup}, whereas the overheads due to delay insertion is proportional to the number of buffers. If each buffer has an area of δA_{buf} and a delay of δT_{buf}, the timing and area increase of an SDT-DFF(n) ($n>0$) as compared with the original DFF can be expressed as functions of the number of buffers n:

Timing: $$\Delta T(n) = \delta T_{dup} + n * \delta T_{buf} \quad (4.1)$$

Area: $$\Delta A(n) = \delta A_{dup} + n * \delta A_{buf} \quad (4.2)$$

Since designs usually have certain timing and area budget, SDT insertion is limited to a certain extend due to the associated area and timing overhead. In order to decide where the SDT insertion should be done and how many buffers should be used in the delay cell, two factors has to be considered. The first is how much a DFF can tolerate the area and timing overhead, which is determined by the design constraints and budget. The second is which DFF(s) might be more possibly disturbed by transient errors and thus need protection the most, which is the topic of the next section.

4.3 Robustness Calibration

The robustness of a DFF can be measured by the likelihood that the stored value is unintentionally altered by transient glitches originated in the combinational circuits. Different DFFs in a design might have very different robustness. The role of robustness calibration is to provide gauging metrics of the circuit robustness, based on which the

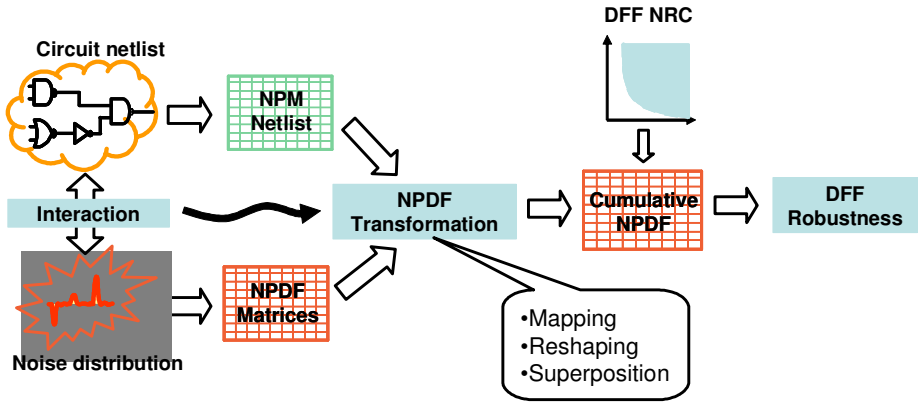
optimization algorithm can provide an optimal insertion scheme when only part of the desired SDT insertions can be made due to the design constraints.

The robustness calibration is based on the "noise impact analysis" introduced in the previous chapter. To summarize its basic operation, as shown in Figure 4-4(a), each combinational gate in a given circuit is modeled by a "Noise Propagation Matrix (NPM)". The distributions of random glitches at all internal nodes are abstracted in a matrix format called "Noise Probability Density Function (NPDF)". As the glitches propagate in the circuit, the three masking effects of the combinational gates will change the distribution, modeled by the "NPDF Transformation", which consists of a series of operations on the NPDFs, such as mapping, reshaping and superposition. When the transformation completes, a "Cumulative NPDF" is obtained at the input of each DFF, which is then post-processed by the noise rejection curves to determine the DFF robustness.

A cumulative NPDF example is shown in Figure 4-4(b). It contains the statistical information of the transient glitches that survives the three masking effects to reach a DFF. Each table entry P_{ij} represents the probability for the noise propagating to the receiving DFF during its sampling window with certain shapes. As an example, P32=0.025 means that for every 1000 glitches reaching the DFF within the sampling window, 25 will have the width of 0.3ns and height of 0.2v.

Projecting the noise rejection curve of the DFF onto the cumulative *NPDF*, the region covered by the "noise sensitive zone" represents the distribution of the glitches that can be captured by the destination DFF to cause stable errors, so the DFF "*Robustness*" is defined as:

$$R(DFF) = \frac{1}{\sum_{[i,j] \in NSZ} P_{ij}} \tag{4.3}$$



(a) NPDF Transformation

H(v) / W(ns)	0	0.1	0.2	0.3	0.4	0.5	0.6	>0.7
>0.7	0	0	0	0	0	0	0	.122
0.6	0	0	0	0	0	0	0	.081
0.5	0	0	0	0	0	0	0	.083
0.4	0	0	0	0	0	0	0	.105
0.3	0	0	.025	.021	0	.009	.013	.072
0.2	0	.030	.014	.012	.013	0	0	.080
0.1	0	.015	.015	0	.013	0	0	0
0	.277	0	0	0	0	0	0	0

dt = 300ps (n=3)
R(SDT-DFF(3)) = 1/.371 =2.70

dt = 200ps (n=2)
R(SDT-DFF(2)) = 1/.485 =2.06

dt = 0 (Original DFF)
R(DFF) = 1/.578 =1.73

(b) Cumulative NPDF Example

Figure 4-4 Robustness Calibration

Note that the *Robustness* is actually the reverse of the noise capture ratio derived in the previous chapter. A higher *Robustness* indicates the DFF is less likely to latch incoming glitches and therefore is more robust. Under the same condition, an SDT-DFF has higher *Robustness* because of its smaller *NCZ* and its *Robustness* increases with *dt*.

For example, in Figure 4-4(b), if each buffer has a delay of 100ps, the *Robustness* of an SDT-DFF(2) (*dt*=200ps) increases to 2.06 from 1.73 of a regular DFF and to 2.70 for an SDT-DFF(3) (*dt*=300ps).

A "*Robustness Function*" *RF* of the entire circuit with *M* DFFs can also be defined as a weighted sum over the *Robustness* of all individual DFFs, which measures the reliability level of a circuit.

$$RF = \sum_{j=1}^{M} w_j \cdot R(DFF_j) \tag{4.4}$$

where w_j is an optional weighting factor that can be heuristically introduced to represent its functional significance.

4.4 Constraint-Aware Robustness Insertion

The reliability of a circuit can be improved through SDT insertion but it is not always acceptable to introduce excessive area overhead or to insert a large delay on critical timing paths. Therefore, SDT insertion should be selectively used. In this section, an algorithm that searches for the insertion scheme that maximizes the improvement under given design constraints will be presented.

4.4.1 Problem Formulation

The first step in formulating the problem is to derive expressions of timing and area constraints based on the design specification and budget. A specification refers to a set of requirements a design has to meet to fulfill proper functionality; whereas a budget

refers to the extra price the designer is willing to pay for the purpose of reliability improvement.

If the design needs to run at the same clock period *T*, the timing constraint requires that the insertion-related extra delay should not exceed the available timing slack $t_{avail}(j)$ at the j^{th} DFF; if the designer is willing to slow down the clock by ΔT_{OH} in exchange for higher reliability, all DFFs acquire the same extra time for more insertion. Using equation (5.1), the timing constraint for the j^{th} DFF in a design with *M* DFFs can be written as:

$$\delta T_{dup} + n_j * \delta T_{buf} \le t_{aail}(j) + \Delta T_{OH}, \qquad 1 \le j \le M \tag{4.5}$$

Due to the timing constraint, SDT insertion at the endpoint DFFs of critical paths might be limited or prohibited, whereas for timing paths with large positive slacks, it might be possible to insert SDT DFFs with several delay buffers without causing timing violation.

The area constraint requires that the overall insertion-related area increase should not exceed the designer budget ΔA_{OH}:

$$\sum_{j=1}^{M} (\delta A_{dup} + n_j * \delta A_{buf}) \le \Delta A_{OH} \tag{4.6}$$

The timing constraint determines the allowed insertion (n_j) at a specific DFF. For designs with tight speed requirement, ΔT_{OH} can be set to zero so $\Delta T_{OH} \ge 0$. The area

constraint determines the allowed SDT insertion at all DFFs. Area overhead can not be totally avoided so $\Delta A_{OH} > 0$.

Note that the constraints are obtained based on the assumption that the design change due to SDT insertion is localized and unlikely to change the global placement and routing, so the change in wire delay and routing area are negligible.

Using the robustness calibration introduced in the previous section together with the timing and area constraints, the problem of constrained robustness insertion can be viewed as to maximize a circuit's *Robustness Function* through proper SDT insertion under the timing and area constraints. Mathematically, the constraint-aware robustness insertion can be formulated as:

"Given a circuit with M DFFs, find an assignment {n_j} ($1 \le j \le M$) such that if the j^{th} DFF is replaced by an SDT-DFF(n_j), the Robustness Function in (4.4) is maximized, subject to timing constraint defined in (4.5) and area constraint defined in (4.6)."

This problem is similar to the 0-1 knapsack problem, or the multi-constrained money allocation problem [40], and it can be solved by dynamic programming [17]. Obviously, the optimality of the final solution requires the optimal sub-structure:

$$RF_j(n) = \max_{\substack{\Delta T(n_j) \le t_{avail}(j) + \Delta T_{OH} \\ \sum_{l=1}^{j} (\delta A_{dup} + n_l * \delta A_{buf}) \le \Delta A_{OH}}} \left\{ RF_{j-1}(n), RF_{j-1}(n - n_j) + w_j \cdot R(n_j) \right\} \qquad (4.7)$$

where $RF_j(n)$ is the *Robustness Function* when a total of n buffers are inserted into the first j DFFs, n_j is the number of buffers inserted at the j^{th} DFF, and $R(n_j)$ is the *Robustness* of the j^{th} DFF with n_j buffers.

4.4.2 A Dynamic Programming Solution

The solution to this optimization problem consists of two steps. The first step is to iteratively compute the maximum *RF* using the recursive relation (4.7). The second step is to reconstruct the corresponding assignment $\{n_j\}$.

The pseudo-code *Calculate_Maximum_Robustness* in Figure 4-(a) describes the first step. First, the available area (a_{avail}) is initialized to the area budget ΔA_{OH} (line 1) and the total number of inserted buffers n is set to zero (line 2). Then one DFF is being considered in each iteration of the main loop (line 3-18): Line 4-5 calculates the number of buffers (n_{limit}) allowed for the current DFF based on timing ($(n_j)_{MAX}$) and area constraint (a_{avail}). If no insertion is allowed, the *Robustness Function* and the buffer assignment remain unchanged (line 6-8). Otherwise, as the inner loop (10-18) iterates through each allowed buffer number, the new *Robustness Function* is calculated. If it produces a better result, the *Robustness Function*, the total number of buffers and available area are updated accordingly (line 15-18). A two-dimensional array *TB_DIR[n,j]* is maintained to remember the decision. If no insertion at the current DFF, the value is set to be "*Left*", otherwise it is set to be "*Up*".

The pseudo-code *Find_Buffer_Assignment* in Figure 4-(b) describes the second step by back-tracing the *TB_DIR* array. A one-dimensional array *DU[j]* is used to store

the number of buffers inserted at the j^{th} DFF. Starting from *Find_Buffer_Assignment(K,M)*, with *K* being the total number of inserted buffers and *M* being the total number of DFFs, whenever an "*Up*" (line 3) is encountered in *TB_DIR[n, j]*, *DU[j]* is incremented by 1 (line 5), and the trace-back continues for the same DFF (line 4). Otherwise, the trace-back proceeds to the $(j-1)^{th}$ DFF (line 7). The recursion ends when either all DFFs are considered or the allowed number of buffer is reached (line 1-2), when *DU[j]* gives the number of buffers to be inserted to the j^{th} DFF.

```
Calculate_Maximum_Robustness(M, ΔA_OH, {(n_j)_MAX})
1.  a_avail ← ΔA_OH
2.  n←0
3.  for j←1 to M
        /* Calculate the number of delay buffers allowed */
4.      n_avail = int((a_avail-δa_dup)/δa_buf)
5.      n_limit = min {n_avail, (n_j)_MAX}
6.      if (n_limit == 0) then
            /* If no more buffer is allowed, the total robustness function remains unchanged */
7.          RF_j(n) ← RF_j-1(n)
8.          TB_DIR[n,j] ← "left"
9.      else
10.         for n_j ← 0 to n_limit
11.             if (RF_j-1(n-n_j)+w_jR(n_j)) < RF_k-1(n)) then
                    /* If the insertion of n_j delay buffers does not increase the robustness function,
                       nothing is changed */
12.                 RF_j(n) ← RF_j-1(n)
13.                 TB_DIR[n,j] ← "left"
14.             else
                    /* Otherwise, adopt this insertion and update the robustness function and remaining
                       area constraint. The traceback path is updated as well */
15.                 RF_j(n) ← RF_j-1(n-n_j) + w_jR(n_j)
16.                 TB_DIR[n,j] ← "up"
17.                 a_avail ← a_avail – Δa(n_j)
18.                 n ← n + n_j
```

(a) Pseudo-code: *Calculate_Maximum_Robustness*

Figure 4-5 Pseudo-Code: Robustness Optimization

```
Find_Buffer_Assignment (n, j)
1.      if n=0 or j = 0
            /* End of back tracing */
2.          return
3.      if TB_DIR[n,j] == "Up"
4.          Find_Delay_Assignment(n-1,j)
        /* Insert one more buffer unit to the jth DFF */
        /* Note: DU[j] should be initialized to zero before back-tracing. */
5.          DU[j] ← DU[j]+1
6.      else
        /* Do not insert more delay buffer to the jth DFF */
7.              Find_Delay_Assignment(n,j-1)
```

(b) Pseudo-code: *Find_Delay_Assignment*

Figure 4-5 Pseudo-Code: Robustness Optimization (continued)

4.4.3 Implementation Framework

An integrated framework has been developed to automate the entire SDT insertion process, as shown in Figure 4-6. A collection of "SDT-DFF Cells" is first constructed and their noise rejection curves were pre-characterized and saved in the "NRC Database". Then given the gate-level netlist of a "Circuit to Be Hardened", the "Robustness Calibration Engine" runs the NPDF transformation to obtain the cumulative NPDFs at the input of all DFFs, and calculates the *Robustness* of individual DFF under different configurations. The "Constraint Generation Engine" derives the area and timing constraints based on static timing analysis ("STA") result and the "Design Budgets". The "Robustness Optimization Engine" uses all information and executes the algorithm described in Section 4.4.2 to find the optimal buffer assignment for all DFFs. Finally the selected DFFs are replaced by SDT-DFFs with proper delay values to produce the "Hardened Circuit". The entire framework was linked together using TCL scripts for a

smooth interface with TCL-based STA tools such as PrimeTime. The two key engines, the Robustness Calibration Engine and the Robustness Optimization Engine, are implemented in C.

Since the majority of the required information is pre-characterized (such as the noise rejection curves) or available through other regular design analysis (such as STA), the execution of this framework is highly efficient and can be conveniently integrated to the existing design flow. It is also worth emphasizing that the validity of the optimization algorithm does not depend on a particular choice of the hardening cell design or the robustness calibration technique. Any hardening cell that possesses configurability in its error-resiliency and any robustness calibration method that measures the relative DFF robustness can be used in the framework.

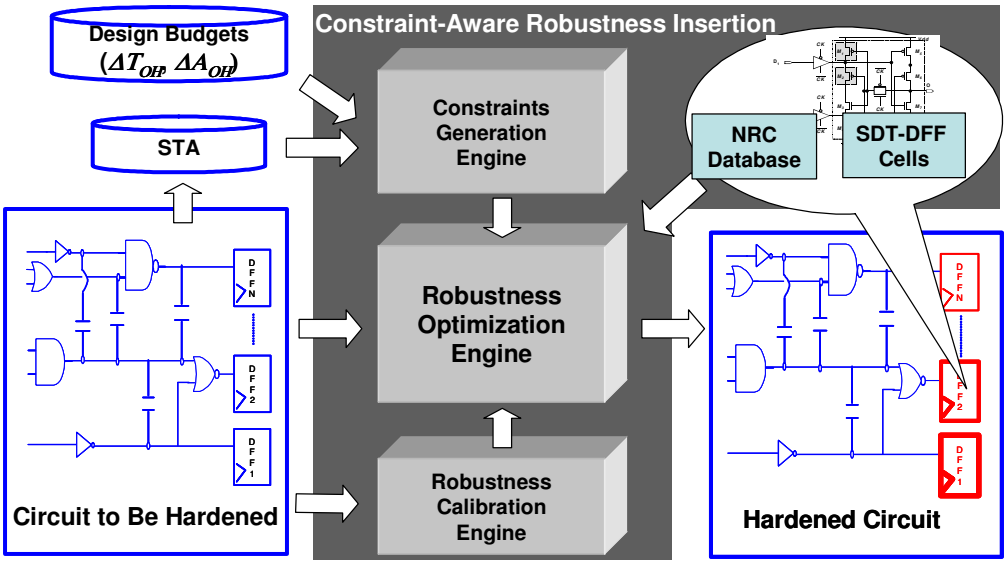


Figure 4-6 Constraint-Aware Robustness Insertion Framework

4.5 Experimental Results

In this section, some preliminary experimental results will be presented. The first experiment involves the construction and characterization of the SDT-DFFs. Then a detailed case analysis is performed on a small circuit. Next, the methodology is applied to several circuits with various sizes. Finally, a large commercial circuit block is used to demonstrate the efficiency of the methodology, where an interesting observation is discovered and discussed.

4.5.1 STD-DFF Construction and Characterization

Three SDT-DFFs with 1, 2 and 3 delay buffers were constructed from one regular DFF in a 0.18μm cell library and their timing and area were then characterized. Table 4-1 lists the results: The column labeled "Original DFF" shows the total area and timing overhead (which is zero) of the original DFF. The next three columns show the cell area and timing overhead of the three SDT-DFFs. The values of δA_{dup}, δA_{buf}, δT_{dup} and δT_{buf} are all listed above the table.

Table 4-1 Area and Timing Overhead of SDT-DFFs

$\delta A_{dup} = 29$, $\delta A_{buf} = 13$, $\delta T_{dup} = 100ps$, $\delta T_{buf} = 100ps$

	Original DFF	SDT-DFF(1)	SDT-DFF(2)	SDT-DFF(3)
Area (μm^2)	70	112	125	138
ΔT (ps)	0	200	300	400

4.5.2 Full Case Analysis on CUT0

In order to check the entire flow and to prove the validity of the methodology, a full case analysis was performed on CUT0: a small circuit implemented in the same 0.18μm cell library. CUT0 has 4 primary inputs, 12 internal nodes and 21 gates (including 8 DFFs). The entire case analysis consists of 4 steps.

Step 1: Calibrating DFF *Robustness*

The Robustness Calibration Engine calculated the *Robustness* values of all the DFFs with different configurations, as shown in Table 4-2. The column labeled $R_j(0)$ lists the values of the j^{th} original DFFs. The columns labeled $R_j(n)$ ($n=1,2,3$) list the values if the j^{th} DFF were replaced by an SDT-DFF(n). Also two facts about the DFF *Robustness* have been proved: first, different DFFs have very different values; second, the value monotonously increases with the number of buffers for each individual DFF.

Table 4-2 Robustness Calibration and Constraint Setting of CUT0

j	$R_j(0)$	$R_j(1)$	$R_j(2)$	$R_j(3)$	$t_{avail}(j)$	$n_{max}(j)$
1	12.0	15.7	36.9	57.2	227ps	1
2	10.8	13.5	32.2	59.4	379ps	2
3	12.9	15.2	27.4	43.1	318ps	2
4	9.4	14.2	32.9	50.0	304ps	2
5	9.8	11.9	28.4	49.2	83ps	0
6	8.4	11.0	27.1	52.3	48ps	0
7	6.7	8.6	19.7	31.1	254ps	1
8	5.0	7.7	18.1	29.9	122ps	0

Step 2: Setting Design Constraints and Budgets

The Constraint Generation Engine derived the timing and area constraints, also shown in Table 4-2. For the timing constraints, the available timing slack of all DFFs are

obtained from STA, listed in the column "$t_{avail}(j)$". Then the additional timing budget ΔT_{OH} is set to zero and the corresponding allowed number of buffers at all DFFs are listed in column "$n_{max}(j)$". The area constraint is arbitrarily set to be 10% of the total area, equivalent to 118μm^2.

Step 3: Running Optimization

The Robustness Optimization Engine then searched for the optimal insertion scheme. A 2-D table was first developed to iteratively compute the maximum *RF* (Table 4-3). The column "$RF_j(n)$" ($1 \le j \le 8$) shows the *Robustness Function* after the first *j* DFFs have been considered and *n* delay units have been inserted during the execution of *Calculate_Maximum_Robustness*. As the process was completed, the maximum *Robustness Function* can be found at $RF_8(4)$, which is 119.9. Then the table was traced back to reconstruct the corresponding insertion scheme. The "↑"s and "←"s indicate the traceback path when executing the *Find_Delay_Assignment*. The final insertion is listed in the last row: two instances of SDT-DFF(2) were used to replace the 2nd and 4th DFF.

Table 4-3 Table Growth of the *Robustness Function*: CUT0

n	$RF_1(n)$	$RF_2(n)$	$RF_3(n)$	$RF_4(n)$	$RF_5(n)$	$RF_6(n)$	$RF_7(n)$	$RF_8(n)$
0	12.0	22.8(←)	35.7	45.1	54.9	63.3	70.0	75.0
1		25.5(↑)	38.4	47.8	57.6	66.0	72.7	77.7
2		44.2(↑)	57.1(←)	66.5(←)	76.3	84.7	91.4	96.4
3				71.3(↑)	81.1	89.5	96.2	101.2
4				90.0(↑)	99.8(←)	108.2(←)	114.9(←)	119.9(←)
n	**0**	**2**	**0**	**2**	**0**	**0**	**0**	**0**

Step 4: Checking Results

Static timing analysis performed on the new circuits confirmed that no timing violation occurred as a result of the insertion and the total area increase of 110μm^2 was within the allocated budget. In order to prove the SDT insertion indeed improved the transient error tolerance, SPICE simulation used in section 3.5.2 was performed on both the original and modified circuits. It was discovered that the error captured in the 2nd and 4th DFF were reduced by 83% and 85%, respectively. The total error count in all DFFs was reduced by 46%. Finally, to prove the optimality, all 10 possible allowed insertion schemes were enumerated and it is found that this one gave the largest *RF*.

This case study proved that the proposed methodology is indeed able to efficiently improve the transient error tolerance, under given design constraints and budgets.

4.5.3 Experiments on More Circuits

The methodology was also applied to 5 more circuits of different size: CUT1-CUT4 are common logic units in digital circuits; CUT5, is the block EX in the Xtensa™ processor [18] used in the experiment in section 2.4.3. All circuits were implemented in the same 0.18um library and clocked at 1.6ns. In all experiments, ΔT_{OH} is set to zero, allowing no speed degradation. The experiments were run under two conditions: (a) without area constraint (the area overhead was evaluated after the insertion as the protection cost) and (b) with an area budgets (~80% of the resulting area overhead in evaluated in (a)).

The results are summarized in Table 4-4: Columns 2-4 show the number of DFFs (N_{DFF}), circuit area (A_{total}), and the original *Robustness Function* (RF_{orig}), respectively.

Columns 5-7 under "*Without Area Constraints*" show the associated area overhead (ΔA), the optimized *RF* (RF_{opt}), and the *RF* improvement (ΔRF), respectively for the experiment without area constraints. Columns 8-10 under "*With Area Constraints*" show the allocated area budget (ΔA_{OH}), the optimized *RF* (RF_{opt}), and the *RF* improvement (ΔRF) for the experiment with area constraints.

From the experiment data without area constraint, it can be clearly seen that the *RF*s of all circuits were improved significantly (from 25% for CUT2 to 96% for CUT4). The difference in the improvement is due to different timing conditions because if a design is very timing critical, not many insertions are allowed, the improvement will be limited. It also showed that the area overhead decreased as the circuit size increased because sequential elements usually occupy smaller percentage of the total area in larger circuits. In the experiments with area constraint, although the improvement in *RF* decreased in all cases except for CUT3, it remained at very high level (from 20% for CUT2 to 84% for CUT4). The reason for the exception of CUT4 is that the area is not a limiting factor, i.e. the total number of insertions at all DFFs did not add up to 4% of the total circuit area)

Table 4-4 Robustness Optimization Results

CUT	N_{DFF}	A_{total} (μm^2)	RF_{orig}	*Without Area Constraint*			*With Area Constraint*		
				ΔA	RF_{opt}	ΔRF	ΔA_{OH}	RF_{opt}	ΔRF
CUT1	8	1396(40.1%)	62	19%	103	**66%**	15%	95	**54%**
CUT2	8	2490(22.5%)	103	10%	129	**25%**	8%	123	**20%**
CUT3	5	2971(11.8%)	136	5.6%	203	**49%**	4.5%	203	**49%**
CUT4	11	7479(10.3%)	280	3.8%	549	**96%**	3%	515	**84%**
CUT5	97	39894(17.0%)	2561	2.7%	3559	**39%**	2%	3388	**32%**

4.5.4 Robustness-Cost Trade-off

The last experiment was to further study the reliability-cost tradeoff and was done on the largest CUT5. By gradually increasing the timing budget (ΔT_{OH}), the achievable improvement in robustness function RF was obtained, as plotted in Figure 4-7 (the left y-axis). The experiment was also done with and without area constraints. The actual area overheads were also plotted (the right y-axis).

In the experiment without area constraints, s expected, higher reliability can be achieved with more timing penalty because more insertions are allowed. However, the improvement is not linearly increasing with ΔT_{OH}. As an example, with a 100ps timing budget (ΔT_{OH}=100ps), the achievable improvement can be drastically increased from 39% to 72%. However, an additional 100ps (ΔT_{OH}=200ps) merely results in 9% increase. This means when allocating the budget, it is desirable to target at the "Good budget allocation" regions instead of the "Bad budget allocation" region for a more economical protection results. It is also noticed that the area penalty remained at a low level (2.7%-4.5%). In reality, this small area increase might be tolerable for higher reliability unless the design is critically core limited.

When an additional area constraint of 2% was set (in case when the design is indeed core limited), the tradeoff situation became completely different and the zones of "good budget" and "bad budget" changed position with each other. This indicates that the tradeoff among reliability, timing and area needs to be quantitatively investigated when allocating the limited budget and resource.

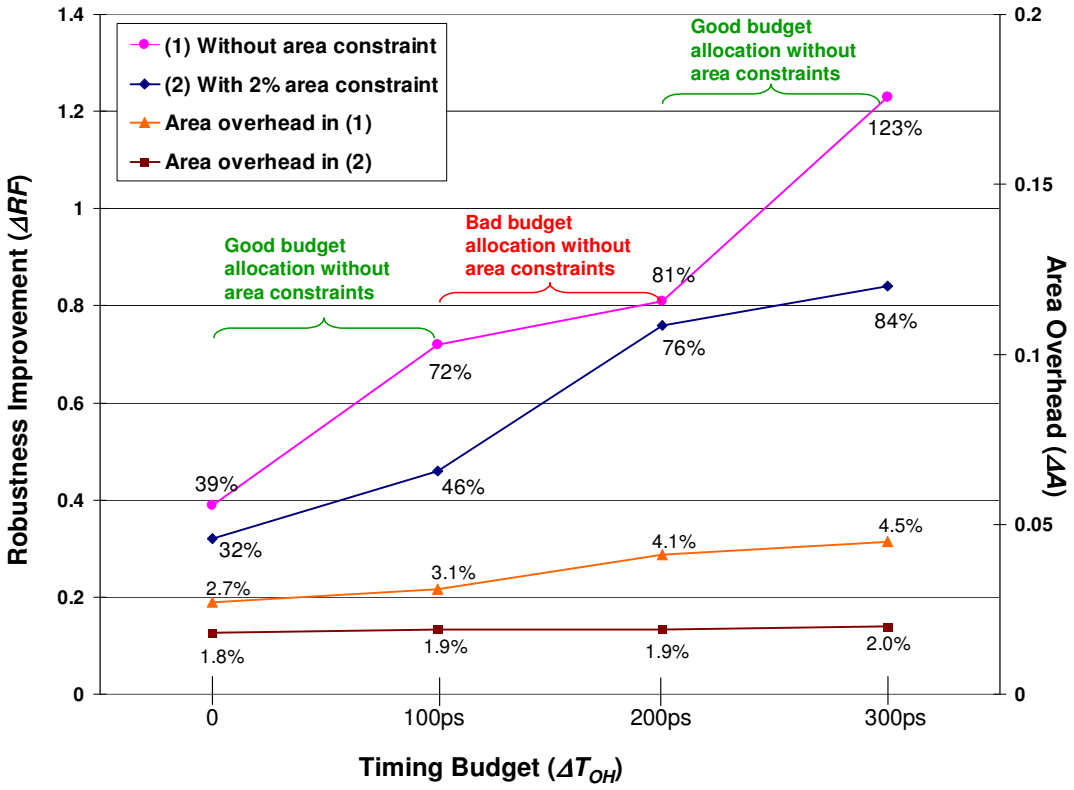


Figure 4-7 Robustness-Cost Trade-off in CUT5

4.6 Conclusion

This chapter presents a "constraint-aware robustness insertion" methodology for optimal transient error tolerance enhancement in static CMOS digital circuits. Using cost-effective hardening sequential cell design and efficient circuit robustness calibration, the robustness optimization algorithm is able to find the optimal robustness insertion scheme to increase the circuit reliability to transient errors while keeping related cost within area

and timing constraints. However, several issues need to be addressed for further improvement.

First, an SDT-DFF has relatively higher power consumption. As power is becoming a major concern in VLSI circuit design, it should be considered as a design constraint. How to achieve optimality considering timing, area and power constraints will be addressed in future works.

Second, although the SDT-DFF has high immunity to the transient glitch propagated from inside the combinational logic as well as inside sequential transient error, the optimization algorithm only considers the transient glitch in the combinational logic. The proposed framework can be further improved by taking into consideration the sequential transient error caused by direct particle strike inside the DFF.

Third, SDT insertion in the critical paths might be limited or prohibited. As a result, the endpoint DFFs of the critical paths might not be improved significantly. In order to address this challenge, further optimizations are to be done inside the combinational logic.

4.7 Acknowledgement

Chapter 4 is based on material in the published paper: Chong Zhao, Yi Zhao, Sujit Dey, "Constraint-Aware Robustness Insertion for Optimal Noise-Tolerance Enhancement in VLSI Circuits," in Proceedings of 42nd Design Automation Conference (DAC), pp. 190-195, June 2005, Anaheim, California, USA, and material in the paper accepted by IEEE Transactions on Very Large Scale Integration Systems (TVLSI): Chong Zhao, Yi

Zhao, Sujit Dey, “An Intelligent Robustness Insertion Methodology for Optimal Transient Error Tolerance”. The dissertation author was the primary investigator and author of these papers.

Chapter 5. Robustness Enhancement in Combinatorial Logics

In the previous chapter, a "constraint-aware robustness insertion" technique has been presented. By selectively hardening the most vulnerable FFs in the design while keeping track of the associated penalty, it can achieve significant improvement of transient error tolerance without causing excessive design overhead. However, as will be illustrated in this chapter, robustness insertion may not be always acceptable due to its inevitable timing penalty, especially on the most timing critical path. Under such circumstances, combinational optimization can serve as a complementary solution.

In this chapter, two circuit-level techniques targeting the combinational circuits will be presented to reduce the propagation of a single-event-transient originated inside the combinational circuit. By applying these techniques to the most vulnerable circuit element identified by the "soft spot analysis" (Chapter 2), significant improvement of transient error tolerance can be achieved. These two techniques, together with the soft spot analysis, formed the basis of the "RObustness COmpiler (ROCO)", a robustness closure framework developed to facilitate the integration of the robustness enhancement effort into the existing design flow. Experiment results show that the proposed methodology is able to greatly improve the circuit reliability with zero timing overhead and very limited area penalty.

5.1 Introduction

Sequential hardening has been considered an effective approach to protect nanometer circuits against radiation-induced soft errors. Specifically, the "constraint-aware robustness insertion" technique presented in Chapter 4 can achieve a high level of protection at limited cost. However, the additional delay incurred by the insertion might not be always acceptable. An FF is usually the endpoint of multiple timing paths so using hardened FFs lengthens all these paths, many of which are likely to be critical paths with limited or no timing slack available to consume.

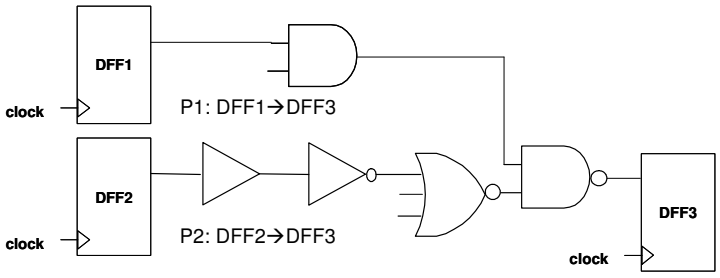


Figure 5-1 The Limitation of FF Hardening

Consider the circuit in Figure 5-1, if P2 is a critical path with little timing slack, hardening is not allowed at DFF3 because the extra delay will cause timing violation on P2. If solely relying on FF hardening, one has no choice but leaving all gates unprotected. However, since there is large timing slack in path P1, it is possible to improve the error tolerance of P1 by inserting redundancies along the path as long as the extra delay does not exceed the available positive slack; furthermore, P2 might also get certain protection by proper choices of the combinational gates on the path without increasing the delay. This means that hardening FFs is not always feasible or optimized solution. Instead,

combinational optimizations need to be utilized as complementary approach when the FF hardening can not be realized.

In this chapter, two circuit-level techniques, “gate cloning” and “cell resizing”, will be introduced to enhance the SET tolerance of combinational logics in static CMOS digital circuits. Both techniques have been adopted in the timing closure process [34][35]. With different optimization criterion, they can be used to improve SET tolerance as well. However, due to design constraints, they can only be applied to limited circuit locations because excessive protection will cause high timing and area overhead. Therefore, it is essential to first evaluate the vulnerability of all circuit nodes and select the most vulnerable nodes as the targets. The soft spot analysis can properly serve the purpose of vulnerability identification. It confirmed that transient glitches occurring at different nodes have different chances to become observable errors and only a small number of nodes are highly vulnerable. Hence, both techniques can be applied to these identified soft spots to achieve maximally improvement while limiting the design cost.

To reduce the engineering cost and avoid lengthened design cycle associated with the robustness enhancement effort, it is desirable to integrate these techniques with the existing design flow. In order to achieve this, the “RObustness COmpiler (ROCO)”, an integrated optimization engine that seamlessly interfaces with the current design flow has also been developed. As will be discussed, ROCO can serve as an implementation of a “robustness closure” step in the new integrated design follow.

The rest of the chapter is organized as follows: section 5.2 reviews the soft spot analysis; section 5.3 and 5.4 elaborate on gate cloning and cell resizing in great detail;

section 5.5 describes the ROCO framework and introduces the concept of robustness closure; section 5.6 presents the experimental results; section 5.7 concludes the chapter.

5.2 Review of Soft Spot Analysis

The soft spot analysis is a static technique that evaluates the circuit vulnerability based on the circuits' structure. A "softness" (S_N) is calculated for each node N to measure its vulnerability by considering three masking effects that tend to prevent transient at node N from causing observable errors. For the convenience of explanation, This chapter will use the notations T_N, L_N and E_N for the measurement of the timing, logic and electrical masking effects at node N, respectively. Explicitly, T_N is equivalent to the "effective noise window", L_N is equivalent to the "propagation probability", and E_N is equivalent to the "noise propagation ratio". So the softness of an individual node N can be calculated as a product of the three factors:

$$S_N(L_N, T_N, E_N) = T_N * L_N * E_N \tag{5.1}$$

and the overall circuit vulnerability can be measured by a weighted sum over softness of all circuit nodes:

$$S_{total} = \sum_N w_N * S_N(L_N, T_N, E_N) \tag{5.2}$$

where w_N is a designer-specified weighting factor used to emphasize the functional significance of node N.

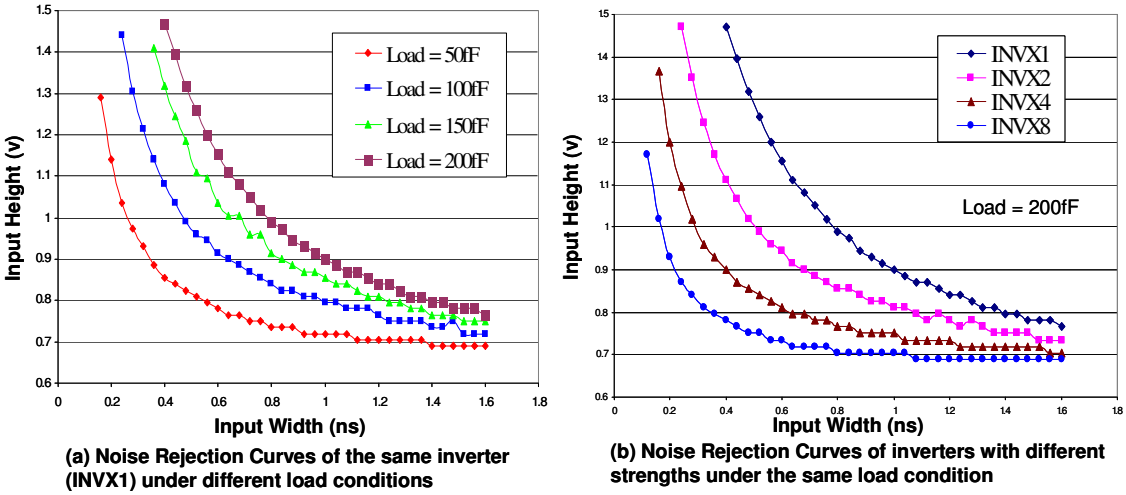


Figure 5-2 Noise Rejection Curves

Of the three factors, the electrical factor, which is characterized by the "sensitive region" on the "noise rejection curve (NRC)" of the driving gate, deserves further investigation. Its strength is dependent on two factors that can be controlled by the designer during the chip design phase. Therefore, it is possible to be optimized for stronger electrical masking. First, its strength is a decreasing function of the load capacitance at the gate output: the higher the load capacitance, the smaller the sensitive region and the weaker the electrical factor (Figure 5-2(a)). Second, its strength is an increasing function of the size of the gate: for gates of the same type, the larger the cell, the larger the sensitive region and the stronger the electrical factor (Figure 5-2 (b). In summary, the electrical factor is equal to the noise propagation ratio, which is a function of the gate size and load capacitance: $E_N = R_e^N(gatesize, load)$.

From equation (5.1) and (5.2), it is obvious that reducing one or more of the three factors will result in the reduction of the softness of a node. If the softness values of the

identified soft spots are reduced, the overall circuit transient error tolerance can be improved significantly. In the next two sections, two circuit-tuning techniques will be introduced. Their objective is to eliminate the soft spots by reducing one or more of the three factors.

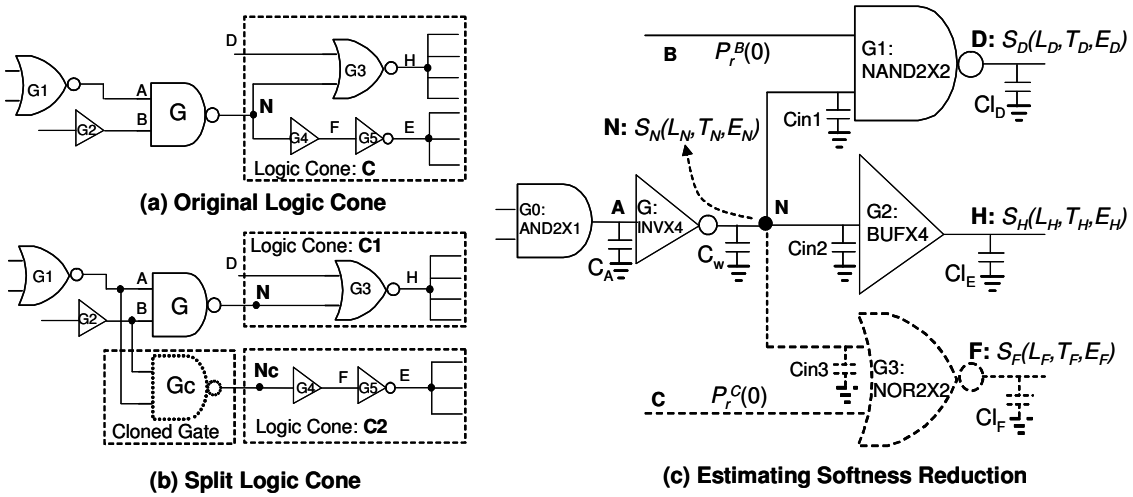


Figure 5-3 Gate Cloning

5.3 Gate Cloning

In the circuit segment shown in Figure 5-3(a), node *N* (output of gate G) drives a large logic cone C so a glitch at *N* can potentially reach many POs during a large timing window. If C is split into two smaller logic cones C1 and C2 (Figure 5-3 (b)), respectively driven by G and a newly created gate Gc that shares the same inputs as G, the new circuit is functionally equivalent to the original one but noise at *N* or N_C (the output of Gc) will have lower probability of causing functional errors at POs. This operation is called "gate cloning" because the new gate is identical to the original gate and shares part of its

functionality. Through gate cloning, a highly vulnerable circuit node is replaced by two (or more) less vulnerable nodes, causing the softness to be redistributed in the vicinity of the changed circuit. Gate cloning not only affects all three factors in softness of the involved circuit nodes but also change the delay of the affected paths.

First, the path delay is affected in two ways. On one hand, gate delays of G and Gc in Figure 5-3 (b) are both shorter than the delay of G in Figure 5-3 (a) because of driving less load. Denoting the delay of a gate G driving a total load of C (including the net capacitance and the pin capacitance of the fanout gates) as $t_G|_{Cload=C}$, the total delay decrease can be calculated as:

$$\delta t_G = t_G\Big|_{Cload=C_N} - \max\{t_G\Big|_{Cload=C_N'}, t_{G_C}\Big|_{Cload=C_{N_C}}\} \tag{5.3}$$

On the other hand, delays of driving gate G1 and G2 are increased because of the additional input capacitance of Gc. If the wire capacitance of net A and B are *CwA* and *CwB*, the pin capacitance of input A and B of gate G are *CinA* and *CinB*, the delay increase can be calculated as:

$$\delta t_{G1} = t_{G1}\Big|_{Cload=CwA+2*CinA} - t_{G1}\Big|_{Cload=CwA+CinA} \tag{5.4}$$

$$\delta t_{G2} = t_{G2}\Big|_{Cload=CwB+2*CinB} - t_{G2}\Big|_{Cload=CwB+CinB} \tag{5.5}$$

These timing changes should be compared against the available timing slack of the related timing paths. For example, if the minimum positive timing slack through *N* is Δt, the following condition has to be satisfied:

$$\Delta t \geq \max\{\delta t_{G1}, \delta t_{G2}\} - \delta t_G \tag{5.6}$$

Next, another example circuit is used to better illustrate how to estimate the softness changes. As shown in Figure 5-3 (c), node *N*, having softness $S_N(L_N, T_N, E_N)$, drives a logic cone through three fanouts G1, G2 and G3. Suppose G3 is removed from the logic cone as a result of splitting this logic cone, the changes in softness of the affected nodes need to be estimated.

First, the relation between the logic factor of node N and its fanout nodes (D, H, F) can be expressed as:

$$L_N = P_r^B(1) * L_D + L_H + P_r^C(0) * L_F \tag{5.7}$$

where $P_r^B(1)$ is the probability for node B carrying a logic 1, the non-controlling value of a NAND gate, and $P_r^B(1)*L_D$ is the probability for noise at *N* to propagate through gate G1; $P_r^C(0)$ is the probability for node C carrying a logic 0, and $P_r^C(0)*L_H$ is the probability for noise at *N* to propagate through gate G3. The single-input buffer G2 (BUFX4) does not change the probability for propagation. Similarly, the logic factor after G3 is removed from the logic cone is given by:

$$L_N' = P_r^B(1) * L_D + L_H \tag{5.8}$$

Second, the timing factor change can be illustrated by Figure 5-4. T_N can be derived from the timing factors of its fanout nodes and the respective gate delays. Let T_D, T_H and T_F be the sensitive windows whose start and end times are: $\{t_D^{start}, t_D^{end}\}$, $\{t_H^{start}, t_H^{end}\}$ and $\{t_F^{start}, t_F^{end}\}$, respectively. If the maximum and minimum delays of G1, G2 and

G3 are $\{d_{G1}^{min}, d_{G1}^{max}\}$, $\{d_{G1}^{min}, d_{G1}^{max}\}$ and $\{d_{G1}^{min}, d_{G1}^{max}\}$, respectively, the original noise sensitive window at node N has a start time and an end time:

$$(t_N^{start})=min\{t_D^{start}-d_{G1}^{max}, t_H^{start}-d_{G2}^{max}, t_F^{start}-d_{G3}^{max}\}$$

$$(t_N^{end})=max\{t_D^{end}-d_{G1}^{min}, t_H^{end}-t_{G2}^{min}, t_F^{end}-t_{G3}^{min}\}$$

and the noise sensitive window after removing G3 has a start time and an end time:

$$(t_N^{start})'=min\{t_D^{start}-d_{G1}^{max}, t_H^{start}-d_{G2}^{max}\}$$

$$(t_N^{end})'=max\{t_D^{end}-d_{G1}^{min}, t_H^{end}-t_{G2}^{min}\}$$

The old and new timing factors can be calculated as:

$T_N=(t_N^{end})-(t_N^{start})$ and $T_N'=(t_N^{end})'-(t_N^{start})'$

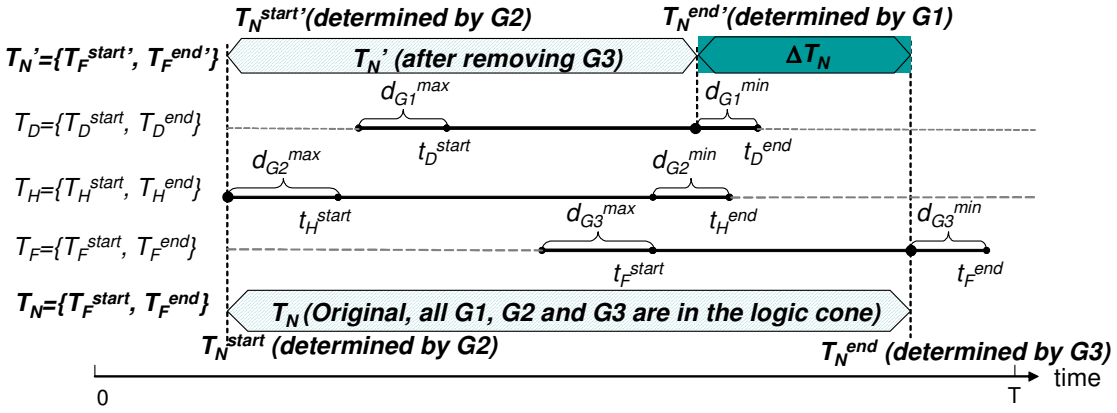


Figure 5-4 Timing window change

Third, the change in the electrical factor E_N due to logic cone split is caused by the change of load capacitance. Originally, $E_N=R_e^N(INVX1,C_{total})$, where $C_{total}=Cw+Cin1+Cin2+Cin3$, and Cw is the wire capacitance and $Cin1$, $Cin2$, $Cin3$ are input

capacitances of the three fanout gates. After removing G3, E_N' changes to $R_e^N(INVX1, C_{total}')$, where $C_{total}' = Cw + Cin1 + Cin2$.

In summary, the new softness after removing G3 from the original logic cone can be calculated as:

$$S_N'(L_N', T_N', E_N') = T_N' * L_N' * E_N' \tag{5.9}$$

A circuit node driving large logic cones may have many fanouts. If the number of fanouts is K, there are 2^K ways to split the logic cone. A proper logic cone partition is crucial to the effectiveness of the gate cloning. A procedure has been developed to search for a proper partition that optimally reduces S_N without causing timing violation, as described by the pseudo-code *SplitLogicCone* shown in Figure 5-5. The process *SplitLogicCone* has three arguments: the node *N*, a list of its fanout (*FanoutList*), and a target softness (S_{th}). Since the logic cone can not be split at node *N* if it has only one load, the first step (Line 1-5) is to forward trace the circuit from *N* to the first node *M* with multiple loads. If a PO is reached during the search, gate cloning is not applicable.

Next, the process attempts to split the logic cone into two partitions (Line 6): *NodeList1*, which will be driven by *N*, is initialized to the original fanout list; and *NodeList2*, which will be driven by the cloned node *Nc*, is initialized to an empty list. Then each iteration of the main loop (Line 8-31) attemps to move one node form *NodeList1* to *NodeList2*. The inner loop (Line 10-28) selects the node to be moved by evaluating the consequence of moving each remaining node in *NodeList1* to *NodeList2*: a move is invalid if it causes timing violations or the softness of the cloned gate to become larger than S_{th}; if the resulting softness of both *N* and *Nc* are less than S_{th}, the process

terminates successfully. If none of the above is true, a candidate move causing the largest softness reduction will be kept. As the inner loop terminates, both node lists are updated by moving the surviving candidate from *NodeList1* to *NodeList2* (Line 27-30). After the main loop is executed *K* times (the number of fanouts of node *N*), the process returns a failure if none of the tried partition will meet timing; otherwise, if no partition can be found to meet the condition in Line 19, the process calls itself to further split the logic cone (Line 34-35).

Gate cloning can reduce the logic and timing factor. However, it does not help to reduce the electrical factor. Therefore it can not be applied to the soft spots whose high softness values are caused primarily by a large E_N.

5.4 Cell Resizing

Cell resizing is to replace a logic gate by a functionally equivalent gate with difference size and driving strength. It can reduce the electrical factor and therefore the softness of a node. Based on the previous discussion, there are two ways to reduce E_N: driver downsizing and load upsizing. For example, reducing E_N in Figure 5-3(c) can be done by downsizing gate G from INVX4 to INVX2, or upsizing G3 from NOR2X2 to NOR2X4 or a combination of both. Both approaches have either positive or negative effects on the delays and softness of the affected circuit nodes. However, they do not affect the logic factors, and the effect on the timing factor is negligible. Using this example, the effects of driver downsizing and load upsizing can be explained.

(1) Downsizing gate G: INVX4→INVX2

```
SplitLogicCone (N, FanoutList, Sth)
1.  /* Tracing forward from node N to the node with multiple fanout. */
2.  while (sizeof(FanoutList)==1)
3.        M = The only fanout;
4.        if (M is a PO) return (-1);          /* gate cloning is not possible. */
5.       FanoutList = get_fanout(M);

6.  NodeList1 = FanoutList; NodeList2 = { };
7.  CurrentSn = Sn; TimingOk = 0; K=sizeof(FanoutList)

8.  for i = 1 to K begin          /* Outer Loop for logic cone split */;
9.        MaxDeltaSn = 0;

10.       foreach ThisFanout in {NodeList1} begin
11.          TempNodeList1 = {NodeList1} – ThisFanout;
12.          TempNodeList2 = {NodeList2} + ThisFanout;
13.          if ( check_timing(TempNodeList1, TempNodeList2) == 0 )
14.               continue;        /* Skip the rest if timing is not met */
15.          TimingOk = 1;        /* At least one partition does not fail timing */
16.          Sn1 = estimate_softness(NodeList1);
17.          Sn2 = estimate_softness(NodeList2);
18.          if ( Sn2 > Sth ) continue;  /* Should not create new soft spots */
19.          if (Sn1 <= Sth && Sn2 <= Sth)
20.             NodeList1 = TempNodeList1;
21.             NodeList2 = TempNodeList2;
22.             return;   /* Softness reduction goal achieved, done */
23.          if (CurrentSn – Sn1 – Sn2 > MaxDeltaSn)
24.             MaxDeltaSn = CurrentSn – Sn1 – Sn2;
25.            CandidateSn = Sn1 + Sn2; CandidateNode = ThisFanout;
26.       End /* inner loop */

27.       /* Update the partitions to result in the largest softness reduction */
28.       FanoutList1 = {FanoutList1} – CandidateNode;
29.       FanoutList2 = {FanoutList2} + CandidateNode;
30.       CurrentSn = CandidateSn;
31. End /* outer loop */

32. if ( TimingOk == 0 ) return (-1); /* No partition found, gate cloning failed */
33. /* Further logic cone split needed */
34. if (Sn1 > Sth) SplitLogicCone(M, NodeList1, Sn1, Sth);
35. if (Sn2 > Sth) SplitLogicCone(M, NodeList2, Sn2, Sth);
```

Figure 5-5 Pseudo-code: *SplitLogicCone*

The electrical factor E_N is reduced from $R_e^N(INVX4, C_{total})$ to $R_e^N(INVX2, C_{total})$, where $C_{total}=Cw+Cin1+Cin2+Cin3$ is the total load driven by G. Note that E_A, the

electrical factor of node A is increased as a side effect because of the reduction in C_A, the total load driven by gate G1. This change can be estimated similarly.

(2) Upsizing gate G3: NOR2X2 → NOR2X4

The overall delay change has two parts: δt_G and δt_{G3}. The δt_g is the increased delay of gate G because its total load capacitance is increased due to a larger input capacitance of NOR2X4 as compared to NOR2X2; and δt_{G3} is the decreased delay of gate G3 because NOR2X4 has shorter delay than NOR2X2 under the same load.

The electrical factor E_N is reduced from $R_e^N(INVX4, C_{total})$ to $R_e^N(INVX4, C_{total}')$, where C_{total}' is the total load of G after the upsizing, $C_{total}' > C_{total}$ because of the large input capacitance of NOR2X4 as compared to NOR2X2. Note that the E_F is increased as a side effect from $R_e^F(NOR2X2, Cl_F)$ to $R_e^F(NOR2X4, Cl_F)$, where Cl_F is the total load driven by gate G3.

From real experiences, driver downsizing is more effective than load upsizing. The softness reduction is more significant and the perturbation on other node is smaller. It can also reduce the total cell area.

Figure 5-6 shows the process to search for the solution that requires minimal circuit change to achieve the desired softness reduction. Given a soft spot N, its softness S_N and a target softness value S_{th} ($S_{th} < S_N$), the driver downsizing is first tried. Three conditions have to be met for the downsizing to be feasible: (1) the driver of N has to be "downsize-able", i.e., a smaller gate has to exist in the cell library; (2) the timing requirement of related paths has to be satisfied; and (3) the softness values of all related

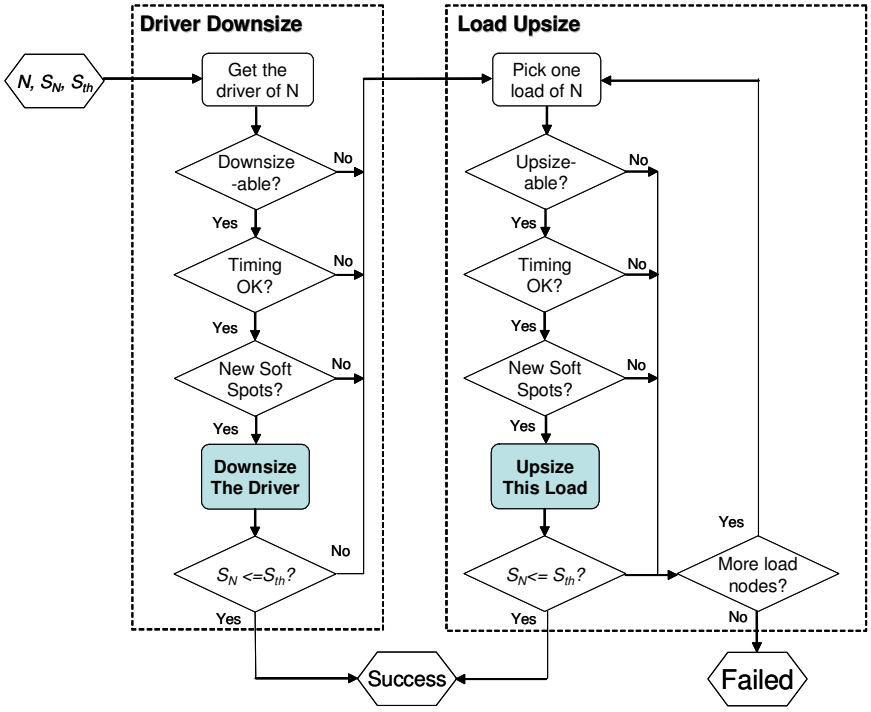


Figure 5-6 Cell Resizing flow

nodes has to remain below S_{th}. The driver gate is then downsized if all three conditions are met. If the resulting S_N is reduced to below S_{th}, the process terminates successfully ("**Done**"). The load upsizing process is entered when any of the three conditions is not satisfied or the driver downsize is feasible but S_N is still larger than S_{th}. It is similar to the driver downsizing and go through every load of N. The process terminates successfully as soon as the softness reduction goal is reached. If S_N can not be reduced to below S_{th} when all load gates have been processed, the process terminates with a "**Failed**" flag.

5.5 Robustness Closure

A technique aiming at SET tolerance enhancement must have no negative impact on the design's timing closure and physical layout. Otherwise, the circuit changes will

incur further timing or layout problems that can only be resolved by more design iterations. The best way to maximize their efficiency is to integrate them with the design flow.

As shown in Figure 5-7, a typical digital design process starts with the "Front-end Design", including logic/circuit design and functional verification, followed by the "Back-end Design", including logic synthesis, physical layout and timing closure. Its primary objective is to produce a functioning design that meets certain speed goal with

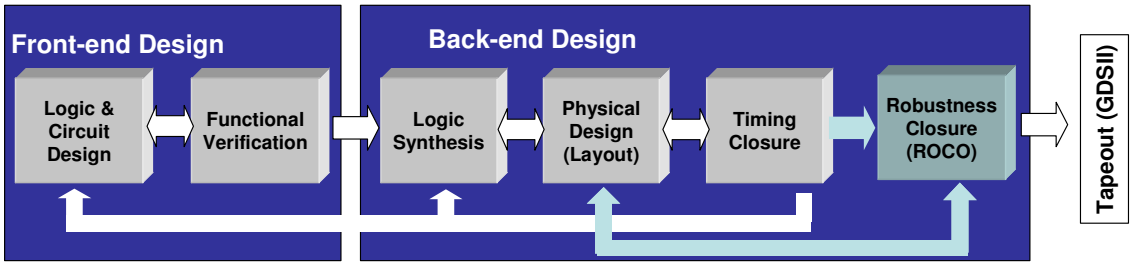


Figure 5-7 Robustness Closure in design flow

minimum area. As the noise-tolerance is not an optimization criterion, certain circuit elements in the timing-closed design might be still highly vulnerable. Therefore, an additional optimization step ("Robustness Closure") to fix these vulnerabilities is necessary. It can reuse the existing analysis results obtained from the other steps in the flow (such as static timing analysis results, extracted RC information, etc.). The RObustness COmpiler (ROCO) was constructed based on these ideas.

ROCO is implemented in TCL environment with its key engines written in C. It can be directly executed from TCL-based STA tools, such as PrimeTime. This seamless

interface can improve its performance as it not only takes advantage of the speed and accuracy of commercial tools, but also provides direct feedback to these tools.

As shown in Figure 5-8, ROCO requires three inputs: a "*Timing-Closed Design*"; a "*Design Analysis Report*", which contains information of the design such as timing, area and RC parasitic; and a "*Robustness Specification*" which defines the desired robustness level using a "soft spot ratio" (R_S).

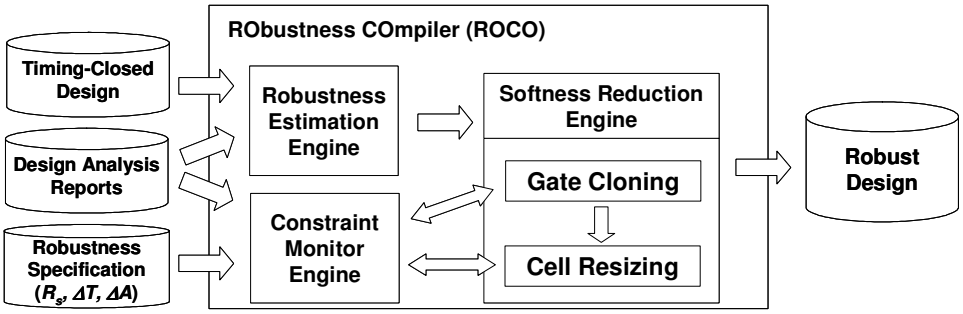


Figure 5-8 RObustness COmpiler (ROCO)

The heuristic R_S specifies a certain percentage of circuit nodes with highest softness values as soft spots and determines the threshold S_{th}. The goal is to reduce softness of all circuit nodes to below S_{th}. The "*Robustness Specification*" also provides the allowed timing (ΔT) and area (ΔA) overhead. The ΔT will allow ROCO to increase the system clock period from T to $T+\Delta T$ in order to meet the robustness specification. ΔT can be zero if speed degradation is not acceptable. The ΔA specifies the maximum allowed area increase. Determining R_S, ΔT and ΔA is beyond the scope of this work.

ROCO first uses a "*Robustness Estimation Engine*" based on the soft spot analysis to evaluate the softness of all circuit nodes. Then it generates a collection of soft spots with $S_N > S_{th}$. Next, the "*Softness Reduction Engine*" uses the "*Gate Cloning*" and "*Cell Resizing*" engines to reduce the softness of the identified soft spots. Gate cloning is applied first because it ensures convergence of the methodology: the softness reduction achieved by gate cloning will not be reversed by the cell resizing. These processes are concurrently checked by the "*Constraint Monitor Engine*" to ensure that no design constraint is violated. Finally ROCO generates a "*Robustness-Closed Design*".

5.6 Experimental Results

This section will demonstrate the efficiency and discuss some distinguished features of ROCO via several experiments.

First, it needs to be verified that both techniques can effectively reduce improve circuit SET tolerance when individually applied on the design by SPICE simulations. Due to the speed and capacity limitations of HSPICE, only small circuits can be simulated to prove the concept. Each experiment circuit was timing-closed, post-layout netlist. ROCO was run to obtain the modified design after gate cloning (CKT-gc) and cell resizing (CKT-cr). SPICE simulations were then run on CKT, CKT-gc and CKT-cr. During the simulation, a large number of glitches with random shape and timing were injected only to the affected nodes. The number of errors observed at circuit POs were counted and compared among all three circuits.

Two experiment circuits were used: CKT1 with 73 nodes and CKT2 with 192 nodes. R_S was set low (CKT1: 8%, CKT2: 2%) to limit the number of soft spots to reduce the simulation time. Among the 6 soft spots in CKT1, 5 nodes were cloned and 4 cells were resized; among the 4 soft spots in CKT2, 4 were cloned and 2 cells were resized. Table 5-1 and Table 5-2 shows the results: the column "Glitch" lists the number of glitches injected during the SPICE simulation; the column "Error" lists the number of errors observed at the POs; and the column "E.R." is the error rate (Error/Glitch). As shown in the last column, both the gate cloning and cell resizing were able to reduce the error rate significantly (from 41% to 61%).

Table 5-1 SPICE Simulation Results: Gate Cloning

	Original Circuit (CKT)			**Post Cloning (CKT-gc)**			**Error**
	Glitch	Error	E.R.	Glitch	Error	E.R.	**Reduction**
CKT1	25600	2464	**9.6%**	66560	3124	**4.7%**	**51%**
CKT2	2048	388	**18.9%**	4096	297	**7.3%**	**61%**

Table 5-2 SPICE Simulation Results: Cell Resizing

	Original Circuit (CKT)			**Post Cell Resizing (CKT-cr)**			**Error**
	Glitch	Error	E.R.	Glitch	Error	E.R.	**Reduction**
CKT1	32768	4439	**13.5%**	32768	2004	**6.1%**	**55%**
CKT2	8192	823	**10.0%**	8192	483	**5.9%**	**41%**

Next, it is to be proved that given realistic constraints, ROCO can effectively and economically improve circuit reliability. In addition to CKT1 and CKT2, one large circuit CKT3 with 3156 circuit nodes was also used.

Table 5-3 lists the robustness specifications: the number of nodes (2nd col.), the soft spot ratio R_s (3rd col.) and the threshold S_{th} (4th col.), the clock period T (5th col.) and the allowed timing overhead ΔT (6th col.). R_s of the two small circuits were set to 20% and R_s of the large CKT3 was set to 2%. Because timing is a more important performance concern than area, ΔT was set to be 0 and relaxed the area overhead constraints.

Table 5-3 Robustness specifications

	Nodes	R_s	S_{th}	T(ns)	ΔT(ns)
CKT1	73	20%	1.75	1.8	0
CKT2	192	20%	1.09	2.4	0
CKT3	3156	2%	3.2	8.0	0

Table 5-4 shows the experiment results: for each circuit the three rows lists the data of the original circuit (Orig.), circuit after gate cloning (G.C.) and cell resizing (C.R), respectively. The five columns show the total softness S_{total}, the maximum softness $(S_N)_{max}$, the number of identified soft spots N_{ss} (the number of nodes whose softness values is larger than S_{th}), the total softness of the soft spots $(S_N)_{ss}$, and the area with percentage change. Next, this experiment is discussed from several aspects.

First, it can be seen that a small number of nodes with high softness values contribute to the majority of the circuit vulnerability. For example, the top 2% nodes (63) in CKT3 make up to 31% (499/1623) of the total softness.

Table 5-4 ROCO execution results

		S_{total}	$(S_N)_{max}$	N_{ss}	$(S_N)_{ss}$	Area (μm^2)
CKT 1	**Orig.**	53.88	6.24	15	27.36	2039
	G.C.	50.26	3.24	4	9.30	2175(+6.7%)
	C.R.	45.59	2.08	2	3.95	2095(+2.7%)
CKT 2	**Orig.**	73.17	2.75	38	34.31	5172
	G.C.	72.14	1.38	8	8.5	5415(+4.7%)
	C.R.	68.87	1.04	0	0	5252(+1.5%)
CKT 3	**Orig.**	1623	46.62	63	499	83915
	G.C.	1441	10.3	12	72.1	85645(+2.1%)
	C.R.	1408	5.8	6	22.9	85269(+1.6%)

Second, ROCO can significantly reduce the number of soft spots and their softness: the number of soft spots (N_{ss}) in CKT1 was reduced from 15 to 4 after GC and to 2 after CR; in CKT2, N_{ss} was reduced to 8 after GC and to zero after CR. For CKT3, N_{ss} was reduced to 12 and 6, respectively; and $(S_N)_{ss}$ was reduced by 95.4% (499 to 22.9).

Third, the total softness S_{total} only slightly decreased (e.g. by 13%, from 1623 to 1408, in CKT3), because the softness of some neighboring nodes may be increased. It can be viewed as "softness redistribution", as illustrated in Figure 5-9. On the left, the x-axis is the softness values in log-scale and the y-axis is the number of nodes with different softness. The top-ranked soft spots correspond to the small tail toward the right end, which shrinks after each step. The number of nodes with intermediate softness values increases slightly. The right graph zooms in on the 63 soft spots on the right tail, where the actual softness (y-axis) is plotted for all circuit nodes (x-axis). It clearly shows how much the softness of the soft spots is reduced.

Fourth, it is not always possible to eliminate all soft spots under given constraints

and budgets. In the end, 2 nodes in CKT1 and 6 nodes in CKT3 still have softness values above S_{th}. For further improvement, more design overhead should be allowed.

Finally, given all required inputs, ROCO finished within ~23 seconds for the small CKT1 and CKT2; for the large CKT3, the runtime is ~678 seconds. ROCO is fast because (1) it is static and does not need dynamic simulation; (2) it takes advantage of the high performance of commercial static timing analysis tool; and (3) it only focus on a limited number nodes.

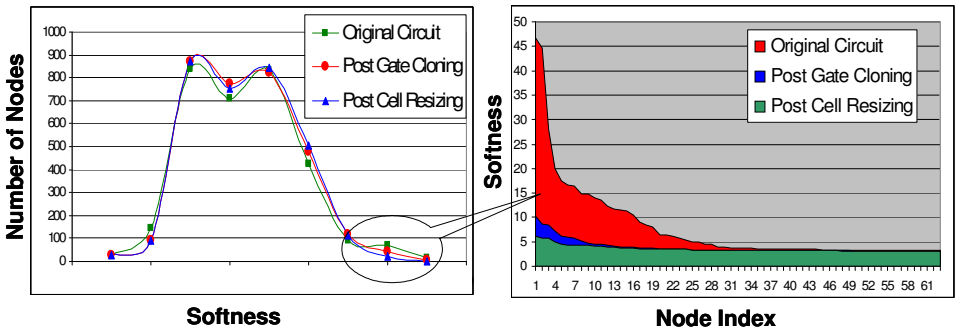


Figure 5-9 Softness redistribution in CKT3

5.7 Conclusion

In this chapter, a robustness closure engine "RObustness COmpiler" was presented. It efficiently improves circuit reliability using two localized circuit modification techniques: gate cloning and cell resizing. Seamlessly integrated with existing design flow, it will greatly facilitate the design of reliable nanometer circuits.

Until now, a complete framework of analysis and design of highly robust nanometer circuits have been established. The soft spot analysis and the noise impact

analysis can first locate the vulnerability in the combinational as well as sequential circuits. The results of these analyses can be then used to guide the reliability improvement of the sequential (through robustness insertion) and combinational (through robustness compiler) circuits. The outcome is a highly robust circuit system that is developed at very limited design cost and effort.

However, as mentioned in previous chapters, various types of interferences co-exist in nanometer circuits that can interact with each other to aggravate their error effects, as the crosstalk noise considered in soft spot analysis. The next chapter will discuss another type of variation that can also lead to higher circuit vulnerability, which is the process variation.

5.8 Acknowledgement

Chapter 5 is based on material in the published paper: Chong Zhao, Sujit Dey, "Improving Transient Error Tolerance of Digital VLSI Circuits Using RObustness COmpiler (ROCO)", in Proceedings of 7th International Symposium on Quality Electronic Design (ISQED), pp. 133-138, March 2006, San Jose, California, USA. The dissertation author was the primary investigator and author of this paper.

Chapter 6. Transient Error Analysis Considering Process Variations

The previous chapters have been focused solely on radiation-induced soft errors, one particular type of environmental variations that may cause reliability degradation of nanometer circuits. With the aggressive technology scaling, process variations are also becoming an increasingly significant factor causing the reliability degradation. Specifically, the presence of the process variation can significantly aggravate the transient soft error effects. Consequently, transient error analysis without considering the impact of process variations will be overly optimistic. In this chapter, a statistical model is developed to analyze the transient error generation and propagation considering the inter-die channel length variation. Experiment results have demonstrated that channel length variation can significantly aggravate the soft error effect, which can be quickly and accurately evaluated using the proposed model.

6.1 Introductions

Nanometer circuits are becoming increasing susceptible to various variations. The transient error discussed throughout the previous chapters is one particular type environmental variations. There exists another category of variation, i.e. the process variation. As technology continues to scale, the control of critical device parameters is becoming more difficult as process geometries shrink. As a result, significant variations

in device length, doping concentrations, and oxide thicknesses have resulted, causing large uncertainties in design performance. Process variations can be classified as inter-die and intra-die variations. Inter-die variation refers to the different features of the same device on a chip among different dies; intra-die variation refers to the variation of the device parameters among different locations on the same die. The impacts of process variations on timing and power have been realized and extensively researched. Statistical timing analysis (SSTA) has been adopted to address the variability of path delay due to process variation [76][77]. Its basic approach is to model the delay as a random variable propagated along a timing path to obtain the joint delay distribution. Statistical approaches of power analysis and optimization considering process variation has also been developed [78] [79]. Process variations may also aggravate the soft error effects. This, unfortunately, has not been thoroughly investigated.

The objective of this chapter is to model the impact of variations in device channel length on the radiation-induced transient error effects. The modeling consists of two steps. First, the channel length variation in the struck transistor may cause the strength of the generated glitch to have a wide distribution, which can be modeled by a random variable. Next, as the transient glitch propagate in the circuit netlist, the channel length variations in the encountered gates may further extend the distribution, which can be modeled as transformations of the random variable. In reality, the developed mode of transient generation and propagation is not applicable directly to large circuit systems for two reasons. First, the model requires heavy computation tasks for the struck gate as well as every encountered gate during the transient propagation. Second, the modeling error will accumulate during the propagation. Therefore, this chapter will also illustrate how to

apply the model of transient generation and propagation to digital circuits effectively and accurately.

The discussion in this chapter will be narrowed to the following scope:

1) Only inter-die variation in channel length is considered. The methodology can be extended to intra-die variations as well as variations in other design parameters with reasonable efforts.

2) Only a strike by a particle with fixed energy levels is considered. In reality, natural particle activity exhibits strong time, altitude and geographical dependencies. This variation is described by the particle's energy spectrum and can be taken into consideration in the model as a second random variable.

The rest of the chapter is organized as follows: section 6.2 reviews the basic mechanisms of single event transients in digital circuits; section 6.3 models the impact of inter-die channel length variations on single event transient generation and propagation; section 6.4 presents simulation results; and section 6.5 concludes this chapter.

6.2 Modeling the Single-Event-Transient (SET)

In Chapter 1, the basic mechanisms of radiation-induced soft error have been reviewed. The *Linear Energy Transfer* (LET) was introduced to relate the energy of the incident particle to the charge deposition in a particular type of material. When a particle with the energy *LET* strikes the sensitive region of a CMOS transistor (usually the drain

of the transistor), the collected charge will cause a double-exponential transient current flow in at the p-n junction contact:

$$I(t) = \frac{Q}{(\tau_\alpha - \tau_\beta)}(e^{-t/\tau_\alpha} - e^{-t/\tau_\beta}) \tag{6.1}$$

where $Q=LET^* \lambda_c * q_d$, is the total deposited charge; τ_a is the "collection time constant"; and τ_β is the "ion track establishment time constant". Typical values are approximately 1.64×10^{-10} sec for τ_α and 5×10^{-11} sec for τ_β.

Figure 6-1 shows the scenario of a particle striking the PMOS transistor in an inverter and the corresponding linear RC modeling. The transient current *I(t)* can be modeled as a voltage controlled current source, with the independent voltage source V_{ctrl} provides the double-exponential term. R_D is the equivalent resistor of the NMOS transistor network. When the input is a stable 1, the PMOS transistor, which is in its OFF state, may be temporarily shorted, causing a transient voltage pulse *V(t)* to appear at the gate output. In [48], an approximate closed-form expression of *V(t)* was obtained:

$$V(t) = \frac{Q}{C_n \tau_\alpha} e^{-t/\tau_n} \left(\frac{e^{t/\tau_n} \cdot e^{-t/\tau_\alpha} - 1}{1/\tau_n - 1/\tau_\alpha}\right) \tag{6.2}$$

where $\tau_n = C_n * R_D$; C_n is the total capacitance (sum of the output capacitance C_o and the load capacitance C_l). In equation (6.2), the contribution of τ_β is ignored as it is relatively small compared to τ_α.

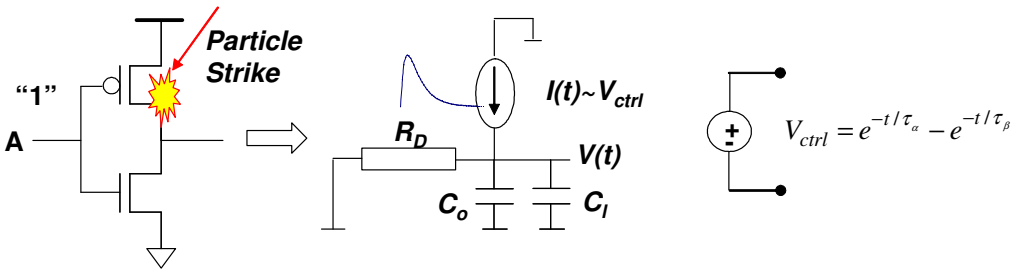


Figure 6-1 Single-Event Transient Modeling

By solving $dV(t)/dt = 0$, it can be derived that $V(t)$ reaches its peak at time t_M:

$$t_M = (\frac{\tau_n \tau_\alpha}{\tau_n - \tau_\alpha}) \ln(\frac{\tau_n}{\tau_\alpha}) \tag{6.3}$$

and the peak value V_G is given by:

$$V_G = V(t_M) = \frac{Q}{C_n}(\frac{\tau_n}{\tau_\alpha})^{\frac{\tau_\alpha}{\tau_\alpha - \tau_n}} = \left[\frac{\lambda_c \cdot q_d}{C_n}(\frac{\tau_n}{\tau_\alpha})^{\frac{\tau_\alpha}{\tau_\alpha - \tau_n}}\right] \cdot LET \tag{6.4}$$

Equation (6.4) indicates that V_G is proportional to the incident energy. However, experiment shows that V_G can be better fit into a quadratic function of *LET*. For example, when a particle strikes the PMOS transistor in the inverter INV1 (Figure 6-2 (a)), V_G at its output is plotted against the incident LET (Figure 6-2 (b)). This can be explained as follows: when the particle energy is low, the dominant mechanism of the strike is the funneling effect, based on which (4) is derived; as the incident energy increases, the ion track shunting becomes significant [80], causing V_G to deviate from the straight line. As LET further increases to LET_{MAX}, V_G saturates to the supply voltage (V_{dd}). Considering the boundary condition that $V_G=0$ at $LET=0$, V_G can be expressed as:

$$V_G(LET) = \begin{cases} k \cdot LET \cdot (LET + q) & LET < LET_{MAX} \\ V_{dd} & LET \geq LET_{MAX} \end{cases} \quad (6.5)$$

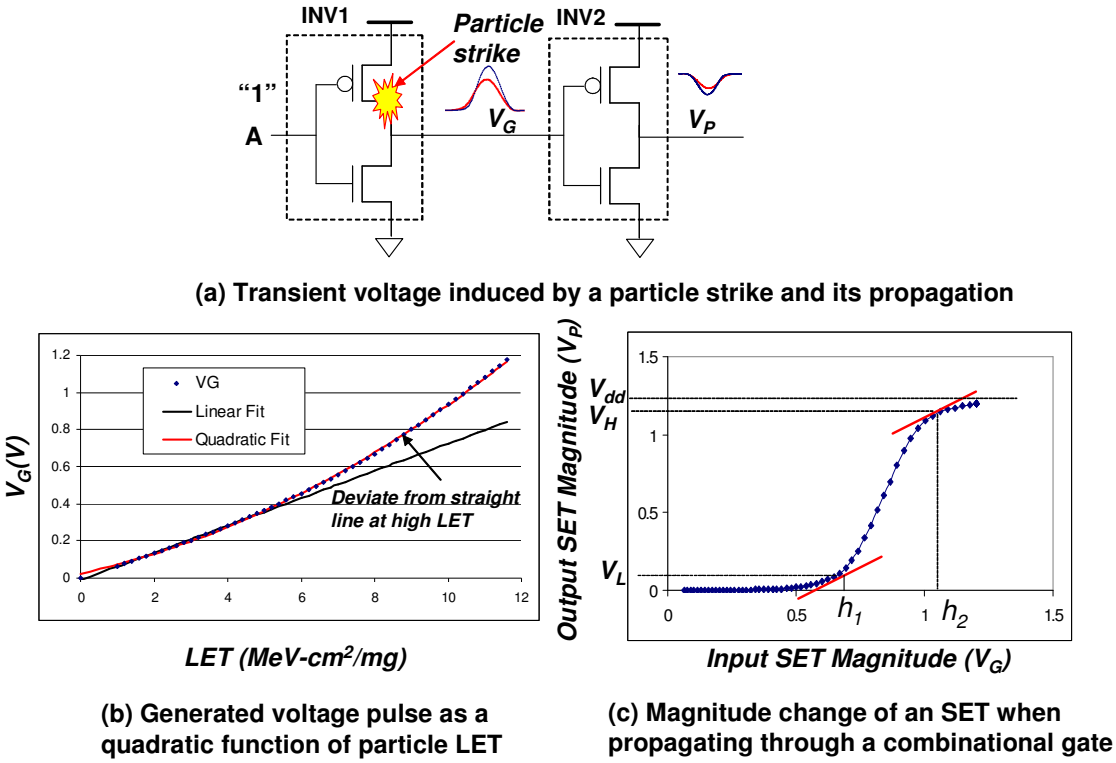


Figure 6-2 SET Generation and Propagation

The saturation case is much easier to handle since the transient will be able to further propagate without attenuation until it reaches the FFs. Hence the focus should be set on the case $LET<LET_{MAX}$ (so $V_G<V_{dd}$). For the current and future technology node, typical LET_{MAX} values are in the range of 10~20, depending on the type of the struck gate, loading condition and supply voltage.

As the transient generated at the output of INV1 propagates through the next gate INV2 in Figure 6-2 (a), its magnitude change is determined by INV2's DC characteristics (Figure 6-2 (c)). Note that V_P, the magnitude of the negative pulse is measured as its

maximum deviation from V_{dd}. If $V_G < h_1$, V_P is attenuated to below a low threshold V_L; the transient is considered completely "masked" by INV2 and stops further propagation; if $V_G > h_2$, V_P is amplified to be above a high threshold V_H and the transient will be further propagated as a full-swing voltage pulse; if $h_1 \le V_G \le h_2$, V_P is considered to be linear to V_G. The region $[h_1, h_2]$ is called the "transition region" on the DC characteristic. In summary, V_P can be expressed as a piece-wise linear function of V_G:

$$V_P = \begin{cases} 0 & V_G < h_1 \\ \alpha \cdot V_G + \beta & h_1 \le V_G \le h_2 \\ V_{dd} & V_G > h_2 \end{cases} \tag{6.6}$$

where h_1 and h_2 are called the "unity gain point" in the inverter's DC characteristics plot [84]. Since V_P is continuous at h_1 and h_2, α and β are solely determined by h_1 and h_2 as:

$$\alpha = V_{dd} /(h_2 - h_1), \quad \beta = -h_1 /(h_2 - h_1) \cdot V_{dd} \tag{6.7}$$

The values of the parameters k, q, h_1, h_2 in (6.5) and (6.6) are all functions of the channel length l. If the channel length is a random variable, these parameters can be modeled as functions of random variables. Consequently, both the generated transient V_G and the propagated transient V_P become random distributions.

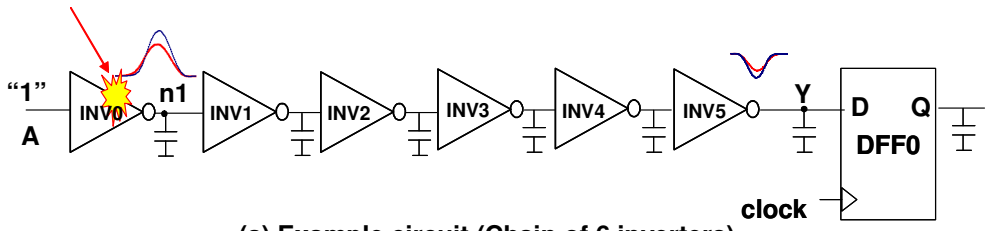
6.3 Modeling Transient Generation and Propagation Considering Channel Length Variation

This section first uses one example to show that the presence of channel length variation can have significant impact on transient error effects (section 6.3.1), thus it is imperative to consider the variation in the analysis; then the model of transient generation

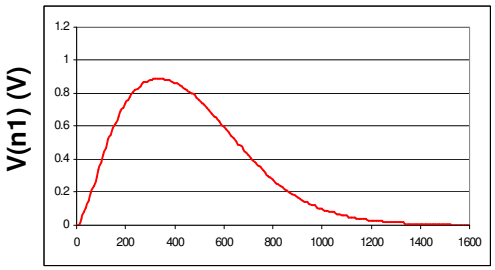
and propagation will be developed in section 0 and 6.3.3 respectively; and the same example will be revisited to demonstrate how to use the model (section 6.3.4). For convenience, the model development will be illustrated using inverters as examples.

6.3.1 An Example: Inverter Chain

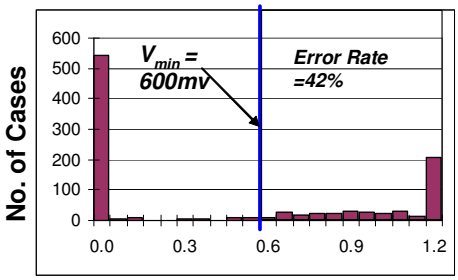
The example circuit in Figure 6-3(a) consists of 6 identical inverters driving a flip-flop DFF0. All gates are from a 0.13µm COMS cell library. The input A is 1 when a particle with *LET=10 MeV-cm²/mg* strikes the PMOS transistor in INV0. If the channel length *l* of all inverters are of nominal value l_0=0.13µm, the transient at the struck node n1 has a magnitude V_G=893.7mV (Figure 6-3 (b)). Simulation shows that it will be attenuated by the inverter chain and will not be able to reach node Y.



(a) Example circuit (Chain of 6 inverters)



(b) Generated and propagated transient at nominal channel length (l_0=0.13µm)



(c) Histogram of transient heights at node Y under normally distributed channel length

Figure 6-3 An Inverter Chain Example

However, assuming the channel lengths of the inverters observes a normal distribution ($\mu_l=l_0$ and $3\sigma_l=30\%*l_0$), Monte Carlo simulation of 1000 samples was run to produce the histogram of the transient magnitude at Y. As shown in Figure 6-3 (c), in a large number of samples, a transient is able to reach Y with finite magnitude. If DFF0 is sensitive to a 600mV input, a latched error will be observed in DFF0 in 42% of all samples. Therefore, the presence of channel length variation can reduce the noise margin of the circuit and worsen the transient error effect. Consequently, any error analysis will be significantly inaccurate without considering process variations.

6.3.2 Modeling Transient Generation

When a particle strikes a combinational gate, the strength of the induced transient varies with the channel length. Figure 6-4(a) shows the voltage waveform at the output of INV1 in Figure 6-2(a) when the gate length varies, from where it can be seen that V_G increases with l. This is caused by the changes in the parameters k and q in (6.5). It has been observed from experiment that q is relatively independent of l, hence V_G can be expressed as:

$$\begin{aligned} V_G(l) &= k(l)\cdot LET\cdot(LET+q) = \left[k(l_0)+\left.\frac{\partial k}{\partial l}\right|_{l=l_0}\cdot(l-l_0)\right]\cdot LET\cdot(LET+q) \\ &= \left[\left.\frac{\partial k}{\partial l}\right|_{l=l_0}\cdot l + (k(l_0)-\left.\frac{\partial k}{\partial l}\right|_{l=l_0}\cdot l_0)\right]\cdot LET\cdot(LET+q) \\ &\equiv (a_G\cdot l + b_G)\cdot LET\cdot(LET+q) \equiv K\cdot l + K_b \end{aligned} \tag{6.8}$$

In (6.8), only the first-order term in the Taylor expansion is kept because the channel length variation is typically small. So V_G is a linear function of l for fixed LET

value, which has been verified by simulations for several LET values in Figure 6-4(b). Note that from the boundary condition that $V_G=0$ when $l=0$ requires that b_G should be zero.

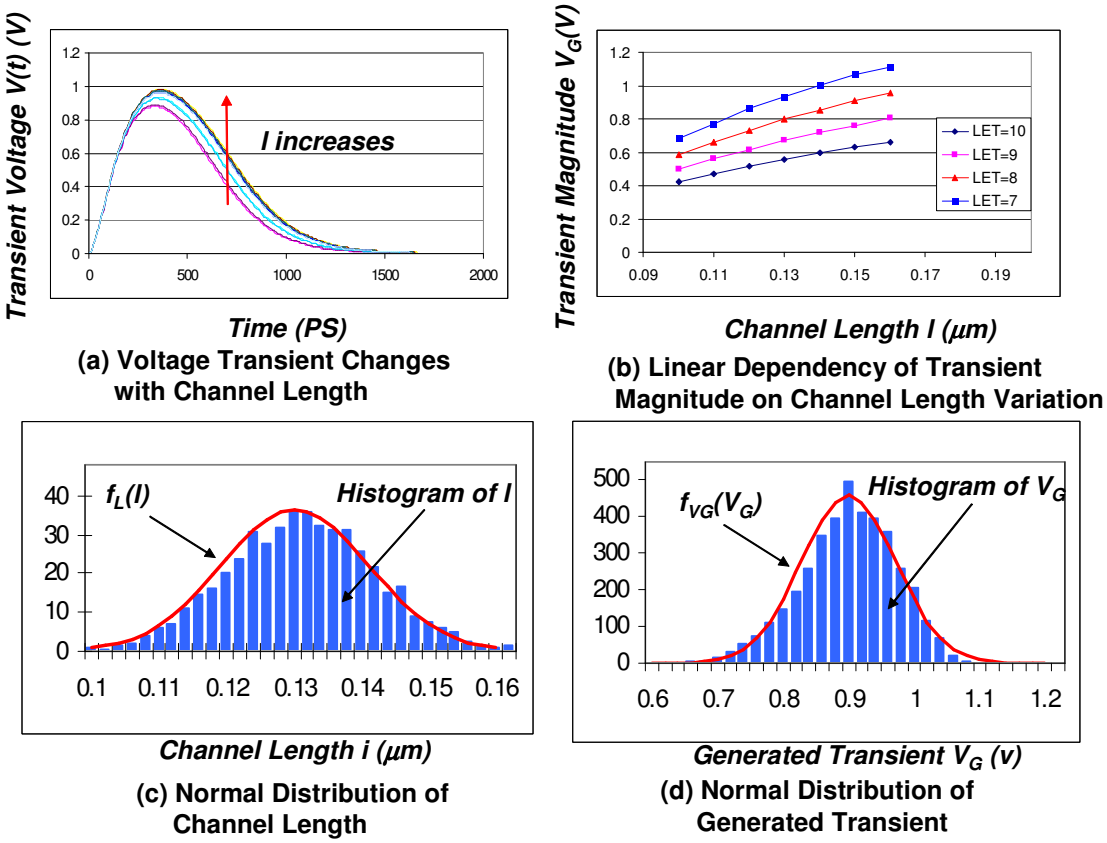


Figure 6-4 Modeling the Variation of Transient Generation

The inter-die channel length variation is generally considered to be a normal distribution $N(\mu_l, \sigma_l^2)$, whose probability density function (PDF) is:

$$f_L(l) = \frac{1}{\sqrt{2\pi}\sigma_l}\exp(-\frac{(l-\mu_l)^2}{2\sigma_l^2}) \tag{6.9}$$

From the rules of "functions of random variables" [83], it can be proved that V_G also has a normal distribution $N(\mu_G, \sigma_G^2)$, whose mean and standard deviation are:

$$\mu_G = V_G(l_0), \sigma_G = K \cdot \sigma \tag{6.10}$$

where $V_G(l_0)$ is the transient magnitude when $l=l_0$. It has been observed from Monte Carlo simulation that if the normal distribution of channel length in INV1 is as shown in Figure 6-4 (c), V_G has a distribution as shown in Figure 6-4 (d). This will be numerically validated in the experimental result section.

6.3.3 Modeling Transient Propagation

It has been proved that the generated transient V_G has a normal distribution due to channel length variation in INV1. When V_G propagates through the inverter INV2 (Figure 6-2 (a)), the output V_P is given by equation (6.6). If INV2 has nominal channel length $l=l_0$, the distribution of V_P can be considered in three regions:

1) $V_G<h_1$, $V_P=0$: the probability is given by the cumulative density function (CDF) of V_G, i.e. $P(V_P=0) = F_{VG}(h_1)$;

2) $h_1 \le V_G \le h_2$, $V_P=\alpha \cdot V_G+\beta$: This is the transition region. V_P, as a linear function of V_G, has a normal distribution: $P(V_P)=N(\mu_P, \sigma_P^2)$, with mean $\mu_P=\alpha \cdot \mu_G+\beta$ and standard deviation $\sigma_P=\alpha \cdot \sigma_G$;

3) $V_G>h_2$, $V_P=V_{dd}$: the probability is given by to the probability of $V_G>h_2$, i.e. $P(V_P=V_{dd}) = 1-F_{VG}(h_2)$.

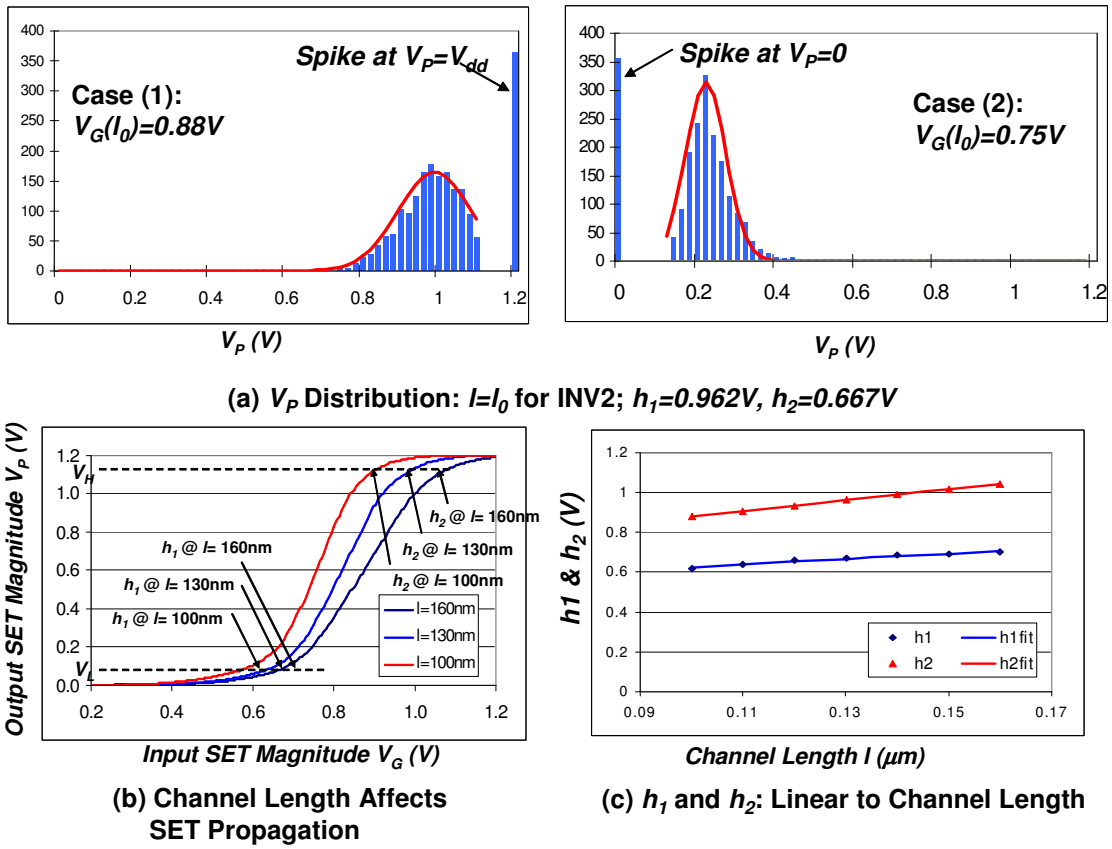


Figure 6-5 Modeling the Variation of Transient Propagation

Figure 6-5 (a) shows two scenarios of the transient propagation. In both cases, $h_1=0.667V$, $h_2=0.962V$, and the nominal $V_G(l_0)$ are in the transition region ($h_1 \le V_G(l_0) \le h_2$), so the nominal V_P is determined by the linear relation in (6.6). However, the variations in INV1 cause a certain portion of V_G to reach beyond the transition region. In case (1), $V_G(l_0)=0.88V$, close to the upper limit of the transition region, for some values of l, $V_G(l)$ will exceeds h_2, causing V_P to be saturated to V_{dd}. This results in a discontinuity in V_P distribution and a spike at $V_P=V_{dd}$. Similarly, in case (2), $V_G(l_0)=0.75V$, close to the lower limit of the transition region, the discontinuity in V_P distribution is on the left side and the

spike appears at $V_P=0$. In both cases, the V_P distribution in the transition region remains normal. This will be numerically validated in the experimental result section.

The above discussion is valid when INV2 has nominal channel length. In reality, the variation in INV2 affects its DC characteristic. As shown in Figure 6-5 (b)), both h_1 and h_2 increase with channel length l. For small variations in l, both h_1 and h_2 are approximately linear functions of l, as supported by the experiment results shown in Figure 6-5 (c):

$$\begin{aligned} h_1(l) &= a_1 \cdot l + b_1 \\ h_2(l) &= a_2 \cdot l + b_2 \end{aligned} \tag{6.11}$$

Hence, both h_1 and h_2 have normal distributions. This variation will cause further deviation in V_P distribution. When considering inter-die variation, the channel lengths of INV1 and INV2 can be treated as 100% correlated random variables. Finding the V_P distribution is equivalent to solving the following problem:

"Given a random variable l, whose PDF is defined in (6.9), and three linear functions $V_G(l)$, $h_1(l)$ and $h_2(l)$ of l, defined in (6.8) and (6.11), find the distribution of V_P as defined in (6.6), which is a function of random variables V_G, h_1 and h_2."

The above problem can be solved by examining the three regions in (6) separately:

1) $V_G<h_1$: $V_P=0$. This probability is given by:

$$\begin{aligned} P(V_P = 0) &= P(V_G - h_1 < 0) = P((K - a_1) \cdot l - (K_b - b_1) < 0) \\ &= P(l < (K_b - b_1)/(K - a_1)) = F_L((K_b - b_1)/(K - a_1)) \end{aligned} \tag{6.12}$$

2) $V_G > h_2$: $V_P = V_{dd}$. This probability is given by:

$$P(V_P = V_{dd}) = P(V_G - h_2 > 0) = P((K - a_2) \cdot l - (K_b - b_2) > 0)$$
$$= P(l > (K_b - b_2)/(K - a_2)) = 1 - F_L((K_b - b_2)/(K - a_2)) \quad (6.13)$$

3) $h_1 \le V_G \le h_2$: V_P is given by:

$$V_P = V_{dd} \cdot (V_G - h_1)/(h_2 - h_1)$$
$$= V_{dd} \cdot [(K - a_1) \cdot l - (K_b - b_1)]/[(a_2 - a_1) \cdot l - (b_2 - b_1)] \equiv g(l) \quad (6.14)$$

and its distribution can be expressed as:

$$P(g(l) < V_P) = P(l < l_v) \cdot P(h_1 \le V_G \le h_2)$$
$$= F_L(l_v) \cdot [F_L((K_b - b_2)/(K - a_2)) - F_L((K_b - b_1)/(K - a_1))] \quad (6.15)$$

where l_v is a function of V_P, defined as:

$$l_v(V_P) = \frac{(b_2 - b_1) \cdot V_P - (K_b - b_1)}{(a_2 - a_1) \cdot V_P - (K - a_1)} \quad (6.16)$$

Therefore, the complete distribution of V_P is defined by equations (6.12), (6.13) and (6.15), and can be obtained by calculating $F_L(l)$, the CDF of channel length variation. This fully characterizes the propagation behavior of a transient, and the further propagation can be considered in the same manner. Although straightforward, it is computation-intensive so it is impractical to apply to a large circuit. The next section will revisit the example circuit in Figure 6-3 (a) and illustrate how to practically use the model in circuit analysis.

6.3.4 Case Study: Inverter Chain

As the transient further propagates, variations in all encountered gates can change its distribution, resulting in a wide range of the probability of its capture by the endpoint FF. To iteratively calculate the distribution along the propagation path is extremely time-consuming. However, considering only inter-die variation (i.e. the channel length of different gates are 100% correlated random variables), the problem can be greatly simplified as it is possible to first compute the transient after it propagates through all gates before considering the variations in all gates on the path.

This section applies the transient generation and propagation models to the inverter chain in Figure 6-3 (a). For the convenience of illustration, it is assumed that these inverters have equal pull-up and pull-down strength, which means that h_1 and h_2 in (6.6) are the same for positive and negative transients.

First, the transient observed at node Y when node n1 is struck by a particle with fixed LET ($LET<LET_{MAX}$) is computed. The transient at INV1 output $V_P^{(1)}$ is given by (6.6). The transient at INV2 output $V_P^{(2)}$ can be calculated as by applying (6.6) to $V_P^{(1)}$:

$$V_P^{(2)} = V_{dd} \cdot \frac{V_P^{(1)} - h_1}{h_2 - h_1} = V_{dd} \cdot \frac{V_{dd} \cdot (V_G - h_1)/(h_2 - h_1) - h_1}{(h_2 - h_1)} = \frac{V_G - h_1 \cdot (1+R)}{R^2}$$

where $R=(h_2-h_1)/V_{dd}$. Note that the above equation holds only when $V_P^{(1)}$ is in the transition region of INV2: $h_1 \le V_P^{(1)} \le h_2$, or $(1+R) \cdot h_1 \le V_G \le h_1 + R \cdot h_2$. It is easily proved that $(1+R) \cdot h_1 > h_1$, and $h_1 + R \cdot h_2 < h_2$. So the transition region of the 2 inverters is narrower than that of a single inverter. In other words, the interval during which the transient

propagates linearly is shrinking. Repeating the computation, the transient magnitude at node Y in the transition region can be obtained:

$$V_P^{(N)} = \frac{V_G - h_1 \cdot [1 + R + R^2 + \cdots + R^{N-1}]}{R^N} = \frac{1}{R^N} \cdot (V_G - h_1 \cdot \frac{1 - R^N}{1 - R})$$

where N is the logic depth from the struck node to the endpoint FF0 ($N=5$ in the example). More importantly, the transition region on the DC characteristic of the inverter chain can be expressed as:

$$h_1 \cdot \frac{1-R^N}{1-R} < V_G < h_1 \cdot \frac{1-R^N}{1-R} + R^N \cdot V_{dd}$$

Generally speaking, the transition region of a well-designed inverter is very narrow, i.e. R is very small, so R^N decreases rapidly as N increases. In the example, $R=0.27$ when $l=l_0$, so $R^N \sim 1.4x10^{-3}$ for $N=5$. As N becomes large, and the transition region converges to a single point: $(V_G)_{th}=h_1/(1-R)$. This means that V_Y, the transient at Y, is either 0 or V_{dd} depending on the magnitude of the generated transient V_G at the struck node:

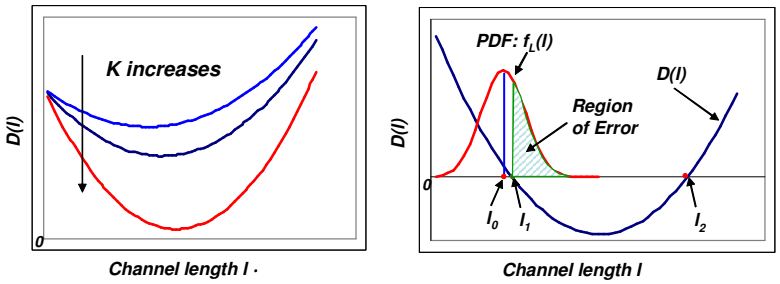
$$V_Y = \begin{cases} 0, & V_G < h_1/(1-R) \\ V_{dd}, & V_G \geq h_1/(1-R) \end{cases}$$

Next, the effect of channel length variation is analyzed for this particular example. Since V_G, h_1 and $R=(h_2-h_1)/V_{dd}$ are all linear functions of the channel length l, when n1 is struck, the probability for the generated transient of *not* being able to reach Y is equal to the probability that: $V_G(l)-[h_1(l)/(1-R(l))]<0$. This condition can be rewritten as a quadratic function of l:

$$D(l) \equiv (K \cdot a_3) \cdot l^2 + (a_1 - K \cdot b_3^{'}) \cdot l + b_1 > 0$$

where $K=a_G \cdot LET \cdot (LET+q)$, $a_3=a_2-a_1$, $b_3'=1-b_3=1-(b_2-b_1)$; and a_G, q are as defined in (6.8); a_2, a_1, b_2, b_1 are as defined in (6.11). It will be shown later in the experimental result section that all the parameters are positive and $(a_1-K \cdot b_3')<0$. The discriminant of $D(l)$ is a quadratic function of K:

$$\Delta_l(K) = (a_1 - K \cdot b_3')^2 - 4 \cdot K \cdot a_3 \cdot b_1 = b_3^{'2} \cdot K^2 - (2 \cdot b_3^{'} \cdot a_1 + 4 \cdot a_3 \cdot b_1) \cdot K + a$$



(a) Case 1) – $\Delta_l(K)<0$: $D(l)>0$; V_G at $n1$ can not reach Y

(b) Case 3) – $\Delta_l(K)>0$: $D(l)>0$; V_G at $n1$ can reach Y if $l_1<l<l_2$

Figure 6-6 Transient Error Probability Dependency on *LET* and channel length (*l*)

Now several possible situations will be discussed separately:

1) $\Delta_l(K)<0$ (Figure 6-6 (a)). This means that $D(l)>0$ for all values of l because the coefficient of l^2 $(K \cdot a_3)$ is positive. The discriminant of the quadratic $\Delta_l(k)$ is given by:

$$\Delta_K = (2 \cdot b_3^{'} \cdot a_1 + 4 \cdot a_3 \cdot b_1)^2 - 4 \cdot a_1^{\ 2} \cdot b_3^{'2} \quad = 16 \cdot (a_1 \cdot b_1 \cdot a_3 \cdot b_3^{'} + a_3^{\ 2} \cdot b_1^{\ 2})$$

Obviously, Δ_K is always positive because a_1, b_1, a_3 and b_3' are all positive. Therefore, $\Delta_l(K)$ always has two real zeros $K_{1,2}$ and since the coefficient of the K^2 is positive ($b_3'^2>0$), the range of K for $\Delta_l(K)<0$ is $K_1<K<K_2$. In reality, it can be proved that

$K>K_1$ is always true because $K<K_1$ causes the two zeros of $D(l)$ to be negative. Therefore, if $K<K_2$, $\Delta_l(K)<0$, and $D(l)$ is always positive. This means that for any value of channel length, the transient at n1 will not be able to propagate to Y. Knowing that $K=a_G \cdot LET \cdot (LET+q)$, it can be concluded that if the particle's energy is sufficiently small, the incurred transient will not reach Y regardless of the actual channel lengths of the inverters. Also, $D(l)$ shifts toward the x-axis as K increases and will intersect with x-axis when K reaches certain value.

2) $\Delta_l(K)=0$. This determines minimum value K_{th} for $D(l)$ to have real zeros. Correspondingly, it determines LET_{th}, the threshold of the particle's LET for causing a transient to reach Y.

3) $\Delta_l(K)>0$ (Figure 6-6 (b)). This requires that the incident particle's LET is sufficiently high. $D(l)$ has two real zeros $l_{1,2}$:

$$l_{1,2} = \frac{(K \cdot b_3^{'} - a_1) \pm \sqrt{\Delta_l(K)}}{2 \cdot K \cdot a_3}$$

It is obvious that $D(l)>0$ when $l<l_1$ or $l>l_2$. In other words, a strike at n1 can reach Y only when $l_1<l<l_2$. Experiment has shown that l_2 is typically much greater than the realistic upper bound of the possible channel length. Therefore, P_e, the probability of an error at Y, is equal to the probability of the channel length being greater than l_1, or $P_e=1-F_L(l_1)$, where $F_L(l)$ is the CDF of the channel length distribution. Note that l_1 is a function of K (and therefore a function of LET). As the incident LET increases, K increases and the curve of $D(l)$ shift downwards, which causes the root l_1 to decrease. Consequently, the

error probability $P_e=1-F_L(l_1)$ increases with LET because $F_L(l_1)$ is a monotonously increasing function of l_1. This means that a strike by a particle with higher energy will cause higher probability of error.

Using the conclusion derived above, the distribution in Figure 6-3 (c) can be explained as following. In the example, $l_1>l_0$ (as indicated in Figure 6-6 (b)), so at nominal channel length l_0, V_G caused by a strike of $LET=10$ at n1 is not enough to propagate to node Y. With the variation in channel length, V_G will be able to reach Y with magnitude V_{dd} in the samples where l surpasses l_1; in samples where l remains below l_1, Y should remain undisturbed. However, there are still a small number of samples in which V_Y is less than V_{dd}. This is not consistent with the prediction that the transient at node Y is either 0 or V_{dd}, depending on the incident energy. It is because the transition region is still of a finite width in the experiment since the logic depth N is not large enough.

6.4 Experimental Results

This section presents some experimental results to support the modeling. It first validates and characterizes the model of transient generation and propagation, which is then applied to the inverter chain example.

All simulations are performed in SPICE on one type of inverter cell in a 0.13μm cell library, custom-designed using the Predictive Technology Model (PTM) for bulk CMOS [82] [83]. During the simulation, a particle strike is realized by a voltage-controlled current source between the V_{dd} and the output node (for a strike on the PMOS transistor) or the output and the ground (for a strike on the NMOS transistor), as shown in

Figure 6-1. The magnitude of the current is controlled by the independent voltage source V_{ctrl}, which provides the double-exponential term in equation (6.1).

6.4.1 Modeling of Transient Generation and Propagation

The development and characterization of the model involves extensive simulation efforts. Many of the results have been illustrated graphically in the previous sections. This sub-section simply present numerical results.

First, the inverter is characterized to determine all parameters in the model (Table 6-1). The parameters are cell-specific, and dependent on many factors, including the load capacitances, supply voltage, as well as the operating conditions. The values listed in Table 6-1 are for the inverter driving a loading capacitance of 50fF in normal operation conditions, and the V_{dd} is set to 1.2V.

To determine the values of a_G and b_G in (6.8), the channel length l is first fixed and the transistor is disturbed by particles with LET values ranging from 0 to LET_{MAX}, the coefficients k and q in (6.5) are obtained by fitting the transient magnitude V_G to a quadratic function for the specific value of l. Figure 6-2 (b) shows V_G can be well fit into the quadratic function of LET. Then the channel length l is varied from 100nm to 160nm to find different values of k as a function of l. The parameter a_G and b_G is then determined by linear fitting. Note that due to the boundary condition that $V_G=0$ at $l=0$, b_G is negligibly small (~10^{-4}).

To determine a_1, b_1, a_2 and b_2 in (6.11), the experiment circuit was set up as shown in Figure 6-2 (a). Again, the channel length l is fixed when the transistor in INV1

was attacked by particles with LET from 0 to LET_{MAX}, the coefficients h_1 and h_2 are measured from the DC characteristics of INV2. Then l is varied from 100nm to 160nm to find different values of h_1 and h_2 as functions of l. The parameters a_1, b_1, a_2 and b_2 are determined by linear fitting.

Table 6-1 Parameter Characterization

Parameter	Value	Unit
a_G	0.0293	$V\cdot\mu m^{-1}\cdot(MeV\cdot cm^2/mg)^{-2}$
b_G	0.0001	$V\cdot(MeV\cdot cm^2/mg)^{-2}$
q	11.92	$MeV\cdot cm2/mg$
a_1	1.368	-
b_1	0.489	m
a_2	2.732	-
b_2	0.607	m
a_3	1.364	-
b_3	0.118	m
b_3'	0.882	m

Next, the normal distribution of V_G is verified; and its mean and standard deviation is determined. The incident LET is fixed at 10 MeV·cm^2/mg and Monte Carlo simulation was run using 1000 samples with normally distributed channel length (μ_l=*0.13*μm, $\sigma_l=\mu_l\cdot 10\%$=0.013μm), as shown in Figure 6-4 (c). The histogram and fitted normal distribution of V_G are plotted in Figure 6-4 (d), and the mean and deviation are shown in the first two rows in Table 6-2, where the "Calculation" data is obtained from equation (10), which is only 5.1% (for μ) and 6.9% (for σ) different from the data measured from simulation.

Table 6-2 Modeling SET Generation and Propagation: Distribution of V_G and V_P

			Simulation	Calculation	|Error|
(1) SET Generation		Mean (μ_G)	938.9mV	893.7mV	**5.1%**
		Deviation (σ_G)	77.7mV	83.5mV	**6.9%**
(2) SET Propagation	***Case (1)***	Mean (μ_P)	990mV	923mV	**6.8%**
		Deviation (σ_P)	96.7mV	106.1mV	**9.8%**
		$P(V_P=V_{dd})$	18.2%	20.1%	**10.4%**
	Case (2)	Mean (μ_P)	220mV	241mV	**9.6%**
		Deviation (σ_P)	50.7mV	55.4mV	**9.3%**
		$P(V_P=0)$	17.8%	19.3%	**8.5%**

Next, the distribution of the propagated transient through INV2 is verified when its channel length is fixed at l_0. The result is shown in the lower part of Table 6-2. In both cases in Figure 6-5 (a), the mean, deviation in the transition region calculated using the model are compared to the results measured from simulation. The probability of $V_P=V_{dd}$ in case (1) and $V_P=0$ in case (2) are al compared. Although the relative errors of the transient propagation model remain reasonably low (6.8%~10.4%), they are higher than the error of the transient generation modeling, because the modeling error in V_G distribution accumulates to the distribution of V_P. This gives another reason that repetitively calculating the distribution of the transient at the output of each logic gate is not an optimal approach. Instead, the approach proposed in section 6.3.4 should be adopted, where the propagation through multiple logic levels is considered before the distribution of the final transient is calculated.

6.4.2 Case Study: Inverter Chain

Now the SET generation and propagation model will be applied to the inverter chain example. Using the data in Table 6-1, the three cases in section 6.3.4 can be detailed as:

1) $\Delta_l(K)<0$ if $K_1<K<K_2$, where $K_{1,2}$ are the two zeros, and $K_1=0.392$ and $K_2=6.14$. Note that $K_1<K$ always holds to guarantee at least one of the two zeros $l_{1,2}$ of $D(l)$ is positive. Solving $K=a_G \cdot LET \cdot (LET+q)$ for LET with $K=K_2$, one positive real zero $LET_2=9.695$ (the other root LET_1 is negative) is obtained. So if the incident particle's LET is less than LET_2, the resulting transient at n1 will not reach Y, regardless of the channel length.

2) $\Delta_l(K)=0$ if $K=K_{th}=6.14$. This determines the value of the threshold LET that might cause an error: $LET_{th}=9.695$.

3) $\Delta_l(K)<0$ if $K>K_{th}$: for any $LET>LET_{th}$, $D(l)$ has two real zeros $l_{1,2}$. And the transient will reach node Y when $l1<l<l2$. In other words, for any value of l, there exists a minimum value LET_{min} such that when $LET>LET_{min}$, the transient will reach Y.

Two experiments were performed on this inverter chain example. The first experiment found the values of LET_{min} for different channel lengths. The results are shown in Figure 6-7 (a). The "Simulation" data is obtained by gradually increasing the LET value for a fixed l until an error is observed in DFF0 during the simulation. The relative error between the simulation and calculation is on average 5.6%.

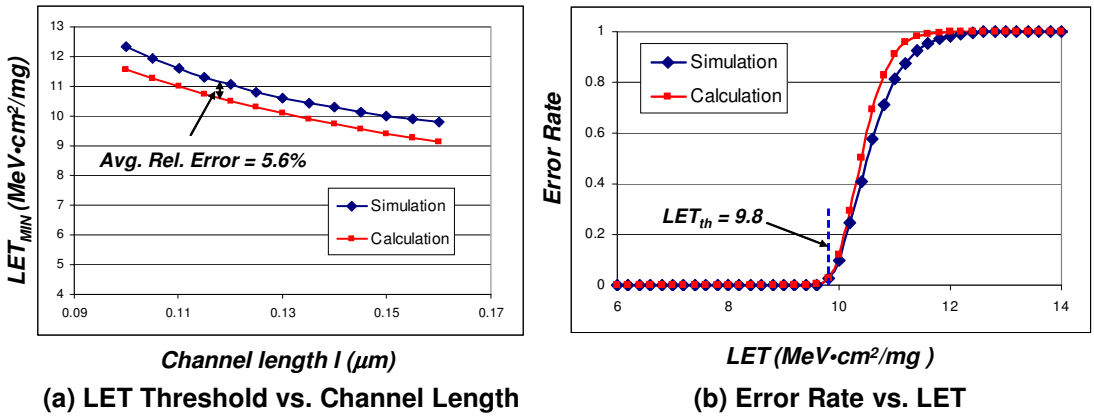


Figure 6-7 Experiment Results of the Inverter Chain Example

The second experiment measured the error rate in DFF0. The result is shown in Figure 6-7(b), where the error rate is plotted against LET. The "Calculation" data is obtained by first determining the value of l_1 for a fixed LET, then calculating the error rate as $1\text{-}F_L(l_1)$. The "Simulation" data is obtained from SPICE Monte Carlo simulation: for each LET value, 10,000 samples of varying channel length are used and the number of errors observed in DFF0 is counted. It can be seen that the model's prediction matches the experimental result very well. Specifically, the simulation data showed that the error rate remains zero until $LET=9.8$ MeV·cm2/mg, whereas the predicted threshold value is $LET_{th}=9.695$ MeV·cm^2/mg.

From these two experiments, it is demonstrated that when applying the statistical model, the impact of inter-die channel length variation on the transient error rate can be accurately predicted and the analysis accuracy is not degraded as the transient propagates further along the logic path.

6.5 Conclusions

This chapter developed a statistical model of the generation and propagation of radiation-induced single event transient with the presence of inter-die channel length variation. The analysis has shown that process variation can significantly aggravate the error effect of environmental variations, and experiments proved that the proposed model can accurately compute the varying error rate.

As the first effort to address the problem, this model only normally distributed inter-die channel length variation. Intra-die variation, on the contrary, has strong spatial correlations and does not follow a simple normal distribution, which will complicate the statistical model. Also, the activities of cosmic particles should be included in the model as another random variable. How to incorporate both factors into the model constitutes a major task in the ongoing research.

6.6 Acknowledgement

Chapter 6 is based on material to be published in the Proceedings of International Conference on Computer Design (ICCD), 2007, Lake Tahoe, California, USA: Chong Zhao, Sujit Dey, “Modeling Soft Error Effects Considering Process Variations”. The dissertation author was the primary investigator and author of this paper.

Chapter 7. Conclusion and Future Research Direction

This chapter serves two purposes. Section 7.1 provides a high-level summary of the presented methodologies and techniques; it also concludes the major contributions of research work in this dissertation. Section 7.2 discusses the open issues, the limitations of the research work and gives the future research directions in the relevant field.

7.1 Summary of Research Contributions

The primary focus of the research work presented in this dissertation is developing methodologies in the analysis, design and optimization of highly reliable nanometer circuits and systems. There are many different sources of reliability issues in nanometer circuits. This dissertation use one of them: the radiation-induced transient single-event-upset (SEU), as the primary noise model. Its brief history, basic mechanisms, error impacts on VLSI circuits and previous related research have been reviewed in Chapter 1. From Chapter 2 to Chapter 5, four static techniques (*Soft Spot Analysis, Noise Impact Analysis, Robustness Insertion and Robustness Enhancement*) are elaborated in great detail. They serve different purposes toward a single design goal; their functionalities are closely related and interdependent; and they can expedite the execution and improve the performance of each other. Together they form a unified *reliability optimization framework*, as shown in Figure 7-1:

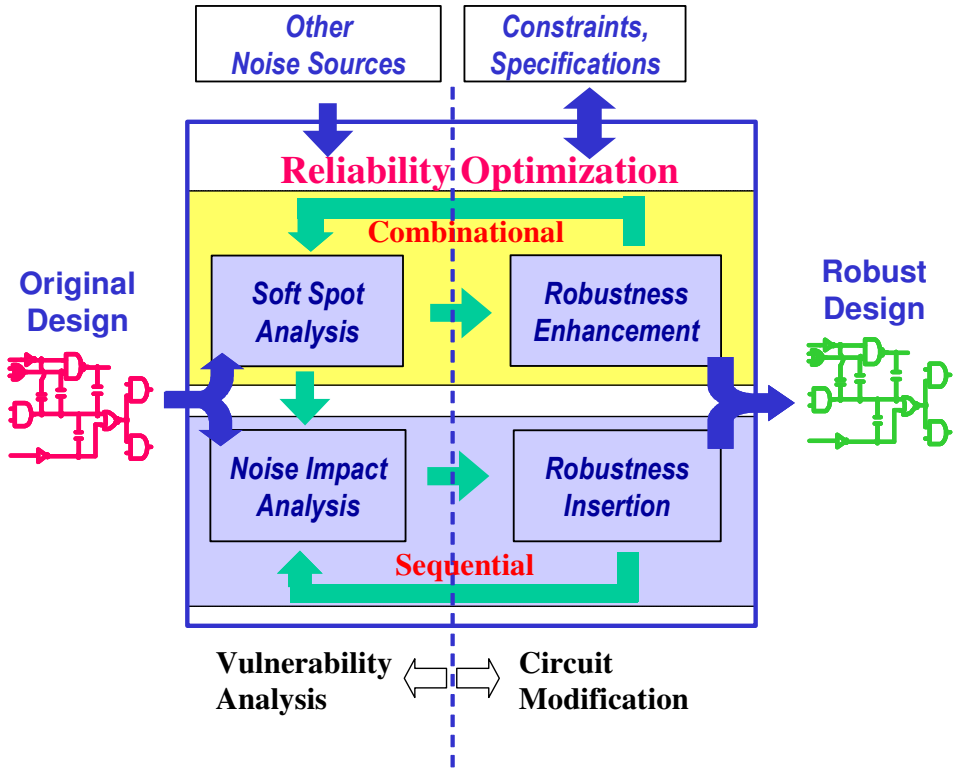


Figure 7-1 The Integrated Framework of Reliability Optimization

Given a design that is timing-closed but still highly vulnerable to transient errors, the reliability optimization framework first quickly and accurately identifies the vulnerabilities (in both the combinational and sequential logics) through soft spot analysis and noise impact analysis. Information on other type of noise sources (process variation, crosstalk, etc.) can also be incorporated into the analysis to improve the accuracy and applicability of the results. Next, in order to fix these vulnerabilities, it performs necessary circuit modifications and inserts proper protections into the sequential circuits (robustness insertion) and combinational circuits (robustness enhancement). During the circuit modification, design specifications and constraints are constantly and closely monitored to keep the associated overhead cost within acceptable range. If necessary,

more iteration can be performed by feeding back the modified circuit through the analysis and modification. A robust circuit with high transient error tolerance and minimum design penalty is produced as the output of the framework. This framework is the most important contribution of the dissertation. It has several distinguished features that make it an efficient and economical flow.

1) The soft spot analysis and noise impact analysis are static reliability analyzing techniques. They can quickly identify the most vulnerable circuit elements (both combinational and sequential) and provide an overall vulnerability map of the design. Neither technique requires extensive dynamic simulations or intensive computations. Therefore, they are fast, efficient and scalable to large complicated circuit systems.

2) Guided by the analysis results of the two reliability analyzing techniques, the robustness insertion and robustness enhancement techniques can be applied to the locations that need improvement and protection the most. Both techniques utilize constraint-aware optimizations to ensure they are applied without causing unacceptable overheads. Using these two techniques, great reliability improvement can be achieved with limited design efforts and cost.

3) This framework can be smoothly interfaced with the existing chip design flow. On one hand, it takes the advantage of the accurate analysis results (such as static timing analysis and RC extraction) from commercial CAD tools. On the other hand, its outputs can be directly applied back to the design flow. This seamless

interface further improves the efficiency of all individual components and minimizes the reliability improvement efforts.

4) Similar to the timing-closure flow, circuit vulnerabilities may not be able to be completely eliminated in a single trial. Therefore, several iterations are needed. The proposed framework forms a self-contained flow so the optimization goal can be gradually approached by iterative execution of one or more of the techniques.

5) This framework is extensible. By incorporating information on other noise sources, it can optimize the circuit to become more resilient to multiple noise sources and their compound effects. This extensibility has been demonstrated by jointly considering process variations and crosstalk effects.

In summary, the proposed reliability optimization framework can greatly facilitate the analysis, design and optimization of nanometer VLSI circuits.

7.2 Direction of Future Research

7.2.1 Improvement of the Reliability Optimization Framework

The reliability optimization framework is a prototype of the integrated and automated flow that takes reliability as its primary optimization target. It still has many aspects that need further improvement:

1) This framework is a standalone engine. It takes the timing-closed design (which might still have serious reliability concerns) as an input and produces an improved design with higher transient error tolerance as an output. In spite of a

smooth interface with various commercial CAD tools, its relative isolation from the existing design flow may degrade its efficiency. It may incur unnecessary efforts in meeting the design constraints, which may require extra work in timing closure and physical layout. Also, it may not produce the optimal design due to the limited exploration space left in the timing-closed design. Therefore, a complete integration with the existing design flow is the next major task.

2) This framework is focused mainly on the single event transient (SET) in combinational logics. In order to make the framework complete, both types of transient errors need to be addressed. The reliability analysis should draw conclusion based on the aggregated error rate and the transient error mitigation techniques must be able to improve the circuit tolerance to both types of transient errors.

3) Each component of the framework has its own limitations and defects that need to be further improved. Examples include over-simplified assumptions or model abstractions, simplification or approximation of the algorithms, and sub-optimal results due to the lack of optimizations. They have been discussed in detail in the conclusion section of the individual chapters.

7.2.2 Reliability-Cost Metric

The semiconductor industry is facing a big dilemma. On one side, it is becoming extremely difficult to ensure the chip can reliably perform the functions it is designed to so achieving high reliability is a high-cost task. On the other side, cost is becoming the greatest threat to the continuation of the semiconductor roadmap and an indispensable

requirement in mainstream applications. A paradigm shift to relaxing the requirement of 100% correctness may dramatically reduce costs of manufacturing, verification, and test. In other words, reliability requires design tradeoffs and the ultimate goal is to provide the best value for the reliability cost spent. This subsection introduces a new direction of developing a reliability-cost metric that can lead to the automatic determination of the optimal point in such tradeoffs.

Cost related to design reliability can be divided to two parts: "cost of protection (*CoP*)" and "cost of failure (*CoF*)".

CoP refers to the cost of reaching a proper level of reliability. It includes (1) C_{NRE}: the non-recurring engineering cost required to perform the reliability optimization tasks. C_{NRE} can be estimated from the additional engineering cost and the lengthened design cycle. (2) C_{PP}: the performance penalty (per die) incurred by the redundancies for protection purpose. C_{PP} can be estimated from the performance penalty due to the lowered clock frequency (less operations per unit time), the increased die size (higher manufacturing and packaging cost), the higher power consumption, and etc. If the total number of chips to be shipped is *N*, *CoP* can be expressed as:

$$CoP = C_{NRE} + N \cdot C_{PP} \tag{7.1}$$

CoF refers to the total operation loss due to the operation failure (caused by the non-protected vulnerabilities). If the "mean time to failure" (MTTF) of the chip is *m*, the life expectancy is *M*, and the average operation loss of each failure is C_{FAIL}, *CoF* can be expressed as:

$$CoF = (M / m \cdot C_{FAIL}) \cdot N \tag{7.2}$$

C_{FAIL} is a subjective measurement heavily depending on the nature of the application. The majority of applications belong to one of the three scenarios:

a) $C_{FAIL} = \infty$: for mission critical applications, such as space/military applications and life-saving devices, the cost of a transient failure is too high to be measured by dollars. The only goal is to achieve the highest possible reliability;

b) $C_{FAIL} = 0$: for many consumer applications, such as electronic toys and MP3 players, an occasional transient glitch will most probably cause no damage, extra protections are virtually unnecessary; and

c) $0 < C_{FAIL} < \infty$: for some non-critical applications, such as stock trading and automatic banking systems, an occasional glitch may cause loss to some degree. Therefore, a certain level of reliability needs to be ensured. The total reliability cost can be expressed as:

$$C_{TOTAL} = CoP + CoF = (C_{NRE} + N \cdot C_{PP}) + N \cdot (M / m \cdot C_{FAIL}) \tag{7.3}$$

C_{TOTAL} is a function of several cost factors (C_{NRE}, C_{PP}, C_{FAIL}) as well as non-cost factors (M, N, m). M and N are determined by the product specifications and market requirements; C_{NRE} and C_{FAIL} can be determined by empirical studies. C_{PP} is related to MTTF: $C_{PP}=C_{PP}(m)$, because in order to reach a long MTTF, more design overhead is necessary. Hence CoP and CoF are both functions of MTTF. This can be explained using the reliability-cost curve in Figure 7-2. The CoP curve starts with a base MTTF m_0, i.e.

the MTTF without any additional protection, and increases almost exponentially with MTTF. The *CoF* curve is inversely proportional to MTTF. As a sum of the *CoP* and *CoF*, the C_{TOTAL} curve reaches its lowest point C_{min} at MTTF=m_{min}, which can be determined by solving $dC_{TOTAL}/dm=0$. Theoretically, given the exact form of $C_{PP}(m)$, the optimal MTTF value m_{min} and the corresponding C_{min} can be calculated.

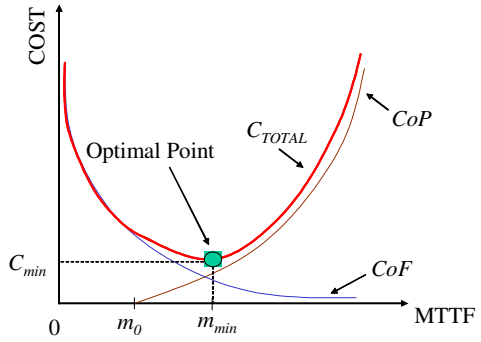


Figure 7-2 The Reliability-Cost Curve

The development of a reliability-cost metric is an extensive, systematic project and once completed, it will greatly expedite the design process to produce chips that meet the reliability requirement at the minimum cost. Unfortunately, it has not been able to draw enough research attention. Methodologies and techniques need to be developed for the determination of: (1) the exact shapes of the *CoP* and *CoF* curves; (2) the values of the all parameters; (3) the exact form of $C_{PP}(m)$. Each is a non-trivial task and requires theoretical contributions as well as empirical guidelines. The determination of $C_{PP}(m)$ is crucial. The evaluation of the reliability of a redundant system and the estimation of the level of redundancy needed to achieve a certain level of reliability is particularly challenging. The research on reliability-cost metric will gradually become a prominent

research field as reliability is becoming the most significant concern in the semiconductor industry.

7.2.3 Integrated Design-for-Robustness Framework

The traditional IC implementation process is composed of several isolated layers of design practice: logical design is the process of mapping from the system-level design handoff to a gate-level representation that is suitable for input to physical design. Circuit design addresses creation of device and interconnect topologies that achieve prescribed electrical and physical properties. Physical design addresses aspects of chip implementation. The output of physical design is the handoff to manufacturing, along with verifications of correctness and constraints. Together, logical, circuit and physical design comprise the implementation layer of semiconductor products.

Silicon complexity makes it virtually impossible to estimate and abstract the effect of physics and embed it on design quality. Logical design and eventually system-level design must become more closely linked with physical design. An integrated design-for-reliability design flow need to be developed to address reliability challenges at all levels of design abstraction.

1) System-level reliability analysis and design: reliability issue manifests itself primarily at the physical design level, where the majority of existing methodologies are focused at. As the design complexity explodes, identification of the vulnerability in the physical design becomes prohibitively expensive. Furthermore, it is almost always too late to start the reliability optimization at such a

low level. It is becoming imperative to consider trade-offs at system level and map results to the lower level. For continued improvements in productivity, system-level design that incorporates the reliability optimization is urgently required.

2) Automatic and intelligent robustness insertion: self-repairing/reconfigurable designs at all levels of abstractions are to be developed to cover the transient failures impossible or too expensive to be identified by the deterministic manufacturing testing. More importantly, these techniques need to be aware of the reliability requirement and design constraints, mandating the development of intelligent decision-making mechanisms.

3) Robustness verification: design verification is the task of proving a given design accurately implements the intended behavior. The introduction of the self-repairing and reconfigurable redundancies requires new methodologies to verify the functionality and performance of the inserted robustness. These verification tasks have to be merged with the existing verification flow that has already become the major consumption of the design resource and budget.

4) On-line detection, diagnosis, repair and reconfiguration: no matter how much improvement and protection has been made during the chip design phase, transient errors will inevitably occur during the chip operation. Consequently, it is important to detect the errors in a timely manner, to evaluate the functional impact caused by such errors, and to make necessary repairs to the system. More importantly, the system has to have the ability to change its reconfiguration depending on the operating environment.

In summary, it is an absolute requirement to develop an integrated design-for-reliability flow (as well as CAD tools) that includes the reliability analysis, design and verification at all levels of design abstraction. This will remain one of the great challenges and major tasks for many distinguished researchers to dedicate their work and energy for a long period of time.

7.3 Summary

This dissertation dedicated itself to the development of efficient methodologies and techniques in the analysis, design and optimization of reliable nanometer circuit systems that are highly resilient to transient error effects. The unified reliability optimization framework proposed in the dissertation will enrich the existing flow and facilitate economical and efficient reliable nanometer chip design.

Bibliography

[1] G. Moore, "Cramming more components onto integrated circuits," Electron, vol. 38, pp. 114-117, 1965

[2] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, 2001.

[3] J. Cong, D. Z. Pan, and P. V. Srinivas, "Improved crosstalk modeling for noise constrained interconnect optimization," Proc. ASP-DAC, pp. 373-378, 2001.

[4] Young S. "Identifying IR drop in high performance nanometer design," Electronic Engineering, vol.74, no.905, pp. 30-33, June 2002.

[5] Shen Lin, Chang N., "Challenges in power-ground integrity," IEEE Intl. Conf. Computer Aided Design, pp. 651-644, 2001.

[6] Hongmei Li, Carballido J, Yu HH, Okhmatovski VI, Rosenbaum E, Cangellaris AC., "Comprehensive frequency-dependent substrate noise analysis using boundary element methods," Intl. Conf. Computer Aided Design, pp. 2-9, 2002.

[7] Peter Hazucha, Christer Svensson, "Cosmic-Ray Soft Error Rate Characterization of a Standard 0.6-μm CMOS Process," IEEE Jnl. Solid-State Circuits, vol. 35, no. 10, Oct. 2000.

[8] IROC Technologies, http://www.iroctech.com

[9] Naval Research Laboratory, https://creme96.nrl.navy.mil/, 1998

[10] Y.Zhao, S.Dey, "Separate Dual Transistor Register – a Circuit Solution for On-line Testing of Transient Errors in UDSM-IC," in Proc. of 9th IEEE Intl. On-Line Testing Symposium, pp.7-11, Kos Island, Greece, June 2003.

[11] H. Cha and J. Patel, "Latch design for transient pulse tolerance," in Proc. ACM International Conf. Computer Design (ICCD), Oct. 1994, pp. 395-388.

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

[13] V. Joshi, R. Rao, D. Blaauw, D. Sylvester, “Logic SER Reduction through Flipflop Redesign,” in Proc. 7th International Symposium on Quality Electronic Design, pp. 611-616, March, 2006.

[14] K. Mohanram, N.A. Touba, “Cost-Effective Approach for Reducing Soft Error Failure Rate in Logic Circuits,” IEEE International Test Conference, pp. 893-901, Sept., 2003.

[15] L. B. Freeman, “Critical charge calculations for a bipolar SRAM array,” IBM J. Res. Dev., Vol. 40, pp. 119-129, Jan, 1996.

[16] X. Bai, R. Chandra, S. Dey and P.V. Srinivas, "Noise-Aware Driver Modeling for Nanometer Technology," IEEE International Symposium on Quality Electronic Design, ISQED'03, March 2003, pp.177~182.

[17] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Cliff Stein, “Introduction to Algorithms”, McGraw-Hill, 1990

[18] http://www.tensilica.com/products/xtensa_overview.htm, Tensilica Inc.

[19] Jeong-Taek Kong, “CAD for Nanometer Silicon Design Challenges and Success,” IEEE Trans. VSLI Systems, pp. 1132-1147, Vol. 12, No. 11, Nov. 2004.

[20] Bryant, R.E., Kwang-Ting Cheng, Kahng, A.B., Keutzer, K., Maly, W., Newton, R., Pileggi, L., Rabaey, J.M., Sangiovanni-Vincentelli, A., “Limitations and Challenges of Computer-Aided Design Technology for CMOS VLSI,” Proceedings of the IEEE, Vol. 89, No. 3, March 2001.

[21] Fabrice Caignet, Sonia Delmas-Bendhia, Etienne Sicard, “The Challenge of Signal Integrity in Deep-Submicrometer CMOS Technology,” Proceedings of the IEEE, Vol. 89, No. 4, April 2001.

[22] R.C. Baumann, “The impact of technology scaling on soft error rate performance and limits to the efficacy of error correction,” Digest of Intl. Electron Devices Meeting, pp. 329-332, 2002.

[23] Kerns, S.E., Shafer, B.D., Rockett, L.R., Jr., Pridmore, J.S., Berndt, D.F., van Vonno, N., Barber, F.E., “The design of radiation-hardened ICs for space: a

compendium of approaches," in Proc. of the IEEE, Vol. 76, No. 11, pp. 1470-1509, Nov. 1988.

[24] K. Joe Hass, "Probabilistic Estimates of Upset Caused by Single Event Transients," 8th NASA Symposium on VLIS Design, 4.3.1-4.3.9, 1999.

[25] M. Omana, G. Papasso, D. Rossi, C. Metra, "A Model for Transient Fault Propagation in Combinatorial Logic," in Proc. of 9th IEEE Intl. On-Line Testing Symposium, pp111-115, Kos Island, Greece, June 2003.

[26] C.Zhao, X. Bai, S.Dey, "A scalable soft spot analysis methodology for compound noise effects in nano-meter circuits," in Proc. of 41st Design Automation Conference, pp. 894-899, San Diego, CA, June, 2004.

[27] Premkishore Shivakumar, Michael Kistler, Stephen W. Keckler, Doug Burger, Lorenzo Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," in Proc. Int. Conf. Dependable Systems and Networks, pp. 389–398, 2002.

[28] Chong Zhao, Yi Zhao, Sujit Dey, "Constraint-Aware Robustness Insertion for Optimal Noise-Tolerance Enhancement in VLSI Circuits," in Proc. of 42nd Design Automation Conference, pp.190-195, Anaheim, CA, June 2005.

[29] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, Mario Szegedy, "Proof Verification and the Hardness of Approximation Problems," J. ACM, vol. 45, no. 3, pp. 501-555, May 1998.

[30] Anghel, L., Nicolaidis, M., "Cost Reduction and Evaluation of a Temporary Faults Detecting Technique," Proc. of Design, Automation and Test in Europe Conference, pp. 591-598, March 2000.

[31] Dhillon, Y.S., Diril, A.U., Chatterjee, A., Singh, A.D., "Sizing CMOS circuits for increased transient error tolerance," in Proc. of Intl. On-line Testing Symposium, pp. 6-10, Madeira Island, Portugal, June 2004.

[32] Quming Zhou, Kartic Mohanram, "Transistor Sizing for Radiation Hardening", Intl. Reliability Physics Symposium, pp. 310-315, 2004.

[33] Yuvraj S. Dhillon, Abdulkadir U. Diril, Abhijit Chatterjee, Cecilia Metra, "Load and Logic Co-Optimization for Design of Soft-Error Resistant Nanometer CMOS Circuits," in Proc. of Intl. On-line Testing Symp., pp. 35-40, 2005.

[34] O. Coudert, R. Hadad, “New algorithms for gate sizing: A comparative study,” DAC’96, pp. 734-739, June 1996.

[35] A. Srivastava, C. Chen, and M. Sarrafzadeh, “Timing driven gate duplication in technology independent phase,” ASP-DAC’01, pp. 577–582, Jan. 2001.

[36] S. Mitra, N. Seifert, M. Zhang, Q. Shi and K.S. Kim, “Robust System Design with Built-In Soft Error Resilience,” IEEE Computer, Vol. 38, No. 2, pp. 43-52, Feb. 2005.

[37] T. Karnik, P. Hazucha, J. Patel, “Characterization of Soft Errors Caused by Single Event Upsets in CMOS Processes,”, IEEE Trans. Dependable and Secure Computing, Vol. 1, No.2, pp. 128-143, April-June 2004.

[38] E. Dupont, M. Nicolaidis, and P. Rohr, “Embedded Robustness IPs for Transient-Error-Free ICs,” IEEE Design & Test of Computers, pp.56-70, May-June, 2002.

[39] C.Zhao, X. Bai, S.Dey, “A Static Noise-Impact-Analysis Methodology for Evaluating Transient Error Effects in Digital VLSI Circuits”, Proc. ITC, pp. 40.2, October, 2005.

[40] T. C. Hu and M. T. Shing, Combinatorial Algorithms, Dover Publications, Inc., pp.111-113, 2002.

[41] C. Zhao, S. Dey, “Improving Transient Error Tolerance of Digital VLSI Circuits Using RObustness COmpiler (ROCO)”, ISQED’06, 2006.

[42] G. C. Messenger, “Collection of charge on junction nodes from ion tracks,” in IEEE Trans. Nuclear Science, Vol. 29, no. 6, pp. 2024-2031,Dec. 1982.

[43] V. Carreno, G. Choi, and R. K. Iyer, “Analog-digital simulation of transient-induced logic errors and upset susceptibility of an advanced control system," NASA Technical Memo 4241, Nov. 1990.

[44] J. J. Doroshenko, S. N. Kraitor, T. V. Kuznetsova, K. K. Kushnereva, E. S. Leonov, “New Methods for Measuring Neutron Spectra with Energy from 0.4eV to 10 MeV by Track and Activation Detectors” Nucl. Technol. 33, pp.296-304 (1977)

[45] L. W. Massengill, M. S. Reza, B. L. Bhuva, T. L. Turflinger, “Single-event upset cross-sectional modeling in combinational CMOS logic circuits,” J. of Radiation Effects, Res., Engrg., vol. 16, no. 1, pp.184-190, 1998

[46] Paul E. Dodd, Lloyd W. Massengill, “Basic Mechanisms and Modeling of Single-Event Upset in Digital Microelectronics,” IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 50, NO. 3, pp. 583-602, JUNE 2003.

[47] D. G. Mavis and P. H. Eaton, “Soft error rate mitigation techniques for modern microcircuits,” in Proc. 40th Annual Reliability Physics Symp. 2002, pp. 216–225.

[48] K. Mohanram, "Closed-form simulation and robustness models for SEU-tolerant design," Proc. VLSI Test Symposium, pp. 327-333, 2005.

[49] N. Vana, W. Schöner, M. Fugger, Y. Akatov, “DOSIMIR – Radiation Measurements Inside the Soviet Space Station MIR – First Results,” Proc. Int. Space Year Conference, Munich 1992, ESA ISY-4, 193 (1992).

[50] C. M. Hsieh, P. C. Murley, and R. R. O’Brien, “Dynamics of charge collection from alpha-particle tracks in integrated circuits,” in *Proc. IEEE Int. Reliability Phys. Symp.*, 1981, pp. 38–42.

[51] J. T.Wallmark and S. M. Marcus, “Minimum size and maximum packing density of non redundant semiconductor devices,” Proc. IRE, vol. 50, pp. 286–298, 1962.

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

[53] T. C. May and M. H. Woods, “Alpha-particle-induced soft errors in dynamic memories,” IEEE Trans. Electron. Devices, vol. 26, pp. 2–9, Feb. 1979.

[54] J. C. Pickel and J. T. Blandford, Jr., “Cosmic ray induced errors in MOS memory cells,” IEEE Trans. Nucl. Sci., vol. 25, pp. 1166–1171, Dec. 1978.

[55] J. L. Andrews, J. E. Schroeder, B. L. Gingerich, W. A. Kolasinski, R. Koga, and S. E. Diehl, “Single event error immune CMOS RAM,” IEEE Trans. Nucl. Sci., vol. 29, pp. 2040–2043, Dec. 1982.

[56] A. E. Giddings, F.W. Hewlett, R. K. Treece, D. K. Nichols, L. S. Smith, and J. A. Zoutendyk, “Single event upset immune integrated circuits for Project Galileo,” IEEE Trans. Nucl. Sci., vol. 32, pp. 4159–4163, Dec. 1985.

[57] S. E. Diehl-Nagle, J. E. Vinson, and E. L. Peterson, “Single event upset rate predictions for complex logic systems,” IEEE Trans. Nucl. Sci., vol. 31, pp. 1132–1138, Dec. 1984.

[58] A. L. Friedman, B. Lawton, K. R. Hotelling, J. C. Pickel, V. H. Strahan, and K. Loree, "Single event upset in combinational and sequential current mode logic," IEEE Trans. Nucl. Sci., vol. 32, pp. 4216–4217, Dec. 1985.

[59] C. Dai, N. Hakim, S. Hareland, J. Maiz, and S. W. Lee, "Alpha-SER modeling and simulation for sub-0.24τ m CMOS technology," in Proc. Symp. VLSI Tech., 1999, p. 81.

[60] N. Seifert, D. Moyer, N. Leland, and R. Hokinson, "Historical trends in alpha-particle induced soft error rates of the Alpha microprocessor," in Proc. Int. Reliability Physics Symp., 2000, pp. 259–265.

[61] S. Buchner, M. Baze, D. Brown, D. McMorrow, and J. Melinger, "Comparison of error rates in combinational and sequential logic," IEEE Trans. Nucl. Sci., vol. 44, pp. 2209–2216, Dec. 1997.

[62] Semiconductor Research Corp. Nat. Technology Roadmap, 1999.

[63] M. P. Baze, S. P. Buchner, and D. McMorrow, "A digital CMOS design technique for SEU hardening," IEEE Trans. Nucl. Sci., vol. 47, pp. 2603–2608, Dec. 2000.

[64] S. Lin and D.J. Costello, "Error Control Coding: Fundamentals and Applications", Prentice Hall, 1983.

[65] Yuval Tamir, Marc Tremblay, "High-Performance Fault-Tolerant VLSI Systems Using Micro Rollback," IEEE Transactions on Computers, Vol. 39, No.4, April 1990, pp. 548-554.

[66] S.-W. Fu, A. M. Mohsen, and T. C. May, "Alpha-particle-induced charge collection measurements and the effectiveness of a novel p-well protection barrier on VLSI memories," IEEE Trans. Electron. Devices, vol. 32, pp. 49–54, Feb. 1985.

[67] O. Musseau, "Single-event effects in SOI technologies and devices," IEEE Trans. Nucl. Sci., vol. 43, pp. 603–613, Feb. 1996.

[68] Yuvraj Singh Dhillon, Abdulkadir Utku Diril, Abhijit Chatterjee, "Soft-Error Tolerance Analysis and Optimization of Nanometer Circuits," Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, 2005.

[69] E. M. Buturla, P. E. Cottrell, B. M. Grossman, and K. A. Salsburg, "Finite-element analysis of semiconductor devices: The FIELDAY Program," IBM J. Res. Develop., vol. 25, no. 4, pp. 218–231, 1981.

[70] J. H. Chern, J. T. Maeda, L. A. Arledge, Jr., and P. Yang, "SIERRA: A 3-D device simulator for reliability modeling," IEEE Trans. Computer-Aided Design, vol. 8, pp. 516–527, May 1989.

[71] L. W. Massengill, "Challenges of modeling SEE's in complex sequential logic," in First NASA/SEMATECH/SRC Symposium on Soft Errors, Radiation Effects, and Reliability, Washington, DC, Oct. 1997.

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

[73] N. Seifert, X. Zhu, D. Moyer, R. Mueller, N. Leland, R. Hokinson, M. Shade, and L. Massengill, "Frequency dependence of soft error rates for sub-micron CMOS technologies," in IEDM Tech. Dig., 2001.

[74] N. Seifert, X. Zhu, and L.W. Massengill, "Impact of scaling on soft-error rates in commercial microprocessors," IEEE Trans. Nucl. Sci., vol. 49, pp. 3100–3106, Dec. 2002.

[75] R. Baumann, "Technology scaling trends and accelerated testing for soft errors in commercial silicon devices," in Proc. of 9th IEEE Intl. On-Line Testing Symposium, pp.7-11, Kos Island, Greece, June 2003.

[76] J.J Liou, K.T. Cheng, S. Kundu, A. Krstic, "Fast Statistical Timing Analysis by Probabilistic Event Propagation", Proc. 38th DAC, pp. 661-666, 2001.

[77] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula. "Computation and refinement of statistical bounds on circuit delay," Proc. DAC, pp.348–353, 2003.

[78] L. Wei, K. Roy, and C. Koh, "Power minimization by simultaneous dual-Vth assignment and gate sizing," Proc. CICC, pp.413-416, 2000.

[79] A. Srivastava, D. Sylvester, & D. Blaauw. "Statistical optimization of leakage power considering process variations using dual-Vth and sizing". Proc. DAC, pp. 773-778, 2004.

[80] Pease, R.L., Johnston, A.H., Azarewicz, J.L., "Radiation Testing of Semiconductor Devices for Space Electronics," PROCEEDINGS OF THE IEEE, VOL. 76, NO. 11, pp. 1510-1526, NOV. 1988.

[81] Predictive Technology Model: http://www.eas.asu.edu/~ptm

[82] W. Zhao, Y. Cao, "New generation of Predictive Technology Model for sub-45nm design exploration," pp. 585-590, ISQED, 2006.

[83] R.E.Walpole, R.H.Myers, "Probability and Statistics for Engineers and Scientists," 5th edition, Ch.7, Macmillan Publish Company, 1993.

[84] Neil H. E. Weste, Kamram Eshraghian, "Principles of CMOS VLSI DESIGN: A System Perspective", 2nd edition, Ch.2, ADDISON-WESLEY PUBLISHING COMPANY, 1993.