UNIVERSITÉ DE MONTRÉAL

MULTILEVEL MODELING, FORMAL ANALYSIS, AND CHARACTERIZATION OF SINGLE EVENT TRANSIENTS PROPAGATION IN DIGITAL SYSTEMS

GHAITH BANY HAMAD
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE ÉLECTRIQUE)
AVRIL 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

MULTILEVEL MODELING, FORMAL ANALYSIS, AND CHARACTERIZATION OF SINGLE EVENT TRANSIENTS PROPAGATION IN DIGITAL SYSTEMS

présentée par : BANY HAMAD Ghaith
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. AUDET Yves, Ph. D., président
M. SAVARIA Yvon, Ph. D., membre et directeur de recherche
M. AIT MOHAMED Otmane, Ph. D., membre et codirecteur de recherche
M. BELTRAME Giovanni, Ph. D., membre
M. LEVEUGLE Regis, Ph. D., membre externe

DEDICATION

To my beloved parents, Ibtissem, Lynne, my brothers, and my sisters.

ACKNOWLEDGEMENTS

It has been an amazing experience to accomplish my Ph.D. thesis at Polytechnique Montreal and Hardware Verification Group (HVG). It certainly would not have happened without the support and guidance of several people to whom I owe a great deal.
First of all, I would like to thank my supervisor, Dr. Yvon Savaria. He was fully supportive, understanding, involved and present during all the phases of my research. I have learned many things from him in regard to research, academia, and life in general.

Secondly, I sincerely thank Prof. Otmane Ait Mohamed, for co-supervising my research work. This thesis would not have been possible without his guidance, his expert advice, his support and encouragements. He introduced me to the topic of this thesis and guided me in the right direction. Next, let me thank all the members of HVG for their help and encouragement. Their friendship brought me a warm environment in the lab. Especially, I thank Ibtissem Seghaier and Marwan Ammar.
Last but not least, I thank my family for their constant moral support and their prayers. They are the people who are closest to me and suffered most for my higher study in abroad. Their support was invaluable in completing this thesis.

RÉSUMÉ

La croissance exponentielle du nombre de transistors par puce a apporté des progrès considérables aux performances et fonctionnalités des dispositifs semi-conducteurs avec une miniaturisation des dimensions physiques ainsi qu'une augmentation de vitesse. De nos jours, les appareils électroniques utilisés dans un large éventail d'applications telles que les systèmes de divertissement personnels, l'industrie automobile, les systèmes électroniques médicaux, et le secteur financier ont changé notre façon de vivre. Cependant, des études récentes ont démontré que le rétrécissement permanent de la taille des transistors qui s'approchent des dimensions nanométriques fait surgir des défis majeurs. La réduction de la fiabilité au sens large (c.-à-d., la capacité à fournir la fonction attendue) est l'un d'entre eux. Lorsqu'un système est conçu avec une technologie avancée, on s'attend à ce qu' il connaît plus de défaillances dans sa durée de vie. De telles défaillances peuvent avoir des conséquences graves allant des pertes financières aux pertes humaines.

Les erreurs douces induites par la radiation, qui sont apparues d'abord comme une source de panne plutôt exotique causant des anomalies dans les satellites, sont devenues l'un des problèmes les plus difficiles qui influencent la fiabilité des systèmes microélectroniques modernes, y compris les dispositifs terrestres. Dans le secteur médical par exemple, les erreurs douces ont été responsables de l'échec et du rappel de plusieurs stimulateurs cardiaques implantables.

En fonction du transistor affecté lors de la fabrication, le passage d'une particule peut induire des perturbations isolées qui se manifestent comme un basculement du contenu d'une cellule de mémoire (c.-à-d., Single Event Upsets (SEU)) ou un changement temporaire de la sortie (sous forme de bruit) dans la logique combinatoire (c.-à-d., Single Event Transients (SETs)). Les SEU ont été largement étudiés au cours des trois dernières décennies, car ils étaient considérés comme la cause principale des erreurs douces. Néanmoins, des études expérimentales ont montré qu'avec plus de miniaturisation technologique, la contribution des SET au taux d'erreurs douces est remarquable et qu'elle peut même dépasser celui des SEU dans les systèmes à haute fréquence [1], [2]. Afin de minimiser l'impact des erreurs douces, l'effet des SET doit être modélisé, prédit et atténué. Toutefois, malgré les progrès considérables accomplis dans la vérification fonctionnelle des circuits numériques, il y a eu très peu de progrès en matière de vérification non-fonctionnelle (par exemple, l'analyse des erreurs douces). Ceci est dû au fait que la modélisation et l'analyse des propriétés non-fonctionnelles des SET pose un grand défi. Cela est lié à la nature aléatoire des défauts et à la difficulté de modéliser la variation de leurs caractéristiques lorsqu'ils se propagent. En outre, plusieurs

détails manquent à haut niveau d'abstraction concernant la structure des circuits et les caractéristiques des SET. Ainsi, plusieurs hypothèses sont généralement envisagées pour modéliser le comportement des SET dans des analyses de haut niveau, ce qui affecte l'exactitude des résultats obtenus. Par conséquent, une détection à faible coût des erreurs douces dues aux SET est très difficile et exige des techniques de vérification plus sophistiquées.

Le présent travail présente une méthodologie multi-niveau (niveau transistor, porte logique, et transfert de registres) permettant de modéliser, d'analyser et d'estimer le taux d'erreurs douces induites par des événements singuliers (SEU). La méthodologie proposée étudie la dépendance des caractéristiques des SET aux formes d'ondes en entrée, aux chemins de propagation, à la polarité des impulsions, aux chemins divergents et à la re-convergence dans le circuit au niveau transistor. De nouveaux résultats sur la propagation des SET à travers différentes combinaisons de logique statiques et de logique TSPC sont présentés. Le comportement observé est ensuite caractérisé pour refléter avec précision la propagation des SET à des niveaux d'abstraction plus élevés.

Au niveau portes logiques, plusieurs techniques de vérification de la propagation des SET sont développées en se basant sur des méthodes formelles et des détails appris à bas niveau, aux niveaux transitors et masques. Dans ce travail, la modélisation formelle et l'analyse de la propagation des SET repose sur les Graphes de Décision Multivoie (MDGs) et les Théories de la Satisfiabilité Modulo (SMT). De nouvelles méthodes qui consiédèrent en même temps l'impact des effets de masquage, de la variation de largeur et des chemins re-convergents ont été développées. Les résultats ainsi obtenus montrent que la méthode de modélisation et d'analyse SMT proposée améliore de façon significative l'efficacité des analyses SET en termes de : 1)*précision* dans la mesure où elle donne des estimations exactes de la sensibilité aux SET appris à partir des modèles de portes logiques extraits des masques. Ces résultats ont permis d'acquérir de nouvelles connaissances sur la vulnérabilité des circuits combinatoires aux SET ; 2)*rapidité* étant donné qu'elle est plus rapide que les techniques contemporaines ; 3)*extensibilité* comme elle peut manipuler des circuits larges et complexes tels qu' un multiplicateur 128 bits. De plus, en se basant sur les résultats de ces analyses au niveau portes logiques, des tables de propagation sont développées pour résumer les comportements de propagation des SET.

Au niveau transfert de registre (RTL), cette thèse présente une méthodologie hiérarchique et multi-niveaux pour estimer le taux des erreurs douces dans les circuits combinatoires. La méthodologie repose sur la méthode de vérification des modèles et les tables de propagation au niveau porte logique. La conception RTL est décomposée en sous-composantes et chaque composante est à son tour annotée avec des détails provenant du niveau transistor.

De nouvelles méthodes d'abstraction et de réduction de la conception sont ensuite proposées en fonction des tables de propagation et de la structure du circuit. En outre, deux modèles différents ont été proposés reposant sur le modèle MDG et le processus de décision markovien (MDP) qui sont ensuite analysés à l'aide des vérificateurs de modèles MDG et PRISM, respectivement. De plus, une nouvelle méthode pour estimer le taux d'erreur douces (SER) est proposée. Afin d'illustrer l'utilité pratique de ces techniques de modélisation et d'analyse, nous avons analysé différents circuits combinatoires. Les approches de modélisation et d'abstraction proposées abaissent considérablement le temps et la mémoire liés à la modélisation et l'analyse de la propagation des SET au niveau transfert de registre. Par exemple, le temps de traitement et la mémoire requise sont réduits de plus de 60%. Pour la première fois, une technique basée sur les diagrammes de décision est développée pour analyser des circuits complexes, comme un multiplicateur 16 bits et des additionneurs 256 bits. D'autre part, les résultats expérimentaux démontrent que les analyses par MDP proposées sont plus rapides que les techniques contemporaines tout en assurant une meilleure précision.

ABSTRACT

The exponential growth in the number of transistors per chip brought tremendous progress in the performance and the functionality of semiconductor devices associated with reduced physical dimensions and higher speed. Electronic devices used in a wide range of applications such as personal entertainment systems, automotive industry, medical electronic systems, and financial sector changed the way we live nowadays. However, recent studies reveal that further downscaling of the transistor size at nano-scale technology leads to major challenges. Reliability (i.e., ability to provide intended functionality) is one of them, where a system designed in nano-scale nodes is expected to experience more failures in its lifetime than if it was designed using larger technology node size. Such failures can lead to serious consequences ranging from financial losses to even loss of human life. Soft errors induced by radiation, which were initially considered as a rather exotic failure mechanism causing anomalies in satellites, have become one of the most challenging issues that impact the reliability of modern microelectronic systems, including devices at terrestrial altitudes. For instance, in the medical industry, soft errors have been responsible of the failure and recall of many implantable cardiac pacemakers.

Depending on the affected transistor in the design, a particle strike can manifest as a bit flip in a state element (i.e., Single Event Upset (SEU)) or temporally change the output of a combinational gate (i.e., Single Event Transients (SETs)). Initially, SEUs have been widely studied over the last three decades as they were considered to be the main source of soft errors. However, recent experiments show that with further technology downscaling, the contribution of SETs to the overall soft error rate is remarkable and in high frequency systems, it might exceed that of SEUs [1], [2]. In order to minimize the impact of soft errors, the impact of SETs needs to be modeled, predicted, and mitigated. However, despite considerable progress towards developing efficient methodologies for the functional verification of digital designs, advances in non-functional verification (e.g., soft error analysis) have been lagging. This is due to the fact that the modeling and analysis of non-functional properties related to SETs is very challenging. This can be related to the random nature of these faults and the difficulty of modeling the variation in its characteristics while propagating. Moreover, many details about the design structure and the SETs characteristics may not be available at high abstraction levels. Thus, in high level analysis, many assumptions about the SETs behavior are usually made, which impacts the accuracy of the generated results. Consequently, the low-cost detection of soft errors due to SETs is very challenging and requires more sophisticated techniques.

In this work, we present a multilevel (transistor, gate, and register transfer levels) framework to model, analyze, and estimate the soft error rate due to single event transients. The proposed framework investigates the dependencies of SET characteristics on the input pattern, propagation paths, pulse polarity, diverging paths, and re-converging paths at the transistor level. New insights on SETs propagation through different combinations of static and TSPC logic are reported. The observed behavior was then characterized to accurately model SET propagation at higher abstraction levels.

At gate level, different SET propagation techniques are developed based on formal methods and low level details extracted from transistor level analysis and the design layout. Multiway Decision Graphs (MDGs) and Satisfiability Modulo Theories (SMTs) are utilized to formally model and analyze SETs propagation. New methods to simultaneously include the impact of masking effects, width variation, and re-converging paths are developed. Reported results show that the proposed SMT modeling and analysis significantly enhances the efficiency of SET analysis in terms of: 1) *accuracy* as it gives accurate estimates of SET sensitivity based on gates timing extracted from layout. These results provide new insights on combinational designs vulnerability to SETs; 2) *speed* as it is faster than contemporary techniques; and 3) *scalability* as it can handle large and complex designs such as 128-bit multipliers. Moreover, based on the results of these gate level analyses, propagation tables are developed to abstract SET propagation behaviors.

At Register Transfer Level (RTL), this thesis introduces a hierarchical multi-level methodology to estimate soft error rates due to SETs in combinational designs based on formal model checking and gate level propagation tables. An RTL design is decomposed into subcomponents and then each component is annotated with its gate level details. New abstraction and design reduction methods are proposed based on the gate level propagation tables and design structure. Furthermore, two different models are proposed based on MDGs and Markov Decision Process (MDP) which are then analyzed using MDG and PRISM model checkers, respectively. Furthermore, a new method to estimate the Soft Error Rate (SER) is proposed. In order to illustrate the practical usefulness of these modeling and analysis techniques, we have analyzed different RTL combinational designs. The proposed modeling and abstraction approaches significantly reduce the time and memory requirements required to model and analyze SET propagation at RTL. For instance, the CPU time and the memory required are reduced by more than 60%. For the first time, a decision graph based technique is developed to analyze complex designs e.g., 16-bit multiplier and 256-bit adders. Moreover, experimental results demonstrate that the proposed MDP based analysis is faster than contemporary techniques, while ensuring better accuracy.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ADD	Algebraic Decision Diagram
ASM	Abstract State Machine
BDD	Binary Decision Diagram
BD	Bundle Data
DAG	Directed Acyclic Graph
DI	Delay Insensitive
FOL	First Order Logic
MDG	Multiway Decision Graph
ROBDD	Reduced Order Binary Decision Diagram
SEE	Single Event Effects
SET	Single Event Transient
SEU	Single Event Upset
SER	Soft Error Rate
SRAM	Static Random-Access Memory
SMT	Satisfiability Modulo Theory
EDA	Electronic Design Automation
PMC	Probabilistic Model Checking
FT	Fault Tree
RTL	Register Transfer Level
TPT	Transistor Propagation Table
CIC	Critical Input Combination
CMOS	Complementary metal–oxide–semiconductor
COI	Cone Of Influence
WOV	Window Of Vulnerability
TSPC	True Single Phase Clock
PIPB	Propagation Induced Pulse Broadening

CHAPTER 1 INTRODUCTION

Aggressive technology downscaling has enabled a remarkable improvement of integrated circuits (ICs) performance, power consumption and cost over the past five decades. This evolution made the integrated circuits indispensable part of our daily lives. However, nanometer technology scale has brought into attention reliability issues that were previously not as much of a concern. This is mainly because it is becoming harder to guarantee the correct functionality of integrated circuits in various environments and design configurations. There are three main sources of unreliability in nanoscale designs namely ; runtime variations (e.g., transistor aging degrade), process variations (e.g., variation in the transistor size), and external radiation-induced soft errors. Process variations are naturally occurring variations in the attributes of transistors (length, widths, oxide thickness) when integrated circuits are fabricated. The unreliability induced by process variations and runtime variations can change cells delay and might eventually impact the timing of a system. Designers working at circuit level usually account for such phenomena by introducing additional timing margins (i.e., relaxing the design timing requirements).

On the other hand, the unexpected behaviors introduced in a system due to external radiation-induced soft errors are much harder to mitigate. This is mainly because such faults have a random nature. In other words, soft errors impact the design behavior for very short periods of time, then they disappear and it is very hard to reproduce and relocate them. For instance, they were responsible for the catastrophic failure and the recall of many safety critical systems, such as implantable cardiac pacemakers [7]. Furthermore, developing low-cost analysis and mitigation techniques for soft errors is very challenging and require novel methods and tools. As a result, soft errors have become one of the most challenging types of uncertainties that impact the reliability of modern electronic systems. These errors are the results of an external hit by a radiation-induced particle when striking the sensitive area of a transistor. For example, an alpha particle in packaging material or neutron particle from cosmic rays. These external radiations, if they have the required strength, can change the output of a transistor for a very short period of time. Depending on the affected transistor, it might flip the value stored in a state element (called as Single Event Upset (SEU)) or temporarily change the output of a combinational gate (known as a Single Event Transients (SETs)). Initially, SEUs have been widely studied over the last three decades as they were considered to be the main source of soft errors. However, with further technology downscaling, SETs have become a major source of soft errors in digital circuits. This is mainly because with each new technology node, a sufficient change in the error generation and propagation behavior is

observed. According to recent studies, smaller device geometries, large number of transistors, and the requirement of a high speed design allow particles with smaller energy to generate SETs and eventually cause a soft error [8], [9]. The sensitivity of integrated circuits to soft errors has grown significantly over the past decade. As a result, there is a growing need to analyze and estimate the impact of soft errors on today's complex digital designs as early as possible in the digital design cycle. The purpose of such analysis is to guide the design and the development of circuits that can tolerate soft errors due to SETs in cost and power effective manner. In other words, in order to achieve cost-efficient reliable integrated circuits, it is crucial to take the reliability into consideration alongside with the conventional area, power, and performance metrics in the design flow.

The rest of this chapter is organized as follows. Section 1.1 identifies and provides evidence of the problem we are addressing in this thesis. In this section, the main limitations of existing modeling and analysis techniques at different abstraction levels are summarized. Moreover, in this section, we introduce both the concept of multilevel and cross layer modeling and analysis which will be used through this thesis. Thereafter, in Section 1.2, the main objectives of this thesis are identified in order to advance this area of research. The main contributions presented in this thesis are summarized in Section 1.3.

1.1 Problem Formulation

Soft errors, induced by radiation, are an increasingly relevant issue impacting the reliability of CMOS Integrated Circuits (ICs) adopted not only in safety-critical applications, such as space and avionic, but also in ground-level applications [10], [11]. The progressive shrinking of device sizes in advanced processing technologies, which have scaled from 0.5 μm to 32 nm in less than two decades, leads to miniaturization and performance improvements. However, the possibility of Single Event Transients (SETs) generation, when an energetic particle hits one of the sensitive sites of a digital circuit, has significantly increased in modern Deep Sub-Micron (DSM) technologies. Therefore, ultra-deep sub-micron technologies are more vulnerable to soft errors [10]. Hence, there is a growing need for fast, accurate, and efficient analysis and estimation techniques of SET propagation in modern-DSM technologies. Contemporary techniques for analyzing SET propagation can broadly be classified based on the level of abstraction at which the analysis is performed : at low or at high abstraction level.

1.1.1 Analysis of SETs at Low Levels

At the transistor level, SET propagation is analyzed using both circuit simulations and experimental analysis. Using simulation based techniques, SET propagation scenarios have been investigated in [3], [12], [13], [14]. Moreover, the impact of technology downscaling on SET characteristics variation while propagating has been analyzed in [15], [16]. Similar analyses of the impact of technology scaling on SET characteristics have been done using both mixed-mode simulations [17] and experimental measurements [18]. SET propagation has been modeled in [19], [20], [21] as a function of the technology, gate design and bias history. Further SET studies showed combined effects, such as temperature [22], [23], and technology [24]. Experiments were also performed to characterize SET sensitivity of new technologies using broad beams of heavy ions [25], [26], [27], [28], [29].

However, contemporary techniques which analyze SET propagation at this level suffer from the following shortcomings :

1. Detailed transistor level analysis to investigate the impact of the following factors on SET characteristics while propagating in digital designs is missing :
   - *Diverging and re-converging paths* : SET propagation through a diverging node can lead to multiple faults at different primary outputs. Moreover, due to re-converging paths, the width of SETs, which propagate through different paths between the fault striking node and the re-converging gate, may combine or overlap when arriving simultaneously at the input of a re-converging gate. Performing such analysis by circuit simulation is possible on small circuits but becomes intractable in large digital systems.
   - *Input pattern and SET polarity* : the variation of the propagation delay for different input patterns of multiple inputs static gates is well known to the community ([30], Chapter 6). Hence, it is possible that SET width in a static gate varies for different input patterns. Moreover, the variation in SET characteristics might be dependent to its polarity.
   - *Timing constraints* : in the True Single-Phase-Clocked (TSPC) logic, the characteristics of SET may vary due to the particle strike time and the timing conditions of the gate where SET propagates through. The impact of all these timing constraints on SET characteristics need to be characterized. Moreover, SET generation and propagation in TSPC logic have not been fully analyzed. This logic was first proposed to deal with the skew problem in the dynamic logic, such as clocked CMOS [31], domino [32], and NORA logic [33]. In TSPC logic, it is possible to achieve high clock frequencies because it simplifies the clock distribution and eliminates

phase overlapping problems [34, 35, 36, 37]. Different possible implementations for the TSPC logic, which use low number of transistor, have been presented in [35]. Furthermore, TSPC logic has been used to develop high operating frequency dividers, and to reduce the power dissipation [38, 39].

2. Several simulation based techniques and tools to estimate the SER using the Monte Carlo method at the transistor level have been proposed, such as SEMM [40]. The accuracy of these techniques is directly related to the number of simulation runs. In addition, Monte Carlo techniques introduce randomness in the simulations and fail to cover all possible scenarios. Mixed-mode simulations [17] reduce the simulation time required by a static approach using a mixed-mode simulator, where the current injection part is simulated at the circuit level, while the rest of the circuit can be modeled at the timing level. Moreover, in order to reduce the overhead of detailed circuit simulations, a combination of analytic and simulation based methods to estimate the SER has been implemented in different tools, such as SERA [41] and SEUPER_FAST [42]. However, simulation based techniques are very time consuming when dealing with large systems and require a large amount of resources.

1.1.2 Analysis of SETs at Higher Abstraction Levels

As designs in modern DSM are more vulnerable to soft errors, it has become imperative to address SET propagation issues at an earlier stage in the design flow. Therefore, researchers came up with different SET propagation models operating at gate and higher abstraction levels. Some of these techniques are also based on simulations with fault injection based on random vector generation [43], [44]. Moreover, several Monte Carlo simulation based techniques have been proposed to analyze the impacts of the masking effects (logical, electrical, and timing masking) on SET propagation at gate level, such as [45], [46], [47].

Other research groups have addressed this issue using formal verification methods such as ; Binary Decision Diagram (BDD)-based techniques [48], a combination of Reduced-Order Binary Decision Diagrams (ROBDDs) and Algebraic Decision Diagrams (ADDs) [49], and Boolean satisfiability solvers (SAT-solvers) [6]. However, contemporary techniques at gate and higher levels suffer from the following shortcomings :

1. Several simulation based techniques have been proposed to estimate the SER in combinational logic at gate and higher abstraction level. At gate level, tools, such as FAST [45], ASERTA [46], and ASSA [47], have been proposed to analyze the impact of all masking effects on SET characteristics. These tools use a zero-delay fault simulator to analyze logical masking. Moreover, these tools have different implementations of the

delay degradation model due to electrical masking, which was proposed in [3]. The model in [3] uses look up tables and an equivalent inverters chain based approach. However, all the aforementioned tools are not able to correctly predict the behavior of asynchronous circuits. Such techniques can only handle combinational and synchronous sequential circuits. Additionally, simulation based approaches have serious shortcomings as they can be very time consuming for large designs with many primary inputs and sequential states. Furthermore, these techniques have their drawbacks in terms of accuracy. This is mainly because the accuracy of fault simulation decreases with the ratio of the simulated sample size over the total vector space size.

Different numerical techniques are proposed, such as [50], [51], [52] [41]. Each of these techniques try to estimate the impact of masking effects on the SER. Electrical masking is presented in [50], temporal masking is analyzed in [41], and a model combining all masking effects is presented in [51]. However, these techniques are not scalable and their models do not include the impact of SET broadening and SET re-convergence.

2. State-of-the-art techniques at gate level (such as [48], [49]) analyze the susceptibility of digital circuits to soft errors by modeling only the masking effects that can prevent SET in digital designs from propagating [10]. These techniques omit the possibility that a SET could broaden while propagating. SET broadening was first observed in [53] and it has recently gained more attention and was addressed in [12, 54, 20, 55]. Furthermore, at Register Transfer Level (RTL), many details related to the design structure and SET characteristics are not available. Therefore, in [6, 56], SETs are modeled at RTL as bit flips. Thus, the SER estimated at this abstraction level is generally inaccurate.

3. Techniques based on contemporary formal verification are resource hungry and suffer from a *state explosion* problem. This is mainly due to the intrinsic characteristics of their SET modeling technique. Indeed, in these techniques, each input vector is mapped to a unique state. Therefore, the corresponding Markov model has 2^M states (M primary inputs). Additionally, with these techniques, the formal model of the design size is doubled due to the requirement of two design versions, mainly a golden and a faulty version. For each injection scenario, in order to determine if a SET is propagating, the outputs of both the golden and the faulty version are compared. With such modeling technique, any formal tool rapidly runs out of memory, even when trying to analyze moderate size designs e.g., a 14-bit adder [56].

1.1.3 Multi-level Modeling and Analysis of SET Propagation

Designers and researchers found that the best way to build complex hardware designs is to start from very high level descriptions and synthesize them all the way down to layout as shown in Fig. 1.1. This methodology is only applicable to synthesizable designs. With synthesis, the code representing the design at one abstraction level can be translated into lower level implementations using pre-characterized rules and libraries. In other words, a design is synthesizable if the synthesis tool has the synthesis library (i.e., from which the low level implementation can be generated) for each part of the design. Therefore, the main concept in the design methodology is to utilize the lower level details from pre-characterized data to build large designs. In each synthesis phase between two abstraction levels, more details about the design structure are added.

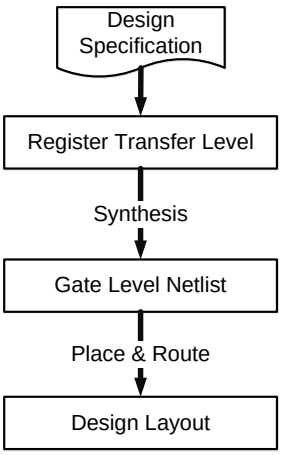


Figure 1.1 The Digital Design Flow.

Unfortunately, when it comes to non-functional design verification, it is totally different and there is no unified approach. Design verification at one abstraction level relies on the information provided at this level. As shown in Table 8.1, low level analysis is very detailed, however, it is resource consuming and not applicable for large designs. On the other hand, higher level analyses are more time and resource efficient. However, their results have limited accuracy and do not provide much useful information to the designers about the design behavior in presence of different kinds of uncertainties. Therefore, there is a growing need to reduce the gap between the fault analyses at these different abstraction levels.

In order to visualize this issue, consider Fig. 1.2. At each abstraction level, there is a number of faults (represented in Fig. 1.2 as dots) which can lead to design failures. Each of these faults are triggered by a number of faults from lower abstraction levels. Thus, if the details of faults from lower abstraction levels are available, then it is possible to accurately model and analyze

them at higher abstraction levels. Verification engineers define possible fault candidates based on the amount of details available about the design structure at one abstraction level. The number of faults in the design increases as we are moving toward lower abstraction levels. However, faults at one abstraction level do not have the same weight, i.e., possibility of occurrence.

To overcome this issue, the usability of the results of the fault analysis at each abstraction level has to be improved. New methods to abstract the design details that directly affect fault propagation are required. However, we have to ensure that the additional overheads in terms of learning and verification are reasonable.

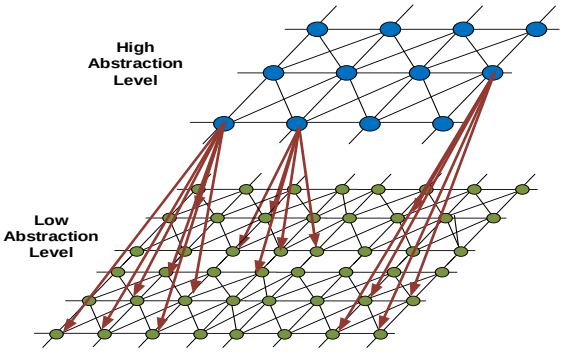


Figure 1.2 The Concept of Modeling SET Propagation at High Levels Based on the Observed Behavior at Low Level

Table 1.1 Comparison Between High Level and Low Level Analysis

	Analysis at Gate and Higher Level	**Analysis at Transistor and lower Level**
Fault Modeling Accuracy	Very Abstract does not reflect actual fault behavior	Very Detailed Reflects actual behavior
Complexity	Less Complex	Very Complex
Result's Accuracy	Under/Over estimation	More accurate
Result's Usability	Not used	Not used
Memory	Less Memory	High Memory
CPU Verification Time	Less Time	Very Large Time

1.1.4 Cross layer Modeling and Analysis of SET Propagation

In order to perform an accurate SET analysis, details from the circuit level are required. At circuit level, parameters extraction and detailed simulations can provide a certain level of accuracy for phenomena such as electrical masking and SET width variation. However, this analysis is very computationally intensive and would be intractable at the chip level and is only tractable at the cell level. In other words, this type of analysis can be conducted on hundreds of transistors at most. Similar to functional verification, existing SET analysis techniques abstract the design details to perform such analysis at higher abstraction levels (i.e., gate level or higher). However, abstraction normally comes at the cost of reduced accuracy, since many details about the design layout and the SET's characteristics are abstracted. Therefore, a new technique which satisfies the following requirements is needed : 1) to be fast ; 2) to be more scalable than circuit level analysis ; and 3) to be able to model SET propagation based on underlying technology details to maintain a certain level of accuracy.

As explained before, single event transients start at the device level (at some sensitive area of the transistor) and can propagate to impair the behavior of a whole system. Therefore, one possible idea is to model each component of a design based on its relation with the injected SET into the following classes : 1) the component that was affected by radiation and an SET generated internally in this component ; 2) the components that propagate the generated SET from where it is injected to one or more primary output ; and 3) the components which do not generate nor propagate SETs i.e., *Error-Free*.

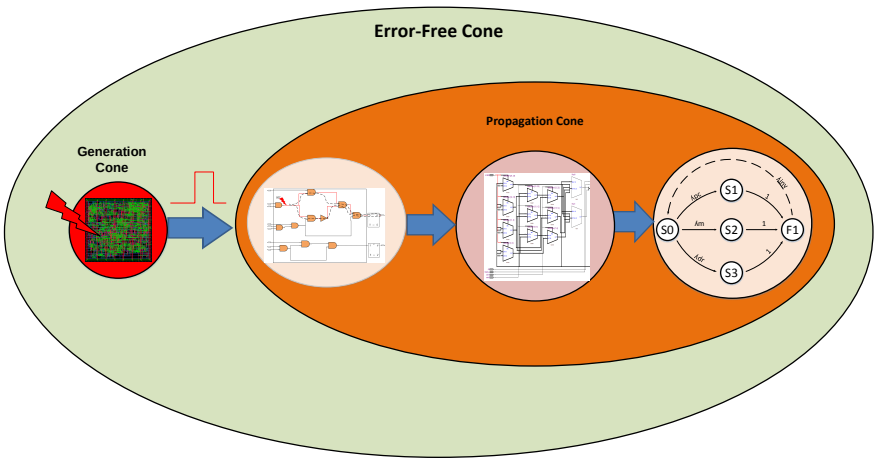


Figure 1.3 The Concept of Cross-Layer Modeling and Analysis of SET Prorogation

Based on this classification, for each injection scenario, a design can be divided into three cones : a) *Generation cone* ; b) *Propagation Cone* ; and c) *Error-Free Cone* as shown in Fig.

1.3. As depicted in this figure, the *Generation cone* is expected to be small in size, but contains more details. On the other hand, the propagation cone can be very large in size, but it contains reduced amount of details (i.e., its accurate modeling is less expensive). In the propagation cone, we can have different sub cones where each has certain amount of details based on SET propagation behavior in this sub-cone (see Fig. 1.3). The *Error-Free Cone* includes all the Error free components which do not directly propagate the SET but they might impact its propagation (by enabling or blocking some paths or by loading other paths). Later in this thesis, this idea of modeling and analyzing the SET propagation using variable levels of details for different regions in a cross-level approach is investigated in details.

1.2 Thesis Objectives

Based on the previous discussions of the limitations in the existing SET modeling methods, it can be noticed that the following important questions are not appropriately addressed in the literature so far :

- **R1 :** *What is the impact of the propagation paths, polarity, and fanout re-convergence on SET characteristics ?*
- **R2 :** *How to abstract the SET propagation behavior observed at transistor level at higher abstraction levels ?*

Based on our discussions of existing SET propagation modeling at higher abstraction levels (i.e., gate level and higher) it can be noticed that the following important questions are not appropriately addressed in the literature so far :

- **R3 :** *How to improve the usability of the results generated from lower abstraction levels such as transistor level analysis ?*
- **R4 :** *How to efficiently utilize formal verification methods to model and analyze SET propagation at high abstraction level ?*
- **R5 :** *How to measure the vulnerability of complex designs at high abstraction levels without losing the accuracy provided from low level analysis ?*
- **R6 :** *Is it possible to improve scalability while preserving accuracy ?*

Our objective in this thesis is to investigate possible solutions to the questions introduced in this section at transistor, gate, and register transfer levels.

1.3 Thesis Contributions

In this thesis, I developed a multilevel framework which accurately models and analyze soft errors in digital circuit due to single event transients from a technology response model derived

at the transistor level all the way to the register transfer level. At each abstraction level, suitable modeling of phenomena related to SET propagation are proposed. Formal verification methods are utilized at the higher abstraction level to build an accurate exhaustive modeling. Moreover, new quantitative measures of the contribution of SETs at each node in the design to the design failure are proposed. Furthermore, in the proposed analysis at each abstraction level, new means of estimating the soft error rate are proposed. The rest of this section summarizes the main contributions developed in this thesis.

1.3.1 Transistor Level Analysis of SET Propagation

I conducted different analyses to fully understand SET propagation behavior at the transistor level and to address the research question reported in Section 1.2 (i.e., R1 and R2). I have analyzed SET characteristics variation while propagating through both static and TSPC logic at the transistor level. This work is distinct in the following ways :

1. **Investigate the impact of propagation path on SET** (*related to R1*) : The variations in SET characteristics based on the characteristics of its propagation paths are investigated. SET width broadening or attenuation based on the propagation paths are characterized. Worst and best propagation paths are identified for the analyzed designs. Moreover, the required timing and characteristic conditions for the generation and the propagation of SETs through TSPC logic are abstracted. The impact of the input patterns and SET polarity (negative ($1 \rightarrow 0 \rightarrow 1$) or positive ($0 \rightarrow 1 \rightarrow 0$)) on SET characteristics variation while propagating is fully explored. The variations in SET width for each possible input pattern and polarity are characterized for both static and TSPC logic.
2. **Investigate the impact of re-converging and diverging paths on SET propagation** (*related to R1*) : The possibilities of SET width attenuation or broadening due to re-converging paths are investigated. Pulses may re-converge and overlap at a gate in a circuit if multiple paths exist between the particle striking node (affected node) and the re-converging gate. The SET propagation scenario which can induce Byzantine faults is identified. Byzantine faults are defined as faults presenting different symptoms (or logic interpretation) to different observers.

The transistor level analysis main results and observations in relation with these issues led to the following publications :

— **C1** : ***G. Bany Hamad****, S. R. Hasan, O. Ait Mohamed, Y. Savaria, (2013)"Investigating the Impact of Input Patterns, Propagation Paths and Re-convergent Paths on*

The Propagation Induced Broadening." 14th IEEE Conference on Radiation Effects on Components and Systems (RADECS' 2013).

— **J1** : ***G. Bany Hamad,*** *S. Rafay Hasan, O. Ait Mohamed, and Y. Savaria, (2014). "New insights into the single event transient propagation through static and TSPC logic", IEEE Transactions on Nuclear Science, vol.61, no.4, pp.1618-1627.* In this thesis, this journal paper is reproduced in Chapter 4.

1.3.2 Modeling and Analysis of SET Propagation at Gate Level

At gate level, I proposed new solutions to abstract, model, and analyze SET propagation using formal verification methods. Following, the main ideas published proposed in this area are listed :

1. **Abstraction of SET propagation behavior from transistor to gate level** (*related R2 and R3*) : In order to bridge the gap between transistor and gate levels modeling, I proposed a new logic abstraction of the SET width variation observed at the circuit level. The impact of the applied input pattern and the gate fan-out on the SET width is abstracted using the Load and Input Combination Factor (LICF). Moreover, I proposed new characterization libraries modeling SET propagation that provide a comprehensive abstraction of several propagation behaviors previously ignored at gate level. The proposed analysis and abstraction advances the state-of-the-art in modeling SET at abstraction levels higher than the circuit level, enabling more accurate estimation of the soft error sensitivity and improved reliability of digital systems. The proposed abstraction model led to the following publication :
   — *G. B. Hamad, S. R. Hasan, O. Ait Mohamed, Y. Savaria, "Abstracting single event transient propagation characteristics to support gate level modeling", IEEE International Symposium on Circuits and Systems (ISCAS' 2014).*
2. **Multiway decision graph based modeling and analysis of SETs** (*related to R3 and R4*) : I proposed new means of modeling SET propagation at gate level by utilizing the Multiway Decision Graphs (MDGs) [57] and transistor level characterization libraries. MDGs are chosen over other types of decision graphs because they allow defining SET width variation and other known masking effects (such as logical masking) in a single decision diagram. This analysis identifies the set of conditions, related to SET propagation and design structure, that may lead to soft errors. These conditions are abstracted as gate level characterization libraries for each design. These libraries can be used to perform SET propagation analysis at higher abstraction levels. Based on the results of this analysis, a new estimation of the design's SER is genera-

ted. This combination leads to more accurate analysis and requires less memory than contemporary techniques. The proposed MDG based modeling led to the following publications :

— **G. Bany Hamad,** S. Rafay Hasan, O. Ait Mohamed, and Y. Savaria, (August, 2014). "*Modeling, analyzing, and abstracting single event transient propagation at gate Level*", In IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), pp.515-518.

— **G. Bany Hamad,** S. Rafay Hasan, O. Ait Mohamed, and Y. Savaria, (2015). "*Characterizing, modeling, and analyzing soft error propagation in asynchronous and synchronous digital circuits*", Microelectronics Reliability, Volume 55, Issue 1, Pages 238-250.

3. **Satisfiability modulo theories based SETs modeling and analysis** (*related to R3, R4, R5, and R6*) :

   I introduced a novel methodology to evaluate the vulnerability of digital designs to SETs at gate level. This methodology provides a new technique for modeling SETs propagation by introducing awareness about the underlying observed behavior at transistor level. SET propagation is modeled as a satisfiability problem leveraging the efficiency of Satisfiability Modulo Theories (SMTs). The theories of linear integer arithmetic and difference logic are utilized to efficiently model SET width and timing constraints. Moreover, in this model, concepts of static timing analysis are adapted to compute the required timing and width for an SET to be latched. In the analysis phase, the proposed methodology computes the timing and width requirements for every vulnerable node in the design. This is done by investigating the width variation and delays along the different propagation paths. I implemented the proposed analysis on different SMT solvers in order to compare the performance of each and decide on an optimal modeling technique and solver. The solvers I used are *Z3* [58], *Yices* [59], *Mathsat* [60], and *CVC4* [61]. All these results are characterized into gate level propagation tables which have the following benefits : 1) They are used to measure the observability of each SET at each node and the vulnerability of the design i.e., SER. 2) They can also be used to further improve the efficiency of the analysis of soft error mitigation techniques.

   The proposed SMT modeling and analysis based on SMTs led to the following publication :

   — **G. Bany Hamad,** G. Kazma, O. Ait Mohamed, and Y. Savaria, (2016). "*Comprehensive Non-Functional Analysis of Combinational Circuits Vulnerability to Single Event Transients*", Forum on specification & Design Languages (FDL).

4. **Layout based gate level estimation of SER due to SETs** (*related to R3, R4, R5, and R6*) :

   I introduce a novel methodology to estimate the vulnerability of combinational designs to soft errors at gate level. This methodology starts with the synthesis of an RTL design into its gate level representation and then the layout of the design is extracted. Next, gates parasitics are extracted and gates timing details are characterized from the layout. These parameters are then employed to model and analyze SET propagation. A new model for SETs propagation is proposed, which captures the variations in the SET characteristics while propagating, such as the SET width attenuation and broadening. Moreover, this model includes the impact of all masking effects (logical, electrical, and temporal) and re-converging paths on SET propagation. Furthermore, a new formalism modeling SET propagation into a Satisfiability problem utilizing SMTs to utilize the design timing extracted from the layout is proposed. A new estimate of the design SER is computed.

   The proposed post-layout modeling and analysis based on SMTs led to the following publications :

   — **G. Bany Hamad,** G. Kazma, O. Ait Mohamed, and Y. Savaria, (2016). "*Efficient and Accurate Analysis of Single Event Transients Propagation Using SMT-Based Techniques*", IEEE International Conference on Computer-Aided Design (ICCAD).

1.3.3 Modeling and Analysis of SET Propagation At Register Transfer Level

I proposed different methods to model and analyze SET propagation at RTL using formal verification methods. This work is distinct in the following ways :

1. **MDG based abstraction, modeling, and analysis of SET propagtion at RTL** (*related R3, R4, R5, and R6*) : Abstraction is one of the most relevant techniques for addressing the *state explosion* problem [62]. I proposed a new abstraction approach in which the components in the RTL design are modeled based on their mode of operations. An RTL component can have three modes of operation ; *Injection*, *Propagation*, or *Error-Free*. For each injection scenario, SET propagation at RTL is modeled based on the sub-components mode of operation and their gate level characterization libraries developed beforehand. Similar to the gate level analysis, I utilized MDGs [57] to analyze SET propagation at RTL. The invariant checking tool from the MDG formal verification tool set [57] is adapted to perform this analysis. The results, which are SET propagation conditions for all injection scenarios, are reported as RTL characterization libraries.

The results of this analysis have been reported in the following publications :

— **G. Bany Hamad,** O. Ait Mohamed, and Y. Savaria, (May, 2016). "*Towards Formal Abstraction, Modeling, and Analysis of Single Event Transients at RTL*", IEEE International Symposium on Circuits and Systems (ISCAS).

2. **Probabilistic modeling and analysis of SET propagation at RTL** (*related to R3, R4, R5, and R6*) : I proposed an efficient probabilistic reduction and modeling techniques to analyze SET propagation. I proposed two efficient reduction methods namely the Cone Of Influence (COI) and the component mode of operation methods. At RTL, SET propagation is modeled based on the proposed fault space mapping technique. The propagation of high level faults for each sub-component is modeled as *Probabilistic Automata (PA)* based on the propagation probabilities of low level faults reported in the pre-characterized sub-component propagation table. The *PAs* of all sub-components are modeled as *Markov Decision Processes (MDPs)*. Thereafter, SET propagation is quantitatively analyzed using the proposed formal probabilistic verification technique that utilize the power of PMC. The results of this analysis are the SET propagation probabilities for all vulnerable nodes. Finally, theses probabilities are utilized to estimate SERs.

   The results of this analysis have been reported in the following publications :

   — **G. Bany Hamad,** O. Ait Mohamed, and Y. Savaria, (2014). "*Probabilistic model checking of single event transient propagation at RTL level*", IEEE International Conference on Electronics Circuits and Systems (ICECS), Marseille, France, pp. 451-454.

   — **G. Bany Hamad,** O. Ait Mohamed, and Y. Savaria, (July, 2015). "*Efficient Multilevel Formal Modeling, Analysis, and Estimation of Design Vulnerability to Soft Error*", IEEE International On-Line Testing Symposium (IOLTS), Athena Pallas Village, Greece, pp. 1-6.

   — **G. Bany Hamad,** O. Ait Mohamed, and Y. Savaria, (2016). "*Comprehensive Multilevel Probabilistic Analysis of Single Event Transients Propagation Induced Soft Errors*" Submitted for publication.

1.4 Thesis Organization

The organization of this thesis is as follows.

In Chapter 3, the main sources of single event transients in digital circuits are discussed. Then, the different formal verification methods which we utilized in this thesis to model and analyze SET propagation at high abstraction levels are introduced.

In Chapter 2, the details of the most relevant SETs modeling and analysis techniques at different abstraction levels from register transfer to post silicon levels are discussed.

Chapter 4 explains in detail our investigation on the impact of the propagation paths, input patterns, and polarity on SET characteristics. This analysis is performed for both static and TSPC CMOS logic. Based on the transistor level netlist, the worst and best propagation paths (WPP and BPP) were identified for the analyzed designs. The impacts of Propagation Induced Pulse Broadening (PIPB) phenomena and SET propagation induced Byzantine faults are characterized.

Chapter 5 first introduces our proposed abstraction of the variation in the SET characteristics while propagating due to electrical masking and width broadening. The impact of the applied input pattern and the gate fan-out on the SET width is abstracted using the Load and Input Combination Factor (LICF). Then, our proposed modeling of SET propagation at gate level using multiway decision graphs is explained in details. This chapter also presents our proposed gate level analysis of SET propagation performed with the MDG model checker. Finally, the characterization of the results of this gate level analysis are introduced.

Chapter 6 first introduces the proposed design and technology node timing characterization. Then, we explain the proposed formulation of SET propagation at gate level (which includes all masking effects and width variation) as an SMT problem. This chapter also presents the proposed analysis of SET propagation using SMT solvers under specific assertions. Finally, the chapter proposes an improved method for estimating the SER based on the generated results (e.g., the set of input vectors that must be present at the primary inputs so that SETs are not logically masked).

Chapter 7 introduces our proposed hierarchical formal method that allows modeling and analyzing SET propagation at register transfer level. The chapter first introduces, the proposed RTL abstraction based on the COI and components mode of operations. Then, it introduces the proposed modeling of the underlying behavior of SET propagation using Multiway Decision Graphs (MDGs). Next, the proposed SET propagation analysis at RTL based on invariant checking tool from the MDG tool set is explained. For each SET injection scenario this analysis returns a CIC that can propagate this SET to the output.

Chapter 8 presents our hierarchical probabilistic framework to quantitatively estimate the effects of SETs at RTL. First, we explain the proposed RTL reduction for each injection scenario and the propagation tables generated from lower abstraction level models. Then, SET propagation through the reduced design is modeled as Markov decision process based on probabilistic automatas of all the RTL sub-components. Next, a method is proposed for probabilistic analysis based on the PRISM model checker to analyze the probability of SET

propagation for all vulnerable nodes. Finally, a new estimation of the Soft Error Rate (SER) based on the results of this analysis is proposed.

Chapter 9 provides a general discussion about the present work, which has been detailed in Chapter 4, Chapter 5, Chapter 6, Chapter 7, and Chapter 8. Chapter 10 summarizes this thesis and proposes some directions for future work.

CHAPTER 2 CRITICAL LITERATURE REVIEW

In this chapter, we briefly review the status of existing related SET propagation analysis techniques ; post-silicon radiation testing and SET analysis at different abstraction levels (transistor to register transfer levels).

2.1 Post-Silicon Validation of SETs Using Radiation Ground Testing

The traditional and most direct approach to evaluate the SEU vulnerability of a system is through a process called *dynamic radiation ground testing* [63], [64]. This method consists in exposing the target system to a radiation flux (to reproduce the desired radiation environment) and counting the number of errors observed. The intensity of the artificial radiation flux is controlled based on the environment that the device has to work in. The outcome is computed in the form of a parameter known as the *dynamic cross section* (σ), which estimates the number of errors that occur in an area of the processor over time. There are several studies investigating the relative contributions of sequential and combinational SER based on the results of radiation testing which use simple test structures such as SOI and bulk inverter chains [1, 2, 65, 12]. The results of these experiments confirm that the contribution of combinational logic is increasing with every new technology node and also has a linear relation with the circuit frequency. One of the main issues with these experiments is the simplicity of the test structures, i.e., hard to apply to complex designs. Another problem with the results of these experiments (e.g., *dynamic cross section* metric) is that any change in the application, design, technology node, device manufacturer, flux density (i.e., the desired environments), and the particle strike type requires a new dynamic test. Thus, resulting in a very expensive and time-consuming method. Furthermore, these analyses have very coarse controllability, problematic reproducibility, very limited observability, and is very difficult to debug. For instance, radiation testing techniques cannot accurately determine the contribution of each component in the design to the overall failure rate of the design. In fact, during radiation testing experiments on complex circuits, it is difficult to differentiate between the contribution of the sequential and the combinational parts of the designs. Although these experiments provide accurate insights into the relative SER of the designs, details about the vulnerability of different components are missing. These details are critical when designing mitigation techniques to harden the design. Therefore, these experiments can be used to verify the existence of some physical phenomena (such as SETs width variation) but they are not suitable for SER estimation during design stage.

2.2 Analysis of SETs Propagation at Transistor Level

A large amount of research has been performed for measuring and simulating SET propagation in deep submicron bulk technologies. The reader is referred to the recent comprehensive reviews presented in [11], [66] for technical details or further discussion of the literature. In this section, we discuss the transistor level analysis that we have utilized to fully understand SET propagation behavior in CMOS logic. The results of these analyses are used to build a more accurate model at higher abstraction levels. The analysis performed in [53] demonstrates that SET width variation while propagating can be attributed to the speed difference between the rising and the falling edge along a given path.

In [13], the distribution of SET width was measured in long SOI chains irradiated with broad beam heavy ions. In this work, the propagation-induced pulse broadening (PIPB) effect was first experimentally modeled and measured in SOI inverter chains.

Subsequently, many researchers have analyzed the broadening phenomena of SET width while propagating and its dependencies on active loading (fan-out) and transistor size such as [54, 20, 55]. In [20], SET width variation while propagating in logic chains is investigated. It is shown that significant broadening or attenuation of the propagated SET width is observed. Moreover, the dependence of the SET width on the struck node capacitance is investigated. Results demonstrated that increasing node capacitance broadens the SET width while propagating.

Other parameters, such as the supply voltage, have a significant impact on PIPB. For example, an inverse relationship has been observed between the PIPB effect and the supply voltage [12, 67, 68]. Moreover, a direct relationship has been observed between the PIPB effect and transistor sizes [54, 20, 55].

As explained in Section 1.1, existing state-of-the-art techniques are unable to analyze the effects of the propagation paths, the re-converging paths, and the input pattern on the SET characteristics.

2.3 Formal Analysis of SETs Propagation at Gate Level

Several formal methods based techniques and tools have been recently constructed to analyze and estimate SET propagation at the gate level. Many of these techniques are based on BDDs such as [69, 48, 70, 49, 71].

In [48], fast analysis of soft-error (FASER) which is an SER estimation tool based on binary decision diagrams is introduced. The proposed BDD model enumerates all possible input

vectors by creating a BDD for each gate in a circuit. Static BDDs (which only include the Boolean functionality) are created for gates outside the SET propagation cone. On the other hand, BDDs which include details about the propagating SET width and amplitude for gates in the propagation cone. Thereafter, these BDDs are combined in topological order to model both the logical and electrical masking of the injected SET. FASER adapts the delay degradation model for SET width variation while propagating. As explained before, this model only includes the case where the width of an SET is attenuated while propagating through logic gates. FASER's BDD representations can consume a lot of memory when implemented on practical circuits. Therefore, FASER partitions large designs, to lessen the amount of memory, into smaller sub-designs.

In [49, 71], the authors used ROBDDs and the algebraic decision diagrams (ADDs) in combination to model soft errors in combinational and sequential circuits to simultaneously analyze the effects of logical, electrical. The SET width attenuation is modeled in the ADD and the logical masking (sensitization paths) are modeled in ROBDD. However, the use of two decision diagrams makes this technique more complex and it consumes a considerable amount of memory.

However, despite the prevalence of circuit partitioning techniques, all BDD-based techniques are inherently limited due to the memory blowup problems associated with them (state space explosion problem). Furthermore, these techniques [69, 48, 70, 49, 71] oversimplify the electrical masking impact on the SET while propagating by simulating inverter chains of the same lengths as the paths.

2.4 Formal Analysis of SETs Propagation at RTL

A new technique was proposed in [6]. It leverages the concepts of Boolean Satisfiability and uses the so-called SAT (satisfiability) solvers. In spite of the use of very efficient SAT solvers, this method is time consuming and resource hungry, partly because of the requirement of unrolling copies of its combinational circuit when analyzing sequential designs.

Soft error analysis at early design phase is essential for applying appropriate mitigation techniques to meet the reliability requirements. In [72], a new approach to investigate the soft error propagation properties at behavioral RTL, especially for the control paths of the design. At RTL, low-level circuit details are generally not available, therefore this technique models soft error propagation as single bit-flips, i.e. Single Event Upsets (SEUs). The probabilistic behavior of RTL circuit is modeled as finite Discrete Time Markov Chains (DTMCs). Thereafter, probabilistic model checking (PRISM) is adopted to analyze soft error propagation

in the RTL DTMC model. However, the well-known state explosion problem [62] limits its applicability and hence scalability improvement techniques are essential.

Recently, in [73], a new probability model of all three masking effects of SET propagation is proposed. This methodology involves analyzing standard digital designs (adders, muxes, etc.), by injecting SETs at all vulnerable nodes and then computing their intrinsic SET rate at the gate level. In this technique, the possibility of SET width broadening and its impact on SET propagation probability is not considered.

Several methodologies have been proposed analyze SET propagation at Register Transfer Level (RTL) using fault simulation [74] and analytical techniques [75]. Other researchers have addressed this issue using formal verification methods such as Boolean Satisfiability solvers [6] and Probabilistic Model Checking (PMC) [56], [76, 4]. All these techniques suffer from the following shortcomings :

1. Contemporary formal verification techniques are resource hungry and limited due to the *state explosion* problem. This is mainly due to intrinsic characteristics of their modeling technique. Indeed, in these techniques, SET propagation is modeled using concrete Boolean diagrams e.g., Binary Decision Diagrams (BDDs). With such techniques, a model checker rapidly runs out of memory, even when modeling moderate size designs e.g., 14-bit adder [56] or 15-bit multiplier [77].
2. Simulation based techniques (such as [74], [75]) have serious shortcomings as they are very time consuming for large designs with many primary inputs. Furthermore, these techniques have their drawbacks in terms of accuracy. This is mainly because their accuracy is determined by the ratio of the simulated sample size over the total vector space size.
3. At RTL, many details about design structure and SET characteristics are not available. Therefore, contemporary techniques make assumptions about SET propagation. For instance, in [75, 56] and [76], SETs are modeled as bit flips. Such assumptions reduce the accuracy of the estimated Soft Error Rate (SER).

CHAPTER 3 BACKGROUND INFORMATION

In this chapter, a brief background is provided about soft errors, single event transients, and formal methods that are utilized in this thesis.

3.1 Basics of Soft Errors due to Single Events Transients

3.1.1 Origins of Single Event Transients

As the name suggest, soft errors do not permanently damage the circuit as hard errors. These errors can change the behavior of the circuit temporarily. In this thesis, we are modeling and analyzing soft errors due to single event transients. These transient faults are the result of a strike of some radiation-induced particle at a drain of a transistor device. Such strike can generate a track of electron hole pairs in the bulk of the device. These charges can be captured to induce a current pulse which may flip the output of a combinational gate for a short period of time (duration of the induced SET). *Alpha particles* (generated from the radioactive impurities in the chip manufacturing and packaging) and *neutron particles* (secondary particles from cosmic rays) are two main sources of SETs. The neutron flux is dependent on the altitude, e.g., the flux at aircrafts flying altitude is more than 300 times larger than at sea level. For the current technology nodes, neutrons are the main source of radiation-induced SETs. In advanced smaller technology node, it is expected that the contribution of protons induced SETs could increase.

3.1.2 SET Masking Effects and Width Variations

The SETs generated in digital designs due to some particle strikes may induce soft errors at the primary outputs if there exist an open logic path from the striking node to the output. Moreover, to propagate, SETs must have sufficient amplitude and duration. Fig. 3.1 shows a chain of four NAND gates. *Out_1*, *Out_2*, and *Out_3* represent the outputs of *G2*, *G3* and *G4*, respectively. It is assumed that a particle struck at *G1*. Contemporary techniques consider the following scenarios for SET characteristics variation while propagating :

1. SETs can be logically masked by a gate if at least one of its inputs is set at a controlling logic value (e. g., '0' for a NAND gate).
2. SETs arriving outside of sequential elements latching window are masked (i.e., temporally masked) [10].

3. If an SET is not logically nor temporally masked then it is propagation to the output is decided based on its width and amplitude. The propagation of such SETs is categorized into the following cases :

   (a) If SET amplitude and width are above the threshold level of the subsequent combinational circuit. Therefore, SET propagates through all the subsequent gates without losing its strength (first scenario in Fig. 3.1).

   (b) If SET amplitude and width are below the threshold level, then it may still propagate. However, subsequent combinational gate attenuates SET (second scenario in Fig. 3.1).

   (c) If SET amplitude and width are sufficiently lower than the threshold level, SET can be completely masked (third scenario in Fig. 3.1).

   (d) If SET amplitude and width are enough for it to propagate then its width may broaden while propagating. This is depicted in the fourth scenario in Fig. 3.1. This scenario was first observed in [53] and it has recently gained more attention and was addressed in [12, 54, 20, 55].

If an SET is latched in sequential elements at the end of the clock cycle where it occurred only logical masking factor can mask the error in the subsequent cycles.

3.2 Formal Verification Methods

3.2.1 Multiway Decision Graphs

Multiway Decision Graphs (MDGs) are an extension of Binary Decision Diagrams (BDDs) in the sense that they represent and manipulate a subset of first-order logic formulae suitable

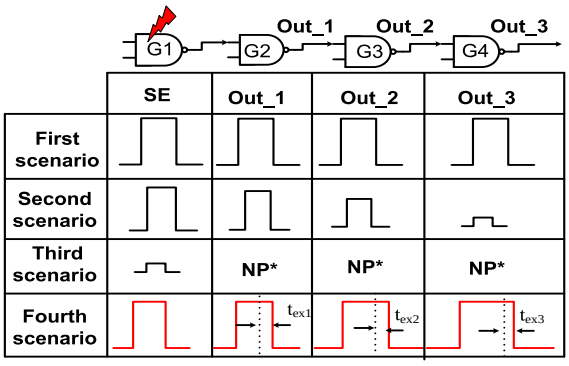


Figure 3.1 Different Scenarios for SET Pulse Propagation. (*NP Means SET Pulse is Not Propagating)

for large data path circuits. One of the advantages of MDGs over BDDs is that a data value can be represented by a single variable of abstract sort, rather than by concrete Boolean variables, and a data operation can be represented by an un-interpreted function symbol. MDG and ROBDD are alike in the sense that both require a fixed order of node labels along all paths. In ROBDDs all variables are Boolean. But in MDGs every signal/variable must belong to an appropriate sort, also a type definition must be provided for all functions.

The enumerated data type in MDG facilitates modeling both the logical masking and the SET pulse width variation (broadening or attenuation) in a single decision diagram. The data operation in MDGs can be represented by a function symbol, which can apply to a pre-defined data type.

3.2.2 MDG-Tool Set

A well known academic tool set for the formal verification of digital systems is based on MDG [57]. The MDG tool has been used to verify various types of complex systems [78, 79, 80]. It includes application procedures for combinational and sequential equivalence checking [57], model checking [81], and invariant checking [57]. Our methodology utilizes invariant checking, which is a formal verification approach that performs reachability analysis to check the potential of system failure due to a particular fault under specified conditions. If an error exists, then an example is generated to demonstrate the condition under which the system may fail (such examples where a property fails are commonly called counterexamples of some property). Moreover, the invariant checking tool allows analyzing the propagation of the injected SET pulse in one version of the design without the need for two versions of the design (faulty and error free version) as in BDD based techniques [49, 71].

The MDG tool uses a *prolog-style* hardware description language called the MDG-HDL [57]. This language supports structural, behavioral and mixed styles of coding. A structural specification is usually a netlist of components connected by signals. A behavioral description consists of a tabular representation of the transition and output relations in the form of a truth table.

3.2.3 PRISM Model Checker

PRISM [82] is a probabilistic symbolic model checker developed at the University of Birmingham. It works with its own high-level modeling language, which is written in form of state-based modules, each composed by a set of guarded commands. PRISM uses DD and Multi-Terminal Binary Decision Diagrams (MTBDDs) [83] to construct and compute the

reachable states of even very large probabilistic models.

PRISM is a flexible tool that allows working with probabilistic real-life models as it allows for the specification of probabilities inside the model and in the properties. Additionally, this model checker evaluate the probability of given property failing. Moreover, PRISM allows step-by-step simulation where the user can chose the simulated variables on the system as well as their initial values. The simulation may be guided, where the user manually selects the next state transition, or random, where the user selects the number of random transitions PRISM should simulate.

3.2.4 Satisfiability Modulo Theories

The advent of Satisfiability Modulo Theories (SMTs) [84] solved the problem of being restricted to pure Boolean representations, which fail to represent many classes of systems. SMT is an extension of the SAT decision problem, where the formula is expressed in first-order logic, with associated background theories. Based on the employed theories to model the problem, specialized algorithms are combined to solve it. The modeling requirements of the target application dictate the choice of theory. For example, when modeling hybrid systems, where variables with real values are required, the theory of linear arithmetic is commonly used. In software verification, the theories of arithmetic over integers and arrays are commonly used. On top of SMT solvers, there are many different verification algorithms that are used to solve different engineering problems, such as analog circuit verification, RTL functional verification, and software model checking.

3.3 Digital Design Flow

Without the layout of an integrated circuit, modeling of electrical masking and SET characteristics variations cannot be fully accurate, due to the lack of exact loading and timing details. This can lead to some SETs not being detected, thereby the calculated quantitative estimates are not accurate and they can serve only as approximations. In this thesis (Chapter 6), the proposed gate level analysis involves generating and characterizing the timings of the layout of a design using EDA tools. The main steps followed to characterize the timings of the cells in the layout are reported in Alg. 1. The typical inputs of the post-layout characterization are (i) a gate level netlist generated from a synthesis tool, (ii) a target technology timing file (i.e., lib file), and (iii) a set of timing constraints used to drive the place and route process, which is reported as the Synopsys Design Constraints (SDC) file. Place-and-route tools create a layout by utilizing the layouts of pre-defined standard cells such that the inter-

connections between cells, as specified in the netlist, are preserved. Place-and-route tools also take into account the detailed timing issues that arise from the actual location of the various cells in the layout. In this thesis, the generation of the design layout and the extraction of its parasitics is done using the *SOC encounter* Cadence tool.

Algorithm 1 Design Timing Characterization From Layout

Inputs : Netlist.v, SDC_file.sdc, Tech.lib, LEF_lib.lef, TPTs.
Outputs : Layout, SPEF_layout.spef, SDF_layout.sdf
Tools : SOC Encounter Cadence, Synopsys PrimeTime
procedure *LayoutTimingExtraction*
    ImportDesing(Netlist, LEF_lib, Tech) ;
    SetUp_Timing(delay_corner, SDC_file, Tech) ;
    FloorPlanning(Core_Size, Core_Bound, Core_utiliz) ;
    PowerPlanning(Add_Ring, Add_Strip) ;
    Place & ***Route***(SRoute, PlaceDesign, RouteDesign, AddFiller) ;
    SPEF_layout ← ***ExtractRC***(RC_corner, Layout) ;
    SDF_layout ← ***WriteSDF_PT***(Netlist, Tech, SPEF_layout)

It is preferable that the timing characterization step employs a highly accurate device-level simulator such as HSPICE [85] or some static timing software at the transistor level or gate level. In this thesis, timing characterization is performed at the gate level using the static timing analysis software from Synopsys (i.e., *PrimeTime* [86]). To do that, (i) the gate netlist, (ii) the detailed layout parasitics extracted in the Standard Parasitic Exchange Format (SPEF) file, and (iii) the timing model lib file are required. The results of this process is the detailed layout timing of each gate, which is characterized into a Standard Delay Format (SDF) file.

CHAPTER 4 ARTICLE 1 : NEW INSIGHTS INTO THE SINGLE EVENT TRANSIENT PROPAGATION THROUGH STATIC AND TSPC LOGIC

Summary of the Chapter

In this chapter, the work which was done in the first phase of this thesis is reported. In this phase, different circuit level simulations were performed to understand the impact of the propagation paths, input patterns, and re-convergent paths on the SET characteristics. Initially, the idea was introduced (and subsequently published) in a paper entitled "Investigating the Impact of Input Patterns, Propagation Paths and Re-convergent Paths on The Propagation Induced Pulse Broadening" which was published in the 14th IEEE Conference on Radiation Effects on Components and Systems (RADECS) on 2013. In order to confirm the observed behaviors this analysis was performed for different circuits from both static and TSPC logic families. Further analyses were performed and all the results were reported in a journal paper which was published in the IEEE Transactions on Nuclear Science on 2014. ***The published journal paper is reproduced in this chapter.***

Title : New Insights Into the Single Event Transient Propagation Through Static and TSPC Logic

Authors—Ghaith Bany Hamad, Syed Rafay Hasan, Otmane Ait Mohamed, and Yvon Savaria

Abstract— An investigation of the Single Event Transient (SET) characteristics (amplitude and width) variation while propagating through static and True Single Phase Clock (TSPC) logic is presented. The dependencies of the SET characteristics on the input patterns, propagation paths, pulse polarity, diverging paths, and re-converging paths are investigated. New insights on the propagation induced pulse broadening (PIPB) phenomenon in different combinations of static and TSPC logic are reported. The worst and the best propagation paths for SET pulse broadening and attenuation are identified. Our results demonstrate that SET pulses propagation can lead to Byzantine faults as they propagate through diverging paths. A new way to abstract all possible interpretations of the SET induced Byzantine fault phenomenon is proposed.

Index Terms—propagation induced pulse broadening (PIPB), Soft Errors, Broadening, Input Pattern, Propagation Path, True Single Phase Clocked (TSPC), single event transient

(SET), Byzantine Faults, Diverging Paths, Re-converging Paths.

4.1 Introduction

Soft errors, induced by radiations, are an increasing issue impacting the reliability of CMOS Integrated Circuits (ICs) adopted not only in safety-critical applications, notably found in space and avionic environments, but also in ground-level applications [10], [11]. The progressive shrinking of device sizes in advanced processing technologies, which have scaled from 0.5 μm to 32 nm in less than two decades, leads to miniaturization and performance improvements, but on the other hand, ultra-deep sub-micron technologies are more vulnerable to soft errors [10].

Several research activities have been done recently in order to analyze the radiation effects in digital circuits at different abstraction levels. Some early work on analyzing soft error sensitivity in digital circuits was based on Monte-Carlo simulations such as the soft error Monte-Carlo modeling program (SEMM) [40]. Other techniques have also been proposed such as binary decision diagram (BDD)-based techniques [48], a combination of reduced-order binary decision diagrams (ROBDDs) and algebraic decision diagrams (ADDs) [49], and Boolean satisfiability solvers (SAT-solvers) [6]. State-of-the-art techniques analyze the susceptibility of digital circuits to soft errors by only modeling the masking effects that can prevent a single event transient (SET) pulse in digital designs from propagating : logical masking (related to the logic operation of the gate), electrical masking (related to the electrical property of logic gates), and latching window masking (related to the sensitive time window of the sequential elements) [10].

However, contemporary techniques (such as [40] - [6]) are not sufficiently accurate in modeling soft error propagation, as these techniques omit the possibility that a soft error pulse could broaden while propagating. In [53, 87], the SET pulse broadening phenomenon was first observed and partly characterized. The results showed that pulse broadening can be attributed to the speed difference between the rising and the falling edge along a given path [53]. Recently, the distribution of SET pulse width was measured in long SOI chains irradiated with broad beam heavy ions. The propagation-induced pulse broadening (PIPB) effect was experimentally modeled and measured in SOI inverter chains for the first time in [13]. Subsequently, many researchers have analyzed the PIPB effect in SOI and bulk inverter chains and its dependencies on active loading (fan-out) and passive loading (interconnect) of the target circuits such as [12]-[55]. A direct relationship has been observed between the PIPB effect and both the node capacitance (capacitive loads) and the transistor size [12]-[55]. Other authors proposed a gate-level SER estimation method in [65]. This method takes into

account both the masking effects and the PIPB effect. Results indicate that the PIPB effect increases the soft error rate (SER) [65]. Nonetheless, existing state-of-the-art techniques are unable to analyze the effects of the propagation paths, the re-converging paths, and the input patterns on the PIPB. Moreover, the SET pulse generation and propagation in dynamic logic family such as the True Single-Phase-Clocked (TSPC) logic have not been fully analyzed. The TSPC logic style has two main features : 1) the combinational functionality is combined with the storage behavior and thus offers low transistor count. 2) It requires just a single clock which simplifies clock generation and clock distribution.

In order to overcome these shortcomings, we explore further the SET pulse characteristic variation while propagating through both static and TSPC logic. This work is distinct in the following ways : 1- For the first time, the relationship between the specific logic structure and the SET pulse width broadening or attenuation is investigated. Worst and best propagation paths are identified for the analyzed designs. 2- We investigate the impact of both the input patterns and the SET pulse polarity (negative ($1 \rightarrow 0 \rightarrow 1$) or positive ($0 \rightarrow 1 \rightarrow 0$)) on the PIPB. 3- The impact of re-converging paths on the SET pulse propagation is investigated. Pulses may re-converge and overlap at a gate in a circuit if multiple paths exist between the particle striking node (affected node) and the re-converging gate. 4- The analysis of diverging paths effect on SET pulse propagation provides direct evidence that it can lead to Byzantine faults, which are defined as faults presenting different symptoms (or logic interpretation) to different observers [88], [89], [90]. 5- The required timing conditions for the generation and the propagation of the SET pulse through TSPC logic are abstracted. Finally, our investigation allows designers to make informed decision when approximating soft error propagation possibilities in micro-architectures.

The rest of this paper is organized as follows. Section 5.2 identifies the problem we are addressing. In Section 4.3, we investigate the SET pulse characteristics variation in static and TSPC logic due to the input patterns and the pulse polarity. The impacts of the propagation path, the diverging paths, and the re-converging paths on the SET pulse characteristics are analyzed in Section 4.4. Section 8.8 concludes this work.

4.2 Problem Formulation

The generated SET pulse in digital designs due to a particle strike might induce a soft error at the primary output if there exists an open logic path from the striking node to the output. Contemporary techniques model SET pulse propagation by considering the following scenarios :

1- If the SET pulse amplitude and width are sufficiently small, the SET pulse is electrically

masked.

2- If a SET pulse width and amplitude are below a threshold level, then the SET pulse may still propagate. However, the subsequent combinational gates will attenuate the pulse amplitude and width until it vanishes.

3- If the SET pulse width and amplitude are above a threshold value then it may broaden while propagating through digital designs [53]-[12]. Moreover, in digital designs, multiple factors can effect the SET pulse propagation which are described below :

- *Fan-out :* the output of a logic gate is connected to the input(s) of one or more logic gates. This increases the capacitive load on the driving gate. Note that fan-out increases the propagation threshold, but this phenomenon can also lead to broadening of a SET pulse width, due to the difference between the rise and the fall times. This factor has been analyzed for the static logic in [12]-[20].
- *Diverging and re-converging paths :* the SET pulse propagation can lead to Multiple Event Transients (MET) if it propagates through a diverging node. Moreover, due to re-converging paths, the width of a SET pulse, which propagates through different paths between the fault striking node and the re-converging gate, may combine or overlap when arriving simultaneously at the input of a re-converging gate.
- *Input patterns and SET pulse polarity :* the variation of the propagation delay for different input patterns of multiple inputs static gates is well known to the community ([30], Chapter 6). Hence, it is possible that the SET pulse width in a static gate varies for different input patterns. Moreover, the variation in a SET pulse characteristics may depend on its polarity.
- *Timing Constraints :* In dynamic logic the characteristics of the SET pulse may vary due to the particle strike time and the dynamic characteristics of the gate through which the SET pulse propagates. In this work, we analyze the impact of all these timing constraints on a SET pulse characteristics for the TSPC logic. This logic was first proposed to deal with the skew problem in dynamic logic, such as clocked CMOS [31], domino [32], and NORA logic [33]. In TSPC logic, it is possible to achieve high clock frequencies because of the simplifications of the clock distribution and the elimination of the phase overlapping problems [34, 35, 36, 37]. Different possible implementations of TSPC logic, which use low number of transistors, have been presented in [35]. Furthermore, TSPC logic has been used in high operating frequency dividers and to reduce power dissipation [38, 39].

To efficiently design digital systems robust to soft errors, engineers should consider soft error effects as early in the design cycle as possible. Therefore, researchers came up with SET propagation models operating at higher abstraction levels, such as gate levels and RTL levels

[48]-[6]. However, many of these models have not considered the PIPB effect. Recently, in [65], authors have proposed a model considering the PIPB effect at the gate level. This model ignores the effects of re-converging paths, propagation paths, and input patterns and only deals with static CMOS logic. Our work analyzes the effects of these unexplored phenomena (namely, propagation path, re-converging paths, diverging paths, and input patterns). We analyze these effects in CMOS circuits, which are sufficiently accurate to serve as a foundation for modeling such phenomena at the gate level and at higher abstraction levels.

4.3 SET Characteristics Variation in Static and TSPC Logic

In this section, the SET pulse characteristics (amplitude and width) variation in static and TSPC logic are investigated. Electrical simulations are performed using HSPICE and a 65 *nm* CMOS technology library from TSMC.

4.3.1 Static Logic

The SET pulse characteristics vary in static logic gates due to the input patterns because each input combination leads to different equivalent internal resistance and capacitance. The schematics of the analyzed 4-input NAND and the 4-input NOR gates are depicted in Fig. 4.1. Positive and negative SET pulses are injected at the primary inputs. t_{pLH} refers to the time for the output transition from logic '0' (low) to logic '1' (high), while t_{pHL} refers to the time for high to low output transition. The input and the output signal width are measured between the 50 % transition points of the waveform. We started by matching the transistors and finding the optimal transistors size for a fan-out of 4. The results reported in Tables 4.1, and 4.2, for the NAND and the NOR gates shown in Fig. 4.1, are obtained by applying a specific input pattern. The input SET pulse width is equal to 100 *ps*. The second columns in both Tables I, and II depict the simulated input pattern and the node where the pulse is injected.

NAND Gate

The CMOS implementation of a NAND gate is shown in Fig. 4.1a. In order to avoid the logical masking effect, when a pulse is injected at one input node then all the other primary inputs must be at logic 1. Δt_p in the last column of Table 4.1 represents the difference between t_{pHL} and t_{pLH} ($t_{pHL} - t_{pLH}$). Simulation results in Table 4.1 lead to the following observations : 1- the NAND gate attenuates the negative pulse and broadens the positive pulse. 2- The amount of the attenuation, broadening, t_{pHL}, t_{pLH}, and Δt_p, can be explained

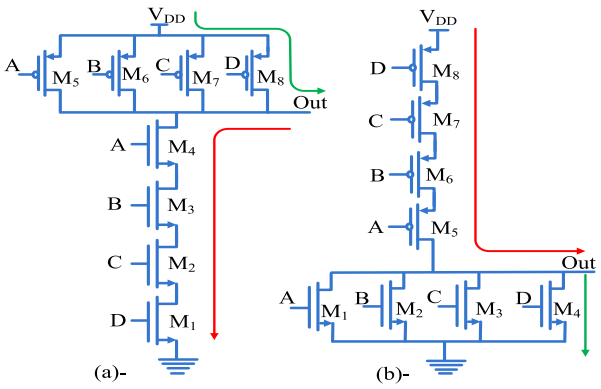


Figure 4.1 CMOS Transistor Level Implementation of (a)- a 4-Input NAND Gate, (b)- a 4-Input NOR Gate.

based on the different RC delay characteristics of the CMOS logic, which is dependent on the input patterns ([30], Chapter 6). For example, the fifth row in Table 4.1 shows the case when a SET pulse of positive polarity is injected at node A of Fig. 4.1a, which resulted in the least SET pulse broadening (*6 ps*). Whereas the most broadening is observed when the same SET pulse occurs at the transistor nearest to ground. It is observed to be *12 ps* and tabulated in the last row of Table 4.1. Similar behavior has been observed for the AND gate (recall that a CMOS AND is a NAND combined with an inverter and polarities on output are inverted).

NOR Gate

The CMOS implementation of a NOR gate is depicted in Fig. 4.1b. To make sure that logical masking does not prevent the SET pulse from propagating, all the other primary inputs are set to non-controlling logic state (logic '0'). Table 4.2 shows the output pulse width of the NOR gate depicted in Fig. 4.1b. Δt_p in the last column of Table 4.2 represents the difference between t_{pLH} and t_{pHL} ($t_{pLH} - t_{pHL}$). Simulation results in Table 4.2 lead to the following observations : 1- the NOR gate attenuates the positive pulse and broadens the negative pulse. 2- Similar to the NAND gate the amount of attenuation, broadening, t_{pHL}, t_{pLH}, and Δt_p, are dependent on the input patterns. For example, when a SET pulse of negative polarity is injected at node A of Fig. 4.1b the SET broadens the least (*8.03 ps*). Whereas the most broadening is observed (*14.5 ps*) when it propagates through input D as depicted in Table 4.2. Similar behavior has been observed for the OR gate.

4.3.2 TSPC Logic

In this section, we analyze all possible SET pulse generation scenarios in TSPC logic. Moreover, we analyze the impact of the input patterns and the pulse polarities on the SET pulse characteristics in TSPC logic. The circuit schematic of a split output TSPC buffer is shown in Fig. 4.2. When a particle strikes a vulnerable node of a TSPC logic gate it may generate a SET pulse. If this pulse reaches the output, it will cause a soft error. In this analysis, the nodes vulnerable to radiation are the surroundings of the reverse biased drain junctions of a transistor biased in the *OFF* state. If a strike occurs at the drain of a NMOS transistor, then a negative SET pulse $1 \rightarrow 0 \rightarrow 1$ is generated (substrate is connected to Ground). If the strike occurs at the drain of a PMOS transistor then a positive SET pulse ($0 \rightarrow 1 \rightarrow 0$) is generated (substrate is connected to V_{DD}).

In this work, the maximum input voltage that can be interpreted as logic '0' is considered to be V_{IL}. Similarly, the minimum input voltage that can be interpreted as a logic '1' is considered to be V_{IH}.

The first detailed analysis consisted of analyzing the impact of the input pattern on SET pulse generation. In the TSPC buffer depicted in Fig. 4.2a, a particle strike can occur either while the clock is ON (M2 is conducting) or when the clock is OFF (M2 is disconnected).

Table 4.1 SET Pulse Propagation Through a 4-Input NAND Gate.

	Input Pattern	Input Width (ps)	Output Width (ps)	t_{pLH} (ps)	t_{pHL} (ps)	Δt_p (ps)
Attenuation	$B = C = D = 1$, $A = 1 \rightarrow 0 \rightarrow 1$	100	95	8.4	13.39	4.99
	$A = C = D = 1$, $B = 1 \rightarrow 0 \rightarrow 1$	100	93.5	12.5	19	6.5
	$A = B = D = 1$, $C = 1 \rightarrow 0 \rightarrow 1$	100	91	13.6	22.63	9
	$A = C = B = 1$, $D = 1 \rightarrow 0 \rightarrow 1$	100	89	13.6	24.6	11
Broadening	$D = B = C = 1$, $A = 0 \rightarrow 1 \rightarrow 0$	100	106	7.45	13.45	6
	$A = C = D = 1$, $B = 0 \rightarrow 1 \rightarrow 0$	100	108	10.17	18.18	8.01
	$A = B = D = 1$, $C = 0 \rightarrow 1 \rightarrow 0$	100	111	10	21	11
	$A = C = B = 1$, $D = 0 \rightarrow 1 \rightarrow 0$	100	112	10.1	22.1	12

Table 4.2 SET Pulse Propagation Through a 4-Input NOR Gate.

	Input Pattern	Input Width (ps)	Output Width (ps)	t_{pLH} (ps)	t_{pHL} (ps)	Δt_p (ps)
Attenuation	$A = C = B = 0$, $D = 0 \rightarrow 1 \rightarrow 0$	100	85	27.85	12.86	14.99
	$A = B = D = 0$, $C = 0 \rightarrow 1 \rightarrow 0$	100	87.5	20.39	8.39	12
	$A = C = D = 0$, $B = 0 \rightarrow 1 \rightarrow 0$	100	90	25.5	15.4	10.1
	$D = C = B = 0$, $A = 0 \rightarrow 1 \rightarrow 0$	100	91.9	14.2	6.15	8.05
Broadening	$A = B = C = 0$, $D = 1 \rightarrow 0 \rightarrow 1$	100	114.5	24.8	10.25	14.55
	$A = B = D = 0$, $C = 1 \rightarrow 0 \rightarrow 1$	100	112	23.43	11.4	12.03
	$A = C = D = 0$, $B = 1 \rightarrow 0 \rightarrow 1$	100	110	19.53	9.03	10.5
	$D = C = B = 0$, $A = 1 \rightarrow 0 \rightarrow 1$	100	108.03	14.1	6.1	8

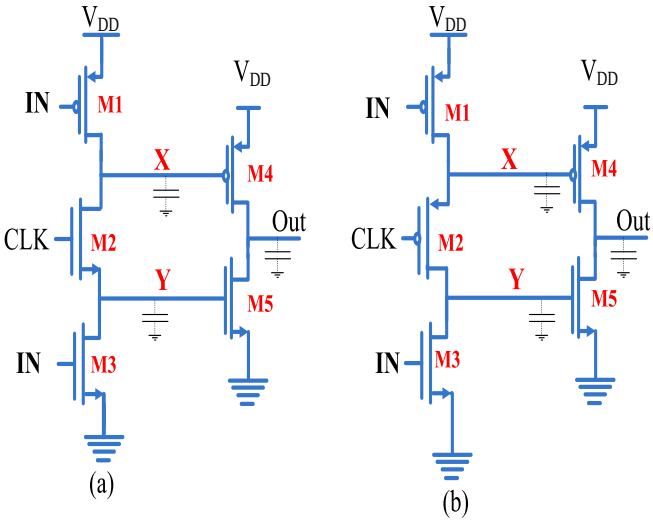


Figure 4.2 Transistor level schematic of TSPC buffer gate (split output implementation). (a) Positive latch (b) Negative latch.

The SET pulse generation scenarios for the TSPC buffer of Fig. 4.2a while the clock is ON are summarized in Table 4.3. The first, second, third, fourth, and fifth columns present the possible logic levels of *IN*, *CLK*, internal nodes (Y, X), and *Out* in Fig. 4.2a, respectively.

First and fourth rows provide the error free scenarios. Table 4.3 summarizes the relationship between input patterns and the generation of the SET pulse. For example, if the applied input pattern is (IN = 1, CLK=1, second and third row of Table 4.3), then under this condition in normal operation, X and Y should be at logic '0'. If a particle strikes at M1, then a positive SET pulse is generated at node X, and may propagate to node Y as M2 is on (CLK = 1). This can result in the generation of a negative pulse at the output. Another possibility of the SET pulse generation, with CLK =1, is when a particle strikes at *M5* (see third row in Table 4.3) then a negative SET pulse is generated at the output. In this case, the affected node is the *Out* node as shown in Fig. 4.2a.

Table 4.3 SET Pulse Generation Scenarios for the TSPC Buffer When the Clock is ON.

IN	**CLK**	**X**	**Y**	**Out**
1	1	0	0	1
1	1	$0 \rightarrow 1 \rightarrow 0$ at M1	$0 \rightarrow 1 \rightarrow 0$	$1 \rightarrow 0 \rightarrow 1$
1	1	0	0	$1 \rightarrow 0 \rightarrow 1$ at M5
0	1	1	1	0
0	1	$1 \rightarrow 0 \rightarrow 1$	$1 \rightarrow 0 \rightarrow 1$ at M3	$0 \rightarrow 1 \rightarrow 0$
0	1	1	1	$0 \rightarrow 1 \rightarrow 0$ at M4

Table 4.4, summarizes the SET generation scenarios for a TSPC buffer (positive latch) when *CLK* (gate of M2 in Fig. 2a) is at logic low (OFF). In this Table, the possibility of flipping the stored logic value at the *Out* node due to soft error is investigated. The first, second, fourth, and fifth columns depict the logic levels of the IN, previous state of the *Out* (*Out(t-1)*), internal nodes (X, Y), and current *Out* node in Fig. 4.2a. As depicted in the first and fifth rows in Table 4.4, it is possible for a SET pulse to be masked (no propagation) after it gets generated at a vulnerable node. For example, in the scenario shown in the first row, node X is charged ($0 \rightarrow 1$) if a particle strike at *M1* (under the conditions that the *out(t-1) = 1*, *IN = 1*, and the stored value at the internal node X is 0). However, there is no path for this pulse to propagate to the output (M4 is OFF). A SET pulse can be generated at the *Out* node due to a strike at *M4* or *M5*, as shown in the second and third row of Table 4.4. Moreover, when *out(t-1) =0*, *IN=0*, and *CLK=0*, a strike at *M2* (node X) can generate a positive pulse at the output as shown in the fourth row in Table 4.4.

To further our analysis, we investigated how a particle strike timing impacts on the SET pulse width. For this purpose, consider the case shown in the second row in Table 4.3, where

IN=1, *CLK*=1, and a particle strikes at *M1*, which generates a negative pulse at the output. Due to the strike time variation, the width of this SET pulse can be one of the following : 1- If the particle strike time is within the *CLK* hold time, then the SET pulse width depends on the radiation strength (collected charge at the vulnerable node) as shown in Fig. 4.3a. 2- If the strike time is near the clock edge, then the SET pulse width extends from the strike time until the next clock cycle. This case is shown in Fig. 4.3b, where T_{SET} is the duration of the SET pulse. Moreover, the width of the SET pulse mainly depends on the clock period.

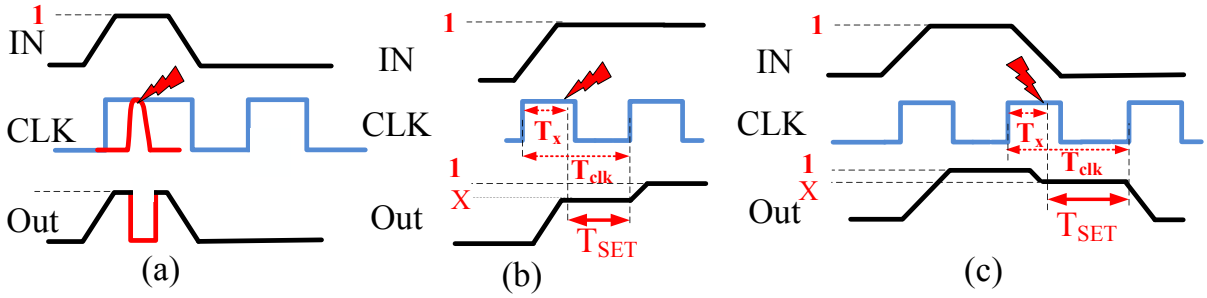


Figure 4.3 SET Pulse Width Variations Due to the Strike Time for the Scenario Shown in the 2nd Row of Table 4.3.

Table 4.4 SET Pulse Generation Scenarios for the TSPC Buffer When the Clock is OFF.

IN	out(t-1)	CLK	X	Y	Out
1	1	0	0 → 1 at M1	0	No propagation
1	1	0	0	0	1 → 0 → 1 at M5
0	0	0	1	1	0 → 1 → 0 at M4
0	0	0	1 → 0 → 1 at M2	1	0 → 1 → 0
0	0	0	1	1 → 0 at M3	No propagation

Table 4.5 shows the result of our findings regarding the impact of the strike time on the SET pulse amplitude. Two possible scenarios which lead to a SET pulse at the output, with an amplitude v where $V_{IL} < v < V_{IH}$, are depicted in both Table 4.5 and Fig. 4.3. The first scenario occurs when *CLK = 1*, *IN* is charging 0 → 1, X=Y= 1 → 0, and *Out*= 0 → 1. Due to a particle strike at *M1*, node *X* is charged again (0 → 1). Immediately, the clock (*CLK*) turns off, thus, *X* cannot switch to 0 (M2 off). Therefore, a SET pulse, with an ambiguous amplitude v, is stored at the output until the next clock cycle, as shown in Fig. 4.3b. The second scenario, depicted in the second column in Table 4.5, occurs when the *CLK = 1*, *IN*

Table 4.5 The Dependence of the SET Pulse Amplitude on the Particle Strike Time for the TSPC Buffer.

	First Scenario	**Second Scenario**
Current State	CLK =1 IN =0 → 1 X=Y=1 → 0 Out=0 → 1	CLK =1 IN =1 → 0 X=Y=0 → 1 Out=1 → 0
SET Generation Scenario	**1-** strike at *M1* ⇒ SET (0 → 1) at X	**1-** strike at *M3* ⇒ SET (1 → 0) at Y
	2- CLK switch OFF	**2-** CLK switch OFF
	3- Out = υ, Where $V_{IL} < \upsilon < V_{IH}$	**3-** Out = υ, Where $V_{IL} < \upsilon < V_{IH}$

is discharging 1 → 0, X=Y= 0 → 1, and *Out*= 1 → 0. Due to a particle strike at *M3*, node Y is discharged again (1 → 0). The clock immediately turns off *M2*, thus X cannot switch to 1. Therefore, a SET pulse, with an ambiguous amplitude υ, is stored at the output until the next clock cycle, as shown in Fig. 4.3c. Our analysis results of the SET pulse amplitude variation due to the strike time are illustrated in Table 4.6 for both scenarios shown in Table 4.5. In the reported analysis, the clock period is *300 ps*, while the clock ON (1) width is *150 ps*. υ and T_{set} are the amplitude and width of the SET pulse, respectively. The first column in Table 4.6 is the strike time T_X. The amplitude of the SET pulse also has an effect on its propagation. For example, in the analysis of the first scenario in Table 4.6, the amplitude of the SET pulse can be divided into three categories based on T_X : 1) if $T_X > 146\,ps$, then the SET pulse is interpreted as a logic '1'. 2) If $T_X < 140\,ps$, then the SET pulse is interpreted as logic '0'. 3) If $140\,ps < T_X < 146\,ps$, then the SET pulse has an amplitude which can be interpreted as '1' or '0' depending on the subsequent TSPC gate.

In summary, in TSPC logic, the input pattern, pulse polarity, strike node, and strike time impact the width of a generated SET pulse. Similarly, the negative latch TSPC buffer shown in Fig. 4.2b can be analyzed. In the next section, we investigate the propagation of SET pulses (including the ambiguous pulses) through different combinations of TSPC logic.

4.4 The Impact of the Logic Structure on the SET Pulse Characteristics

In this section, we investigate the impact of the propagation paths, the diverging paths, and the re-converging paths in both static and TSPC logic on the SET pulse characteristics. The results reported in this section rely on electrical simulations using HSPICE with 65 *nm* CMOS technology library from TSMC.

Table 4.6 Analysis of the SET Pulse Characteristics Variation Due to the Strike Time for the Scenarios in Table 4.5.

Strike Time (T_X) (ps)	First Scenario		Second Scenario	
	υ (V)	T_{set} (ps)	x_{set} (V)	T_{set} (ps)
120	0	180	1.2	180
130	0.330	170	1.11	170
140	0.500	160	0.959	160
144	0.625	156	0.805	156
145	0.700	155	0.653	155
146	0.809	154	0.515	154
147	0.950	153	0.391	153
150	1.2	150	0	150

4.4.1 Static Logic

In this work, we define the best propagation path (BPP) and the worst propagation path (WPP) as the logic paths between the vulnerable node and the primary outputs. In the BPP, SET pulse suffers from the smallest amount of broadening, where in the WPP, SET pulse suffers from the largest amount of broadening. In this section, four detailed analysis of the SET pulse propagation are performed. The first analysis consisted on analyzing the impact of the propagation paths on the SET pulse broadening and identifying the BPP and the WPP of both the NAND and the NOR chain. Positive and negative SET pulses are injected with different initial widths at certain distances from the output. We obtained three kinds of information :

1- Results in section 4.3 showed that a NAND gate attenuates negative pulses and broadens positive pulses with almost the same magnitudes. In case of propagating the SET pulse through a logic chain, where all gates have the same type and size (same t_{pHL}, t_{pLH}) no significant broadening has been observed. This is because Δt_p of the subsequent logic stages has the same magnitude but opposite polarity that broadening and attenuation alternate between subsequent stages. Therefore, in order to analyze one SET pulse propagation phenomena at a time (broadening or attenuation) through a chain of gates, a chain of 126 4-input NAND gates and a chain of 126 4-input NOR gates were simulated as shown in Fig. 4.4, where an inverter is added between every two cascaded gates. The choice of a chain of 126 gates is somewhat arbitrary. Our goal is to have a sufficiently complex combinational circuit to observe the phenomena of interest.

2- Similar to the single gate, our NAND chain also attenuates negative pulses and broadens positive pulses for all input widths. By contrast, the inverse occur in the NOR chain.

3- The amount of broadening is related to the propagation path as depicted Fig. 4.5. Both BPP and WPP for the NAND and NOR gates chain are depicted in Fig. 4.4a and 4.4b respectively. Note that for a positive pulse propagating through the NAND chain, the BPP is when the pulse propagates through the first input of each NAND gate (input A in Fig. 4.1a) as depicted in Fig. 4.4a. For the NOR gates chain, the BPP for a negative SET pulse, as shown in Fig. 4.4b, occurs when the SET pulse propagates through the fourth input of each NOR gate (which is input A in Fig. 4.1b). The PIPB factor shown in Fig. 4.5, e.g. 19 *ps* for the WPP of the NOR chain, can be determined from the slope of the curve, and it represents the broadening occurring at each gate stage. As shown in Fig. 4.5, the broadening increases with the logic depth, and the amount of broadening in the NOR chain is larger than in the NAND chain.

The second investigation we performed is analyzing the effect of both the supply voltage and the propagation path on the PIPB. Fig. 4.6 shows that the SET pulse broadens linearly with the number of gates separating the radiation strike node and the primary output. Fig. 4.6 also shows that the broadening increases when the supply voltage decreases for the WPP

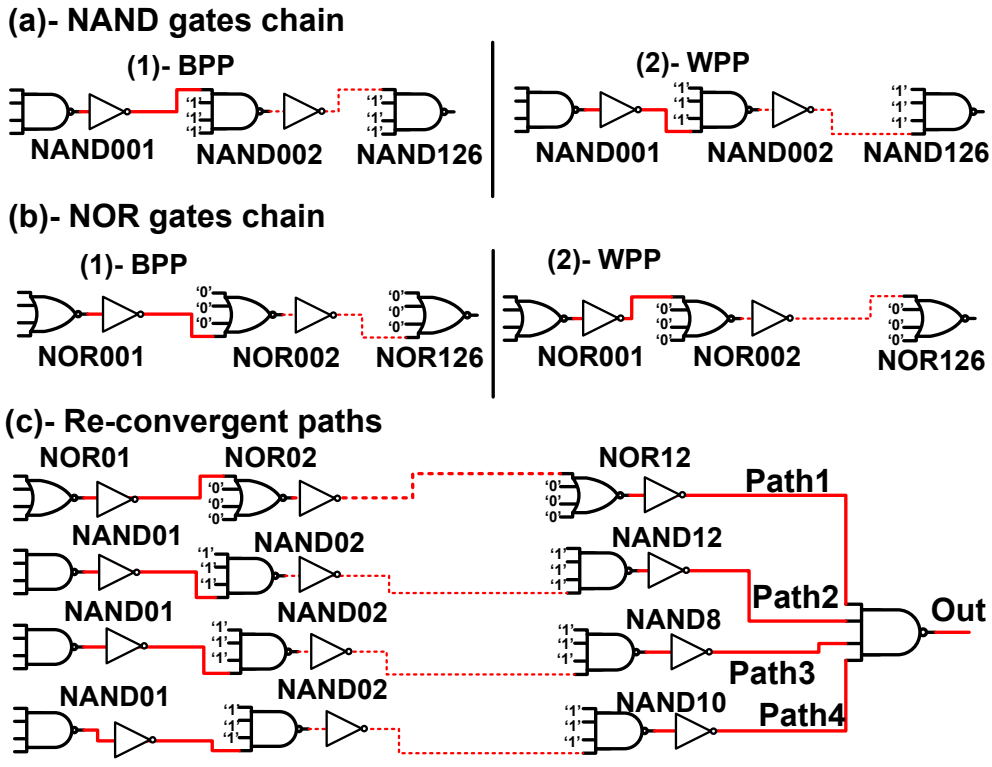


Figure 4.4 Schematic Description of the Chain of the NAND and NOR Gates. (a)- the BPP and the WPP for the NAND Gates Chain (b)- the BPP and the WPP for the NOR Gates Chain (c)- the Re-converging Path Combinational Design

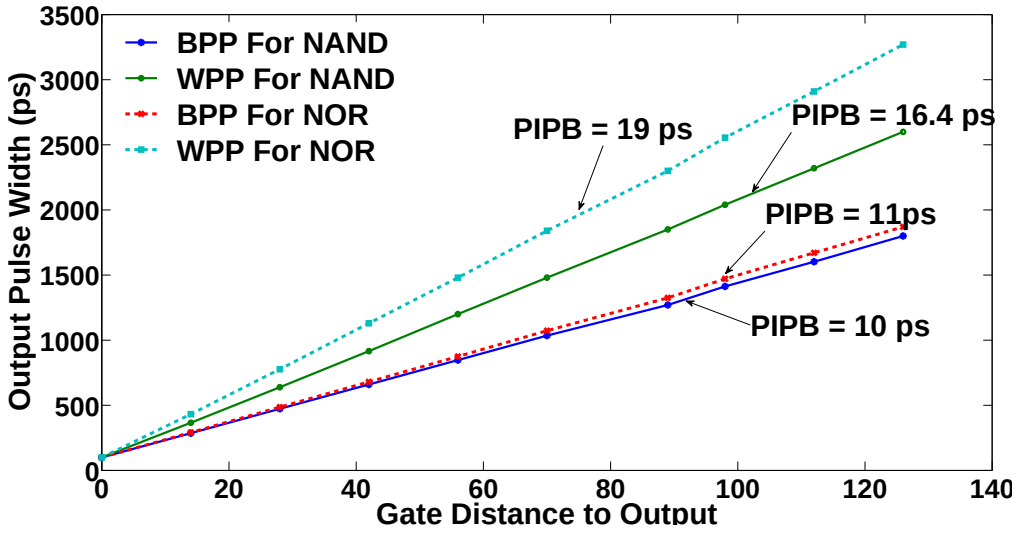


Figure 4.5 The WPP and the BPP for the NAND and NOR Gates Chain. Measured SET Pulse Width Versus the Strike Node Along the NAND Gates Chain. The Chain Supply Voltage 1.2 *V*. The Input SET Pulse Width is 100 *ps*.

and the BPP for both the NAND and the NOR chain. Largest expansion occurs when SET pulse propagates in WPP of the NOR chain and with 1.0 *V* supply voltage.

Fig. 4.7 shows the simulation results for the BPP and the WPP for the NAND chain shown in Fig. 4.4a at different process corners using 65 *nm* CMOS bulk technology. One can see the variation in the broadening, which results from different propagation paths (WPP and BPP) and different process corners (slow-slow (SS) and fast-fast (FF) corners). The largest broadening occurs as the SET pulse propagates in the WPP at the SS corner as depicted in Fig. 4.7.

The combinational design we built to perform our third detailed analysis of the effect of the re-converging paths on the PIPB is depicted in Fig. 4.4c. Four different length logic paths converge in a 4-input NAND gate. SET pulses (transition $0 \rightarrow 1 \rightarrow 0$ for the NAND paths and $1 \rightarrow 0 \rightarrow 1$ for the NOR path) are injected at the input of each path. Results in Table 4.7 indicate that due to the re-converging paths, the individual pulses propagating through *path1–path4*, are adding up at the re-converging gate (NAND gate). Thus, the broadening in the re-converging gate is much larger than what we observed in the NAND chain before. For example, around *185-ps* broadening in the NAND gate (see Table 4.7).

In summary, the SET pulse broadening phenomenon is significant because a relatively short pulse, just sufficient to propagate, can become arbitrarily long, provided the existence of a sufficient logic depth. Moreover, the contemporary transistor level (such as [20], [11], [91]) and higher abstraction level (as [65], [11]) modeling techniques do not consider the effects of

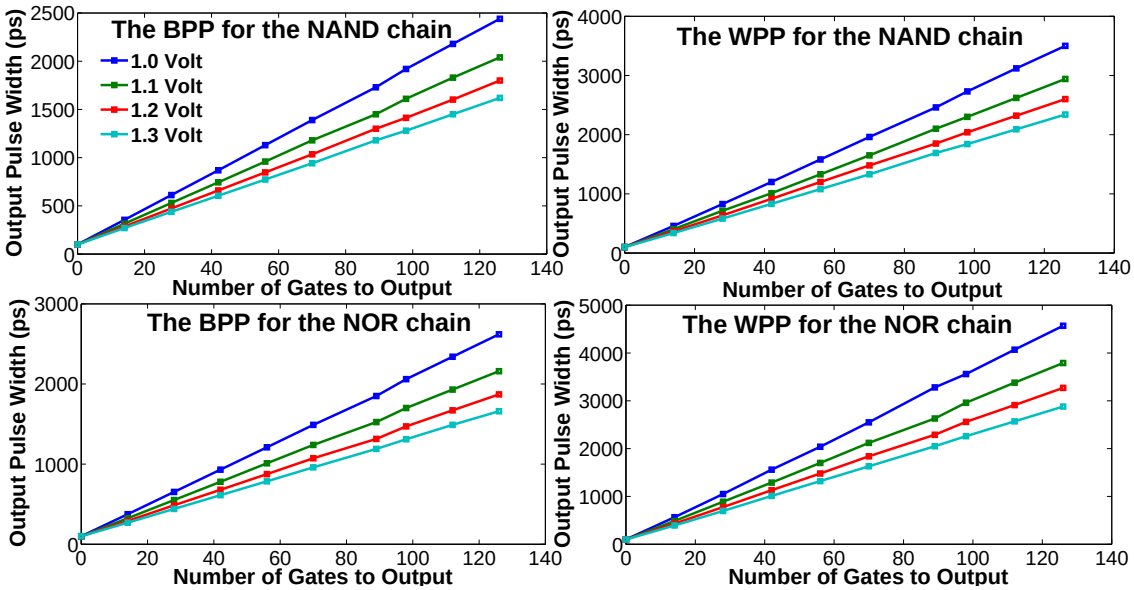


Figure 4.6 Measured SET Pulse Width Versus the Strike Position Along the NAND and the NOR Gates Chains When the Supply Voltage Varies From 1 V to 1.3 V and the Initial SET Pulse Width is 100 ps.

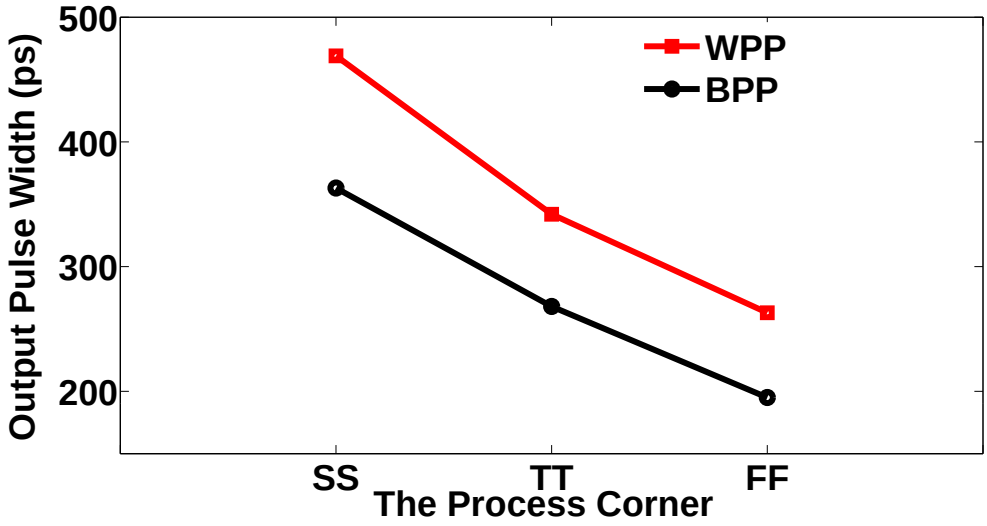


Figure 4.7 Simulation Results for the Best and Worst Propagation Path Among Different Corners for a Chain of 14 NAND Gates.

Table 4.7 The Effect of the Re-converging Paths on the PIPB.

Input Pulse Width (*ps*)	**Output Path1 Width (*ps*)**	**Output Path2 Width (*ps*)**	**Output Path3 Width (*ps*)**	**Output Path4 Width (*ps*)**	**Output Re-converging Width (*ps*)**
100	384	327	250	289	485
1000	1300	1240	1159	1199	1381
1200	1501	1440	1360	1399	1582
3500	3808	3745	3663	3704	3889

the propagation path, the polarity of the SET pulse, and the re-converging paths on the SET pulse propagation. Hence our analysis should lay the foundation of more accurate SET pulse propagation modeling techniques. Results in [65] demonstrate that assuming a rate of *1-ps* width broadening per stage of gates leads to increase the SER. Thus, SER is proportional to the pulse broadening. Our results make evident the relationship between the PIPB and both the propagation path and the input pattern, which will be beneficial in accurate SER estimation.

4.4.2 TSPC Logic

TSPC logic is mainly used in designing concurrent systems implemented as arrays of logic blocks operating in a pipeline manner. Doing so, idle times of the logic blocks are avoided and therefore the overall system performance is improved. In section 4.3.2, multiple SET generation scenarios in single TSPC buffer have been analyzed. In this section, we analyze the impact of the propagation paths, the diverging paths, and the re-converging paths on the SET pulse characteristics. This analysis is based on the worst generation scenarios where the particle strike occurs near the clock edge and has two properties. 1- Its width extends from the strike time until the next clock cycle as shown in Fig. 4.3. 2- Its amplitude can be 0, 1, or υ, where υ is any value between V_{IL} and V_{IH} as shown in Table 4.5. In this work, the *origin* gate is the TSPC gate where the particle strikes and the SET pulse is generated. Moreover, the *observer* is the subsequent TSPC gate where the SET pulse propagates to.

The first detailed analysis consisted on investigating SET pulse propagation behavior in the chain of TSPC buffers depicted in Fig. 4.8a. The N-block and the P-block in Fig. 4.8 are the positive and the negative latch shown in Fig. 4.2a, and Fig. 4.2b, respectively. Fig. 4.8a depicts a chain of of TSPC buffers which alternates the N-block and the P-Block. The dependence of SET pulse propagation on the subsequent gate is investigated in Fig. 4.9 for the design

shown in Fig. 4.8a. The error free behavior is shown in the left side of Fig. 4.9. Due to the timing conditions of the *observer* (N- or P-block), SET pulse can propagate through all the design stages. For example, for the chain in Fig. 4.8a, which alternates between N-blocks and P-Blocks, a SET pulse can expand to a full clock cycle duration at an internal node. This is analogous to flipping the state of a latch in a shift register. The fully stretched pulse can then propagate through the TSPC chain, as in a shift register, from the *origin* gate to the primary output.

Fig. 4.10 depicts the variation in the SET pulse width while propagating through the chain of TSPC buffers shown in Fig. 4.8a. Two observations can be made : 1) the main broadening in the SET pulse width occurs at the *origin* gate, where pulse extends from the strike time until the next clock cycle ; 2) the pulse width is directly related to the clock period.

The second detailed analysis consisted on analyzing the impact of diverging paths on SET pulse propagation. The TSPC logic design used in this analysis is depicted in Fig. 4.8b. If the SET pulse X_{set}, with an amplitude υ, propagates through a diverging node then different interpretations (0, 1, or υ) are observed for different subsequent gates (observers) depending on their thresholds, biases, and timing. This phenomenon is known in the literature as Byzantine faults, which is defined as a fault presenting different symptoms to different observers [88], [89], [90]. These different interpretations lead to different faults, one in each path, hence increasing the soft error rate at the primary outputs. The number of SET pulse interpretations at the diverging node are equal to 3^z, where z is the fan-out at the diverging node. For example, in Table 4.8 we abstracted the nine possible interpretations of the SET pulse by the subsequent gates assuming that the fan-out of the diverging node in Fig. 4.8b is 2. Moreover, the impact of the fan-out of the *observer* gate on the amplitude of the SET pulse is investigated, as shown in Fig. 4.11. In this analysis the generation scenarios in Table 4.5 are applied at the *origin* gate. The fan-out of the TSPC buffers, depicted in Fig. 4.8, (X_T1, X_F1, and X_T2) has been changed from 1 to 4. For each fan-out case the amplitude of the SET pulse was measured at the output of each cascaded gate. As depicted in Fig. 4.11, fan-out increases the threshold voltage of the logic gates. Therefore, fan-out impacts the interpretation of the SET pulse (0, 1, or υ). Furthermore, the SET pulse with υ logic level can propagate through multiple stages of logic and still remain at an ambiguous level.

The third detailed analysis consisted on analyzing the impact of the re-converging paths on the SET pulse propagation. In this scenario, the SET pulse interpretations at the diverging node converge at a subsequent gate in the design. Fig. 4.8b shows the case where two SET pulses (with amplitude υ) re-converge in 2-input TSPC OR gate. The re-converging gate (OR gate) interpretation of the re-converging pulses varies based on its timing, threshold,

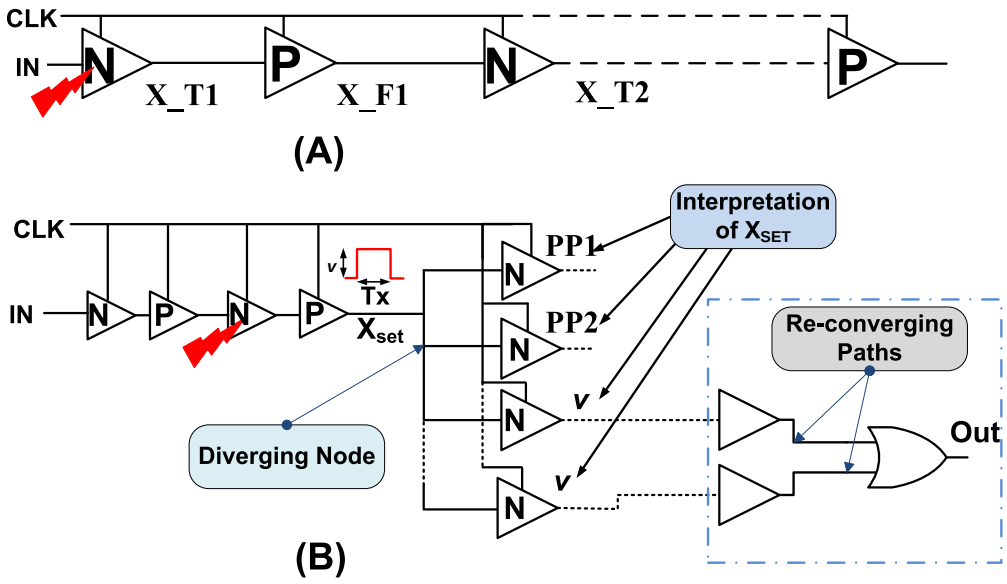


Figure 4.8 Schematic Description of the Combination of the TSPC Logic. (a)- Chain of Alternative N-block and P-block of TSPC Buffers, (b)- Diverging and Re-converging Paths.

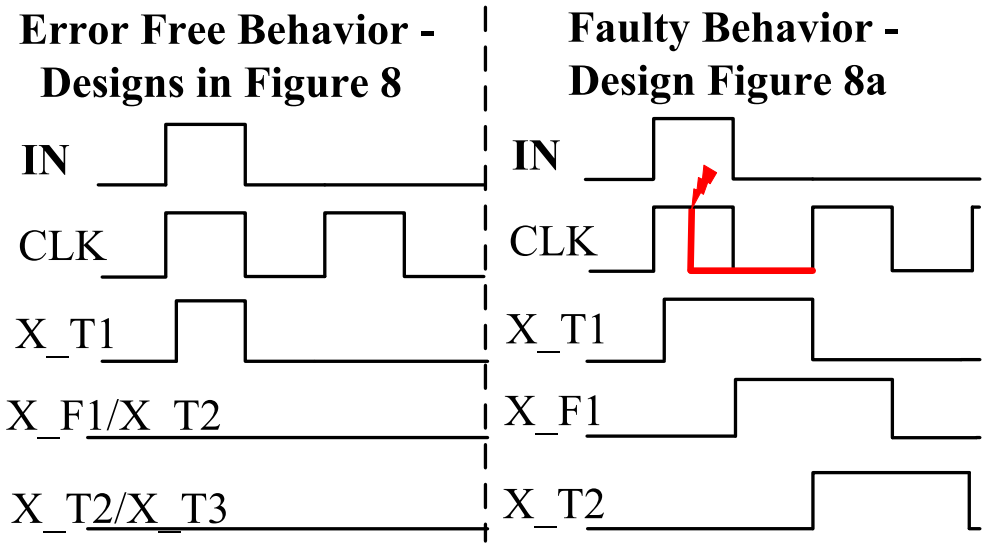


Figure 4.9 Simulation Results of the SET Pulse Propagation Through a Chain of TSPC Buffers Shown in Fig. 4.8(a).

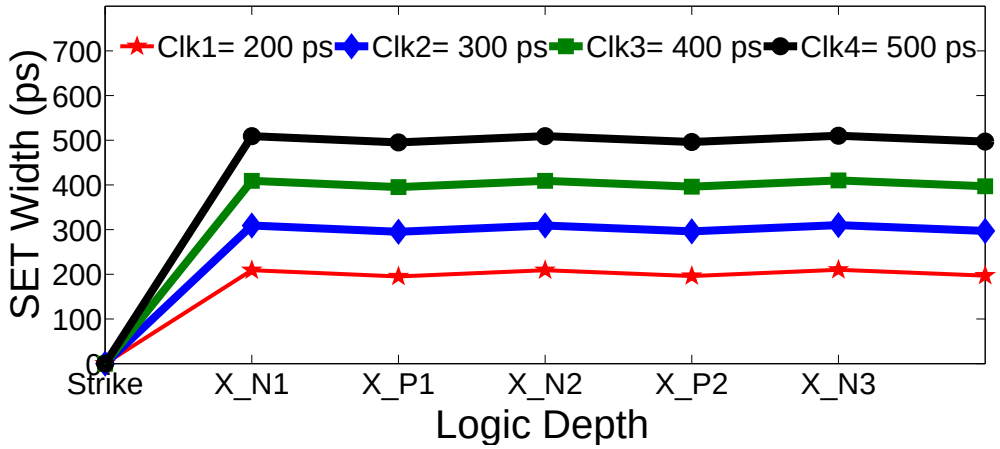


Figure 4.10 Variation in the Width of SET Pulses While Propagating Through a Chain of TSPC Buffer.

Table 4.8 Abstraction of the SET Pulse Propagation Induced Byzantine Fault Scenarios.

SET Byzantine Pulse	**PP1**	**PP2**
υ	0	0
υ	0	1
υ	1	0
υ	1	1
υ	0	υ
υ	1	υ
υ	υ	1
υ	υ	0
υ	υ	υ

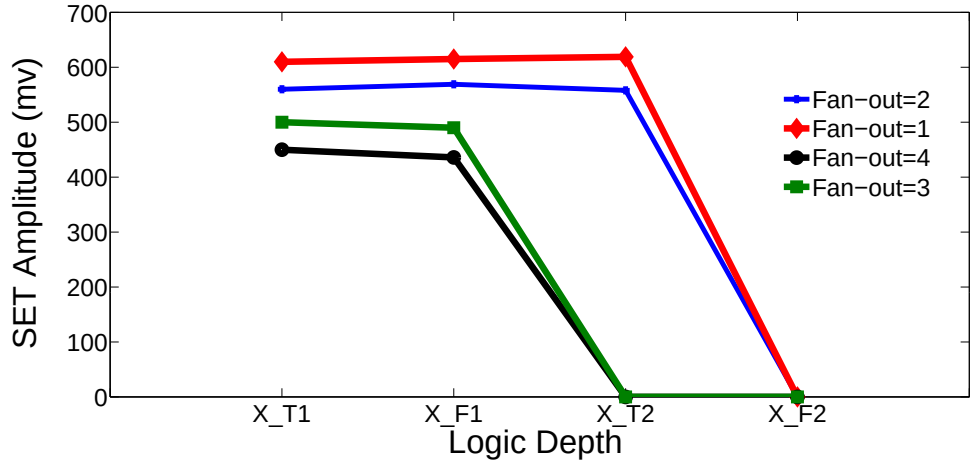


Figure 4.11 The Variation in the Amplitude of the SET Pulse While Propagating Through a Chain of TSPC Buffers.

and pulses amplitude. In Table 4.9 all possible interpretations of these two ambiguous pulses and the corresponding output of the re-converging gate are abstracted. Similarly, tables for other re-converging gates, such as AND, NAND, can be generated.

As a summary, in TSPC logic, the effect of electrical masking is not dominant. Therefore, the main masking effect that can prevent SET pulses from propagating in TSPC logic is logical masking. Our analysis has demonstrated that SET pulses propagation can lead to Byzantine faults (very serious for safety-critical systems [89]). Finally, it is important to develop a soft error tolerant technique which takes into account the impact of all these propagation scenarios.

4.4.3 Abstraction and Automation of the Proposed Analysis

The tendency of SET pulse propagation is worsening as geometric dimensions are scaled down [11]. Moreover, as our results demonstrate, the SET pulse characteristics are dependent on the propagation paths, the input patterns, the strike time, the converging node and the diverging node. Therefore, it is predicted that all these factors will continue to impact the SET pulse characteristics with technology scaling. Moreover, as explained before, the main source of SET pulse width variation is the imbalance between the T_{PLH} and the T_{PHL}. This imbalance is inevitable in most logic families, hence SET pulse width will vary in most logic families.

Digital designs are traditionally structured, i.e. substantial parts of the digital systems can be replicated. Our knowledge of digital systems allows us to safely assume that a majority of the digital design will comprise circuits that fall into one or more of the following categories :

- Chain of similar gates (NAND/NOR chains) ;
- Chain of different gates ;
- Convergent paths ;
- Divergent paths.

Certainly, all these categories may have sub-categories, but in principle these categories cover the majority of digital combinational structures. Most complex digital systems are composed of combinations of one or more of these categories. To abstract the results provided in this paper, characterization libraries that can be developed for each of these categories should include the following information :

- SET Pulse propagation characteristics variation due to all input patterns and all possible initial SET pulse polarity.
- The Best Propagation Paths (BPP) and the Worst Propagation Paths (WPP) are identified based on the SET Pulse propagation characteristics variation.

Table 4.9 Abstraction of the SET Byzantine Pulse Re-converging Propagation Scenario for 2-Input TSPC OR Gate.

X1	**X2**	**Out**
0	0	0
0	1	1
1	0	1
1	1	1
0	υ	υ
1	υ	1
υ	1	1
υ	0	υ
υ	υ	υ

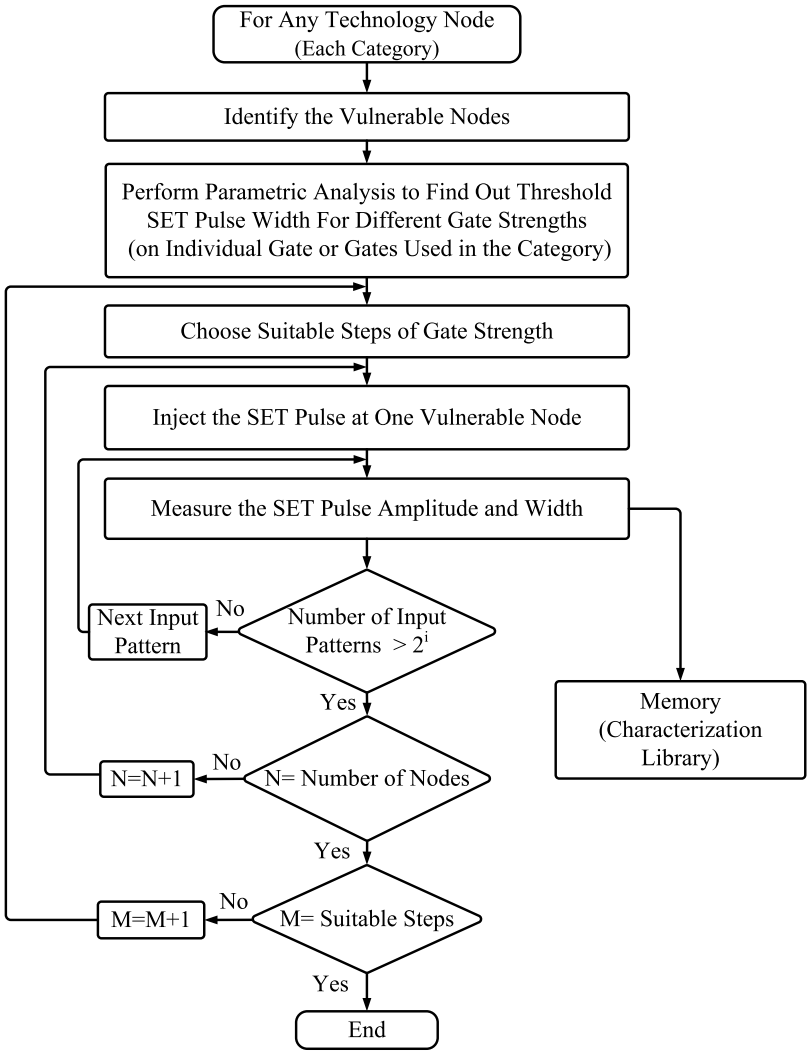


Figure 4.12 General Steps of a Possible Automated SET Pulse Propagation Analysis.

Once characterization libraries of SET pulse propagation are made, then one can package them into a Verilog netlist, which can become part of a standard cell library. This library can be invoked to estimate the possible BPP, and WPP for the SET pulse propagation, which will assist in developing soft error tolerant designs.
In order to attain the full benefit of this work, it is desirable to propagate the observed behaviors from our paper at higher abstraction level. A flow chart of a methodical approach to perform this analysis is depicted in Fig. 4.12, which can be implemented using CAD tools or scripting languages. As depicted in Fig. 4.12, this methodology allows building characterization libraries for each relevant category listed before as following :

— For each circuit category (chain of similar gates, chain of different gates, convergent paths, and divergent paths) the vulnerable nodes are identified.
— The threshold amplitude and duration of the SET pulse propagation is to be determined. In order to find out whether a particular SET pulse would propagate to the output under different constraints, we need to inject SET pulses into circuit level models. SET pulse injection at circuit level is well known to the community [92, 93, 94]. This characterization can be first made on the individual gates ; with parametric analysis to find out the threshold SET pulse width for different gate strengths.
— Next, representative circuits for each of these categories can be characterized for the effects of input patterns and propagation paths against the depth for SET propagation. In the process WPP and BPP can also be identified.

Similar to the automation methodology presented herein, a methodology to characterize thoroughly the defective behavior of transistors level circuit representation of a rich CML digital library using Hspice has been proposed in [95]. Two CAD tools were developed to automate this methodology. The First was an Inductive Fault Analysis (IFA) tool and the second was an Automated Fault Characterization Tool (AFCT). Automating the abstraction process would require similar tools that could be developed as further research.

4.5 Conclusion

In this paper, we investigated the variations of SET pulse characteristics while propagating in both static and TSPC logic. Our analysis of static logic has addressed the impact of propagation paths, input patterns, and polarity of SET pulses (positive or negative) on the SET PIPB phenomenon. We demonstrated that these factors aggravate the SET pulse broadening phenomenon. For example, in one of our simulations, a 200% broadening of the pulse width was observed due to re-converging paths. Moreover, a new analysis of electrical masking of SET propagation was presented. Worst and best propagation paths (WPP and

BPP) were identified for the analyzed designs.

The reported analysis of TSPC logic has addressed the impact of the propagation paths, diverging paths, fan-out, and re-converging paths on the SET pulse amplitude and width. Moreover, timing constraints related to the SET propagation such as timing of the strike and clock period are identified.

Finally, we have demonstrated that propagating a SET pulse through a diverging node may lead to a Byzantine faults. We have proposed a way to abstract all possible interpretations of the SET pulse at diverging nodes.

CHAPTER 5 ARTICLE 2 : MODELING, ANALYZING, AND ABSTRACTING SINGLE EVENT TRANSIENT PROPAGATION AT GATE LEVEL

Summary of the Chapter

In this chapter, our first attempt to bridge the gap between the analysis of SETs at transistor and gate levels is explained. Initially, a new abstraction of the results which were reported in Chapter 4 is introduced. This abstraction was first introduced in a paper entitled "Abstracting Single Event Transient Propagation Characteristics to Support Gate Level Modeling" which was subsequently published in the IEEE International Symposium on Circuits and Systems (ISCAS) on 2014. Thereafter, this abstraction was utilized to accurately model and analyze SET propagation at gate level using multiway decision graphs. The proposed modeling and the results this gate level analysis on different combinational designs was reported in a paper which was published in the IEEE International Midwest Symposium on Circuits and Systems (MWSCAS) on 2014. ***This published paper is reproduced in this chapter.***

Title : Modeling, Analyzing, and Abstracting Single Event Transient Propagation at Gate Level

Authors—Ghaith Bany Hamad, Syed Rafay Hasan, Otmane Ait Mohamed, and Yvon Savaria

Abstract—Soft errors have become one of the most challenging issues that impact the reliability of modern microelectronic systems at terrestrial altitudes. A new methodology to abstract, model, and analyze Single Event Transient (SET) propagation at different abstraction levels (transistor and gate level) is proposed. Transistor level characterization libraries are developed to abstract the impact of input patterns, pulse polarity, and propagation paths characteristics on the SET duration. Thereafter, these libraries are utilized to analyze SET pulse propagation at gate level using MDG model checker. We have implemented the proposed method on different ISCAS85 benchmark combinational circuits. Proposed methodology is orders of magnitude faster circuit level simulations. Moreover, we have developed gate level characterization libraries to abstract SET pulse propagation behavior at the gate level.

Index Terms—Soft Errors, SET, MDG, multiway decision graphs, delay degradation model,

PIPB, Byzantine Faults.

5.1 Introduction

Single Event Transients (SETs) are becoming a major source of errors in digital designs [11]. The SET propensity for propagation is enhanced as the technology size scales down. Moreover, the growing speed and complexity in new generation circuits increased the probability of SET to be captured as errors [11]. As a result, over the past two decades, several analysis techniques of SET pulse propagation operating at different abstraction levels have been proposed.

At transistor level, circuit simulation and experimental analysis have been performed. For instance, SET pulse width broadening and attenuation while propagating have been investigated [96, 3, 12]. Some other studies investigated the effects of fan-out [20] and the impacts of input pattern and logic structure [96] on SET pulse width. The analysis of SET pulse propagation at transistor level consumes large amount of time and requires full details of the design structure and the SET pulse characteristics. Hence, it has become very important to analyze SET pulse propagation at high abstraction levels.

Several research activities have been conducted recently in order to develop new methodologies to analyze SET pulse propagation at the gate level, such as the Fault injection based technique proposed in [43]. Some research groups have addressed this issue using formal methods such as ; Binary Decision Diagram (BDD)-based technique [48], a combination of Reduced-Ordered Binary Decision Diagrams (ROBDDs) and Algebraic Decision Diagrams (ADDs) [71], and a Boolean Satisfiability solver [6] (SAT-solver).

One of the main challenges in analyzing SET pulse propagation at higher abstraction level is to accurately model all the SET pulse propagation scenarios observed at the transistor level. For example, contemporary techniques (such as [48], [6]) are not sufficiently accurate, as these techniques omit the possibility of the SET pulse broadening while propagating. Moreover, state-of-the-art techniques at gate or higher abstraction levels analyze the susceptibility of digital circuits to soft error by only modeling the masking effects that can prevent SET pulses from propagating [11]. Nonetheless, existing state-of-art techniques are unable to model the effects of propagation paths characteristics, re-converging paths, and input patterns on SET pulse characteristics at high abstraction levels.

In order to overcome these shortcomings, in this paper, a new methodology to abstract, model, and analyze SET pulse propagation at gate level is proposed ; this work is original in the following ways. 1- We propose new characterization libraries of SET pulse propagation which

have two main advantages : a) the proposed libraries provide a comprehensive abstraction of several previously unconsidered SET pulse propagation scenarios ; b) these libraries also characterize the impacts of the logic structure and the input pattern. 2- For the first time, such libraries are utilized to accurately model and analyze SET pulse propagation at gate level using Multiway Decision Graphs (MDGs). Moreover, based on this analysis, we develop gate level characterization libraries to accurately abstract the observed SET pulse propagation. 3- We unravel scenarios through which Byzantine faults (defined later on) can occur due to SET pulse propagation through diverging paths.

The rest of this paper is organized as follows. Section 5.2 provides evidence of the problems we are addressing. Section 5.3 explains our proposed multi-level analysis of SET pulse propagation. The proposed abstraction of SET pulse propagation behavior at transistor level is explained in Section 5.4. Our proposed gate level analysis of SET pulse propagation is explained in Section 5.5. Section 8.8 concludes this work.

5.2 Problem Formulation

Based on transistor level analysis of the SET pulse propagation [96], [12], [20], [3] the following propagation scenarios can be observed : a) SET pulses can be logically masked by a gate if one of its inputs is set at a controlling logic value (e. g., '0' for a NAND gate) ; b) SET pulse width can be attenuated (electrically masked) [3] or broadened [96], [12] while propagating ; c) SET pulses can be also masked if their arrival time is outside the latching window of sequential elements.

In parallel with the analysis at the transistor level, different approaches to analyze SET pulse propagation at gate and higher abstraction levels have been proposed (such as [48], [6]). However, state-of-the-art techniques, which operates at high abstraction levels, suffer from the following shortcomings : 1) They omit the possibility of SET pulse broadening, which is significant because a relatively short pulse, just sufficient to propagate, can become arbitrarily long, provided the existence of a sufficient logic depth ; 2) Simulation based approaches (such as [43]) have serious shortcomings as they can be very time consuming and memory intensive for large designs, and the accuracy of fault simulation decreases with the decrease in the ratio of the simulated sample size over the total vector space size ; 3) At gate and higher abstraction levels, contemporary models do not include circuit level details and the impact of the logic structure and the input pattern on the SET pulse characteristics. Deficiencies in conventional models lead to inaccurate estimation of soft error rate (SER). Hence, there is a growing need to better abstract and characterize SET pulse propagation at gate and higher abstraction levels.

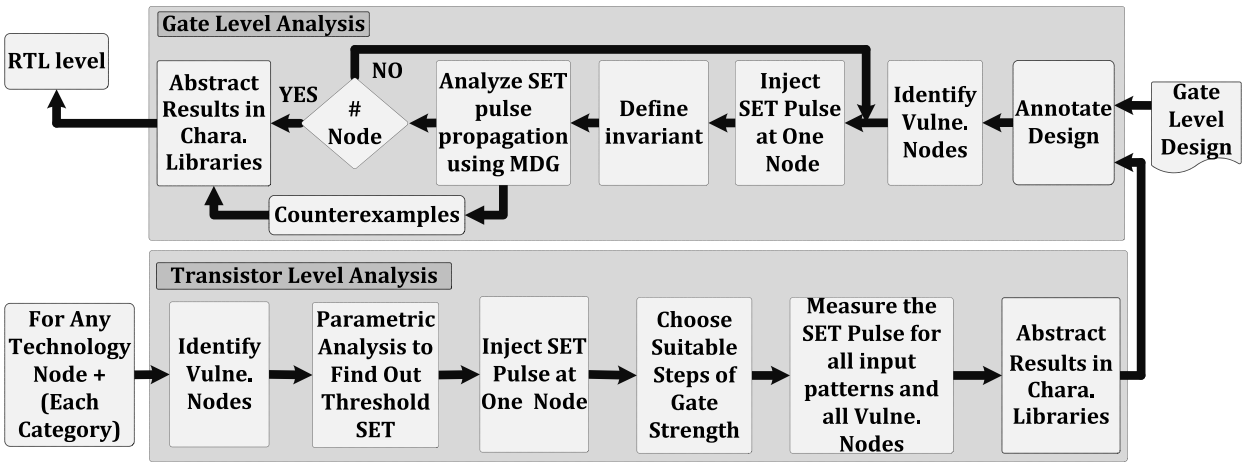


Figure 5.1 General Steps of our Proposed Methodology of SET Pulse Propagation Analysis at Transistor and Gate Level.

5.3 Proposed Multi-level SET Pulse Propagation Analysis

Our proposed general methodology, which includes the analysis of SET pulse propagation at both transistor level and gate level, is depicted in Fig. 5.1. Our analysis of SET pulse propagation at the transistor level was partly reported in [96]. In Section 5.4, essential characteristics of SET pulse propagation are abstracted, which allows propagating the observed behaviors to the gate level.

Our gate level analysis has the following steps, as shown in Fig. 5.1 : 1- Modeling SET pulse propagation at gate level by utilizing transistor level characterization libraries ; 2- Identifying the vulnerable nodes in a design and injecting SET pulse at one of these nodes. 3- Analysis of SET pulse propagation for each injection scenario, with the second and the third step being repeated for all vulnerable nodes in the design. 4- Finally, the results of our analysis of the SET pulse propagation behavior at gate level are abstracted as gate level characterization libraries. In section 5.5, we explain in details our SET pulse propagation analysis method based on MDG.

5.4 Proposed Abstraction of SET Pulse Propagation Based on Characterization Libraries

This section introduces a new approach to abstract SET pulse propagation scenarios observed at transistor level. To obtain this abstraction, we developed characterization libraries to characterize SET pulse propagation as a function of input patterns, pulse polarity, and

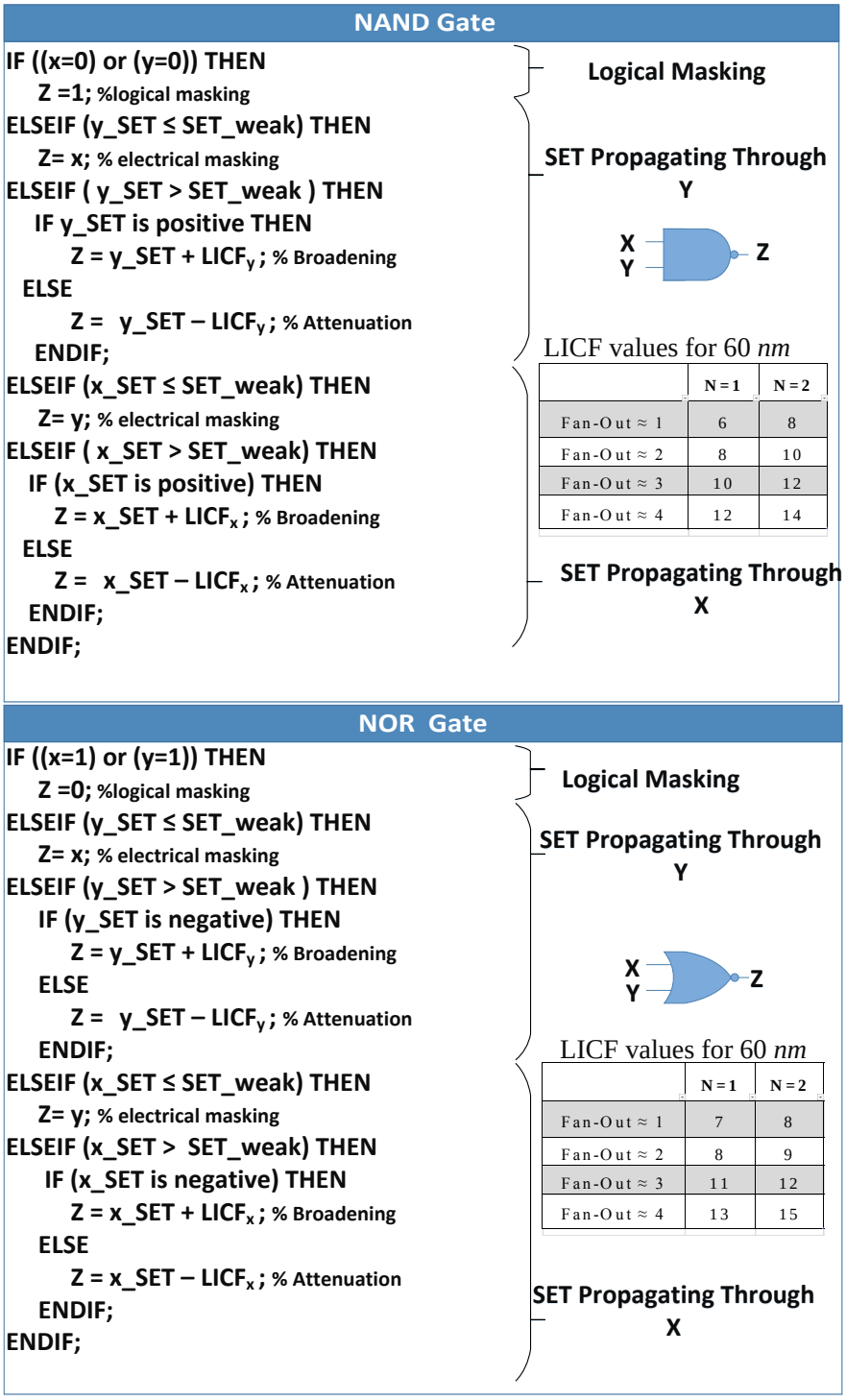


Figure 5.2 Characterization Libraries of Both NAND and NOR Gates.

fan-out. In this work, if the SET pulse width is less than the propagation threshold of the subsequent gate then this width is called *SET_weak*. The characterization libraries of the NAND and the NOR gates are depicted in Fig. 5.2. To illustrate how these libraries are

used, explanations on the NAND gate library use are provided in the sequel. The library element which reflects logical masking expressed as (*IF ((X=0) or (Y = 0)) then (Z = 1)*). Moreover, as shown in Fig. 5.2, if we consider SET pulse propagation through node Y, then the SET pulse width variation is modeled based on its strength according to the following scenarios : a) If ($Y\_SET \leq SET\_weak$), then it will be electrically masked ; b) if ($Y\_SET > SET\_weak$), then the width of the SET pulse at the output depends on its polarity. Positive SET pulse broadens while propagating, while negative SET pulse attenuates. Moreover, SET pulse width variation due to input pattern and load is abstracted using the Load Input Combination Factor (LICF), proposed in [97], as shown in Fig. 5.2.

The second detailed analysis consisted of abstracting the SET pulses re-converging scenarios. The SET pulse width variation while propagating through a re-convergent gate, which is abstracted in the characterization library shown in Fig. 5.3 two factors : a) the difference between the re-converging paths propagation delays ($|D1 - D2|$) ; b) the difference between the duration of the SET pulses ($|D_{SET1} - D_{SET2}|$). There are two main scenarios that influence converging SET pulses :

1- The first scenario occurs if $|D1 - D2|$ is less than the duration of the longest pulse ($MAX(D_{SET1}, D_{SET2})$). The amount of broadening in this scenario is divided into two sub-

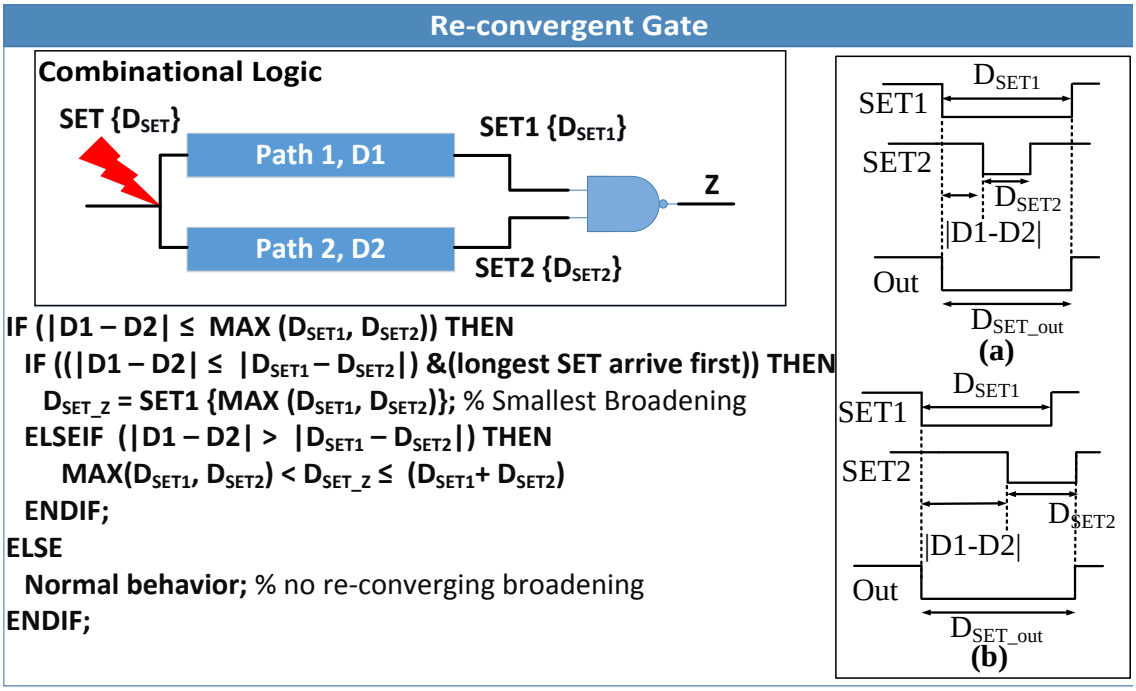
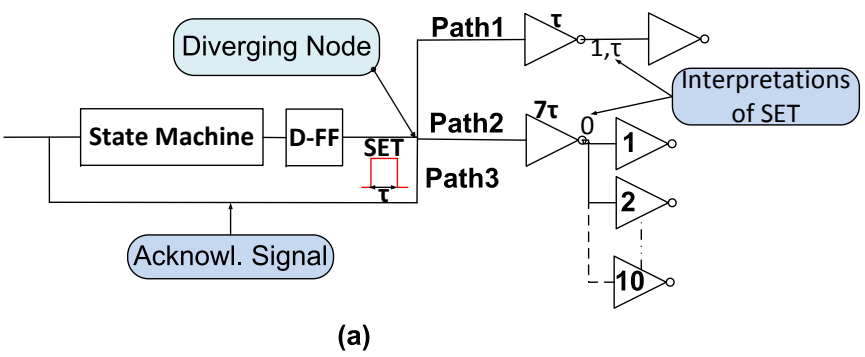


Figure 5.3 Characterization Library Modeling of a Re-convergent Gate.



SET Pulse	Path1	Path2	Path3
τ	0	0	0
τ	0	0	1
τ	0	1	0
τ	0	1	1
τ	1	0	0
τ	1	0	1
τ	1	1	0
τ	1	1	1

(b)

Figure 5.4 (a) SET Pulse Propagation Induced Byzantine Fault in Static Logic. (b) Abstraction of SET Pulse Propagation Induced Byzantine Fault Scenarios.

scenarios. a) If $|D1 - D2| < |D_{SET1} - D_{SET2}|$ and the longest pulse arrives first, then no broadening occurs as shown in Fig. 5.3a. The duration of the SET pulse at the output is nominally the same as the duration of the longest pulse. b) If $|D1 - D2| > |D_{SET1} - D_{SET2}|$, then the SET pulse duration at the output depends on the overlap between the converging pulses which is $(MAX(D_{SET1}, D_{SET2}) < SET\_pulse < (D_{SET1} + D_{SET2}))$ as shown in Fig. 5.3b.

2- The second scenario that occurs if $(|D1 - D2| > MAX(D_{SET1}, D_{SET2}))$, in which case the converging pulses will not overlap (no re-converging broadening). Therefore, the gate propagates one SET pulse at a time (similar to Fig. 5.2).

Our third detailed analysis relates to means of abstracting SET pulse (with duration τ)

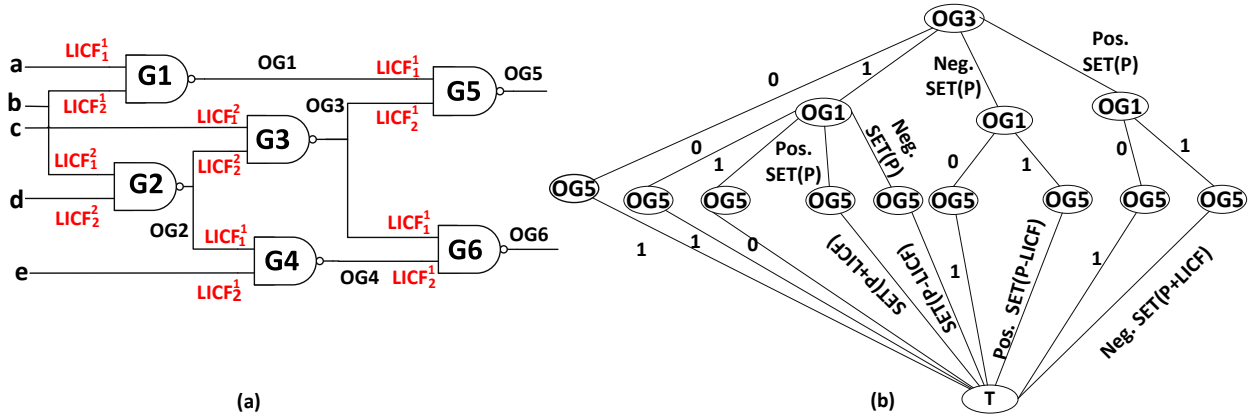


Figure 5.5 (a) The Annotated C17 Design With the LICF Values, (b) Multiway Decision Graph (MDG) for G5 From the C17 Design.

propagation through diverging paths as shown in Fig. 5.4. Different interpretations ('0' or '1') are observed for different diverging paths depending on their thresholds. This phenomenon is known in the literature as the Byzantine fault, which is defined as a fault presenting different symptoms or interpretations to different observers [89]. These different interpretations lead to different faults in different paths, which can have a large impact on the design behavior. For example, the state machine in Fig. 5.4 confirms its action by the acknowledgment signal. However, due to the Byzantine fault the SET pulse at the diverging node can be interpreted as '0' in *path3* and as '1' in *path1*. In such case, the state machine acts as if the action is not performed while the outside world thinks that it has been performed.

The number of the SET pulse interpretations at the diverging node is equal to 2^z, where z is the number of the diverging paths, e.g. Fig. 5.4b depicts all possible interpretations of the SET pulse when *z=3*. However, the knowledge of the diverging paths thresholds and the SET pulse width, can help eliminating some of the cases in Fig. 5.4b. For example, when SET pulse duration is τ and *path2* threshold is 7τ, then all the cases in Fig. 5.4 when *path2* interpretation is '1' are not possible.

Table 5.1 The Characterization Library of the C17

Node	**Pulse Polarity**	**Injected SET**	**Output SET**	**CIC**	PIW
a	Positive	SET(3)	SET(3)	(b=1, c=0)/ (b=1, d=1)	2
	Negative	SET(3)	SET(3)	(b=1, c=0)/(b=1, d=1)	
b	Positive	SET(3)	SET(3)/SET(4)	(a=1, c=0)/(b=1, c=1, a=0)	4
	Negative	SET(3)	SET(3)/SET(2)	(a=1, c=0)/(b=1, c=1, a=0)	
c	Positive	SET(3)	SET(3)	(b=0)/(d=0, b=0)	3
	Negative	SET(3)	SET(3)	(b=0)/(d=0, b=0)	
d	Positive	SET(3)	SET(4)	(b=1, c=1, a=0)	2
	Negative	SET(3)	SET(2)	(b=1, c=1, a=0)	
e	Positive	SET(3)	SET(3)	(d=0, c=0)/(b=0, c=0)	0
	Negative	SET(3)	SET(3)	(d=0, c=0)/(b=0, c=0)	
OG1	Positive	SET(3)	SET(4)	(c=0)/ (b=1, d=1)	-
	Negative	SET(3)	SET(2)	(c=0)/ (b=1, d=1)	
OG2	Positive	SET(3)	SET(3)	(c=1, a=0)/ (c=1, b=0)	-
	Negative	SET(3)	SET(3)	(c=1, a=0)/ (c=1, b=0)	
OG3	Positive	SET(3)	SET(4)	(a=0)/(b=0)	-
	Negative	SET(3)	SET(2)	(a=0)/(b=0)	
OG4	Positive	SET(3)	SET(4)	(c=0)/(d=1, b=1)	-
	Negative	SET(3)	SET(2)	(c=0)/(d=1, b=1)	

5.5 Gate Level Analysis of SET Pulse Propagation

In this section, the SET pulse propagation is analyzed at the gate level by utilizing the transistor level characterization libraries. Moreover, our gate level analysis is performed using the MDG, which is a tool set used for the formal verification of complex digital systems [81]. It includes application procedures for equivalence checking, model checking, and invariant checking [81].

First, the SET pulse categories defined in Section 5.4 are termed into two classes : a) *SET(1)* which corresponds to the *SET_weak* ; b) *SET(P)* which corresponds to the case when SET pulse width is larger than *SET_weak*, P value corresponds to the SET pulse strength. Therefore, the attenuation and the broadening of this pulse are modeled by changing the value of P. As an example, if a positive *SET(P)* pulse propagates through a NAND gate then at the output it is termed as *SET(P+LICF)*. Next, our MDG based technique is applied which has the following steps as shown in Fig. 5.1 :

5.5.1 Design Annotation and SET Pulse Injection

Our methodology start by annotating each node in the design with its corresponding $LICF$ value. In Fig. 5.5, this value is termed as $LICF_N^F$, N is the node where the SET propagates, F is the gate fan-out. These values are depicted in Fig. 5.2, which are characterized beforehand using electrical simulation with 65 nm CMOS technology library from TSMC. Next, the SET pulse is injected using the fault injection element (FIE), which select between the error free mode and the faulty mode during verification. Thereafter, the MDG tool builds graph representation for each gate based on its characterization library. As an example, Fig. 5.5b depicts the MDG of G5 in the circuit shown in Fig. 5.5a. This MDG graph implements the NAND gate characterization library depicted in Fig. 5.2. *G5* is a two input NAND gate ; the

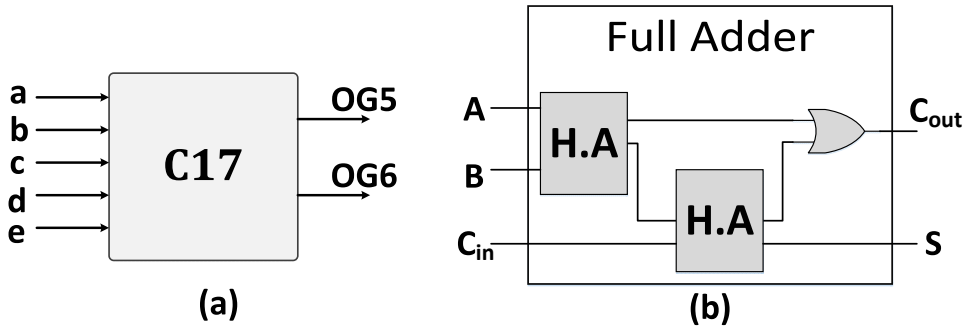


Figure 5.6 The Use of the Characterization Library at the RTL.

output of *G1* (*OG1*) and the output of *G3* (*OG3*). It is shown that, when *OG3* is '0', then *OG5* is '1' irrespective of *OG1* (logical masking). But if *OG3* is '1' then *OG5* is dependent on *OG1*. In the last case, SET pulse at *OG1* can broaden (*OG5* is *SET(P+LICF)*) or attenuate (*OG5* is *SET(P-LICF)*) while propagating.

Table 5.2 Analyzed Benchmark Circuits

Circuit Name	Circuit Function	Total Gates	Input Lines	Output Lines
74182	CLA	19	9	4
74283	Fast Adder	36	9	5
C432	Priority Decoder	160 (18 EXOR)	36	7
C499	ECAT	202 (104 EXOR)	41	32
C880	ALU and Control	383	60	26

5.5.2 Gate Level Analysis and Results Abstraction

Our proposed methodology analyzes SET pulse propagation using the invariant checking tool from the MDG tool set. This tool generates counterexamples if the SET pulse can propagate from the vulnerable node (where SET is injected) to the primary output as shown in Fig. 5.1. This analysis is performed for all vulnerable nodes in the design.

Our proposed methodology abstracts the SET pulse propagation behavior at the gate level by developing gate level characterization libraries. Tables 5.1 is the characterization library of the design shown in Fig. 5.5a. The following information related to the SET pulse propagation can be observed from such library. 1- The Critical Input Combinations (CICs) that allow the SET pulse to propagate to the output. 2- The SET pulse width variation behavior (attenuation or broadening). 3- It reports the Primary Input Weight (PIW) which is a measure of the impact of each input node on the SET pulse propagation. For example, the value of node *b* is related to four SET pulse propagation scenarios. Moreover, these libraries can be utilized at the RTL level. Simple example is shown in Fig. 5.6b, where using the half adder (HA) characterization library, the full adder (FA) can be analyzed without the need of its gates level structure. We have performed our analysis on several designs such as the benchmark designs listed in Table 5.2.

Fig. 5.7 depicts the predicted SET pulse width variation while propagating through the design in Fig. 5.5a by applying our proposed model and the delay degradation model [3]. In this analysis, a SET(2) pulse (P=2) is injected at each node in Fig. 5.5a. According to the delay degradation model this pulse is electrically masked if injected at node *a*, *b*, *c*, *d*, *e*, and *OG2*,

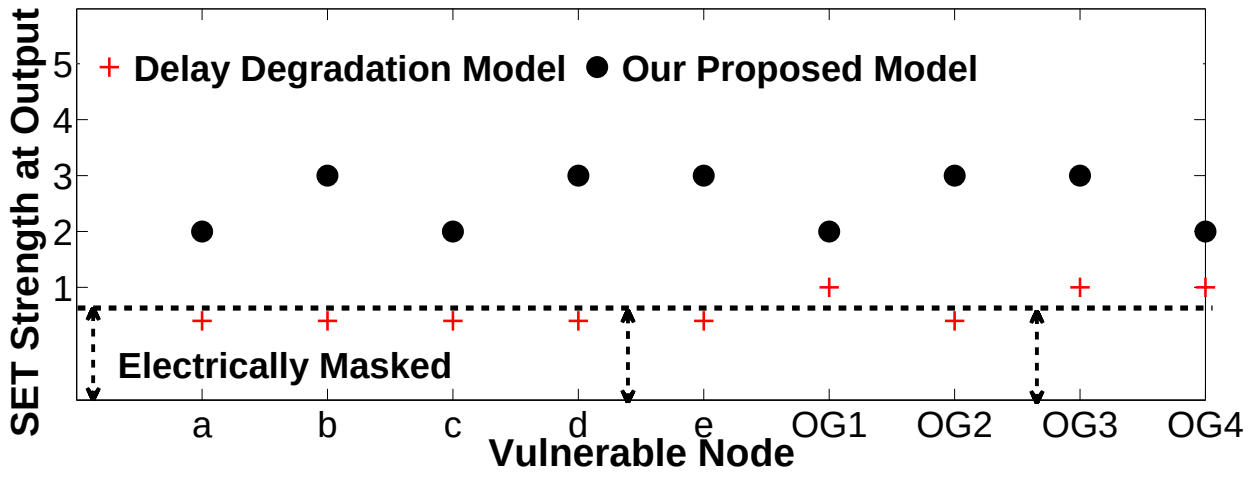


Figure 5.7 The SET Pulse Propagation Based on the Delay Degradation Model (DDM) [3] Versus our Proposed Model.

as depicted in Fig. 5.7. However, our proposed model demonstrates that this pulse may reach the output with sufficient strength for all these nodes. We can conclude that state-of-the-art SET modeling miss some propagation scenarios if the broadening is not considered along with the masking effects.

5.6 Conclusion

In this paper, a new approach to better abstract SET pulse propagation scenarios and to characterize the impact of the pulse polarity, the logic structure, and the input patterns on the propagating SET pulse width is proposed. Moreover, we have utilized transistor level characterization libraries to model and analyze SET pulse propagation at the gate level using the MDG tool. Finally, we proposed new gate level characterization libraries which can be used to accurately analyze SET pulse propagation and estimate the SER at the RTL level.

5.7 Acknowledgments

This research is part of the AVIO-403 project supported by the Consortium for Research and Innovation in Aerospace in Quebec (CRIAQ). Authors would like to thank the industrial partners, namely Bombardier Aerospace, MDA Space Missions and Canadian Space Agency for their support.

CHAPTER 6 ARTICLE 3 : EFFICIENT AND ACCURATE ANALYSIS OF SINGLE EVENT TRANSIENTS PROPAGATION USING SMT-BASED TECHNIQUES

Summary of the Chapter

In this chapter, a new methodology to bridge the gap between SET transistor and gate levels analysis and improve the scalability of gate level analysis is introduced. In this chapter, we explain in details the proposed modeling of SETs propagation as a satisfiability problem by utilizing the efficiency of the Satisfiability Modulo Theories (SMTs) and the underlying details extracted from the layout and characterized from technology node (reported in Chapter 4). The proposed methodology and the results of its implementation on different designs was first introduced (and subsequently published) in a major conference paper in the IEEE International Conference on Computer-Aided Design (ICCAD) on 2016. ***This published paper is reproduced in this chapter.***

Title : Efficient and Accurate Analysis of Single Event Transients Propagation Using SMT-Based Techniques

Authors—Ghaith Bany Hamad, Ghaith Kazma, Otmane Ait Mohamed, and Yvon Savaria

Abstract—This paper presents a hierarchical framework to model, analyze, and estimate digital design vulnerability to soft errors due to Single Event Transients (SETs). A new SET propagation model is proposed. This model simultaneously includes the impact of masking effects, width variation, and re-converging paths by utilizing satisfiability modulo theories. Furthermore, new metrics characterizing the soft error rate of a given design are proposed. Reported results show that the proposed methodology significantly enhances the efficiency of SET analysis in terms of : 1) *accuracy* as it gives accurate estimates of SET sensitivity based on gates timing extracted from layout. These results provide new insights to combinational designs vulnerability to SETs ; 2) *speed* as it is orders of magnitude faster than contemporary techniques ; 3) *scalability* as it can handle large and complex designs such as 128-bit multipliers, whereas contemporary techniques are unable to handle multipliers larger than 32 bits.

Index Terms—Soft Errors, SETs, gate level, layout, technology characterization, Satisfiability Modulo Theories, PIPB, Yices solver, multipliers, making effects, re-converging paths.

6.1 Introduction

Soft errors due to Single Event Transients (SETs) have now become one of the most challenging types of uncertainties that impact the reliability of modern electronic systems. For instance, they were responsible for the catastrophic failure and the recall of many safety critical systems, such as implantable cardiac pacemakers [7]. Therefore, there is a growing need to analyze and estimate the impact of soft errors on today's complex digital designs to be able to develop efficient fault tolerance techniques.

The propagation of SET through combinational designs is affected by three masking effects ; logical, electrical, and temporal. An SET is logically masked by a gate if, while propagating through one or more inputs, at least one of the other inputs has a controlling logic value (e.g., '0' for a NAND gate). Electrical masking occurs when the duration (i.e., width) of an SET is less than the threshold of the subsequent logic gates. An SET is masked due to temporal masking if it arrives outside the latching window of registers. Furthermore, if an SET is not logically and electrically masked, it may be subject to attenuation or broadening as it propagates [12]. Recent radiation testing and circuit simulation experiments have demonstrated that the broadening phenomenon has a high impact on the Soft Error Rate (SER) of a circuit [20], [12].

One of the most challenging issues in evaluating design vulnerability to SETs is to accurately model SET propagation. Different analysis methodologies that operate at different levels of abstraction, both formal and simulation-based, have been proposed.

At circuit level, parameters extraction and detailed simulations can provide a certain level of accuracy for phenomena such as electrical masking and SET width variation. However, this analysis is very computationally intensive and would be intractable at the chip level and is only tractable at the cell level. In other words, this type of analysis could be conducted on hundreds of transistors at most.

Performing such analysis at the gate level comes at the cost of less accuracy, since gates loading and timing details are abstracted. These details, which are critical for modeling electrical and temporal masking, are only available at the post-layout stage. Previous studies at gate level can be categorized into three groups :
1) *Simulation based techniques* [74, 98] : In these techniques, SET propagation has to be analyzed over all possible input vectors, for all possible SET widths, and both polarities. Obviously, complete exhaustive analysis of propagation possibilities in complex systems is intractable at the logic level using simulations. Consequently, such techniques have their limits in accuracy. Generally, their accuracy is determined by the ratio of the simulated sample size

over the total vector space size.

2) *Numerical based techniques* ([50], [51], [52] [41]) : Each of these techniques try to estimate the impact of masking effects on the SER. Electrical masking is presented in [50], temporal masking is analyzed in [41], and a model combining all masking effects is presented in [51]. However, these techniques are not scalable and their models do not include the impact of SET broadening and SET re-converging.

3) *Formal based techniques* [49, 56, 5] : Model checking [5] and equivalence checking [49, 56] based techniques have been developed to improve the coverage of SET analysis. However, these techniques restrict their model to Boolean representation [49, 56] or some enumerated data type [5], which greatly limits the modeling of variations in SET characteristics, such as width variations and timing constraints. Moreover, most of the existing techniques generate two formal models of the design ; a golden and a faulty model. Then, similar to fault simulation, the states of the outputs of both models are compared thus such techniques double the resource requirements. Another issue with existing techniques is that they map each input vector to a unique state. Thus, the corresponding model has at least 2×2^M states (where M is the number of primary inputs). With such techniques, any formal tool rapidly runs out of memory, even when modeling small desings at RTL levels e.g., a 14-bit adder [56]. Additionally, the size of the formal model grows exponentially with the size of some types of arithmetic circuits such as multipliers. This is mostly attributed to the intrinsically complex structure of such circuits i.e., large number of re-converging paths.

In summary, the important question on *"how to measure the vulnerability of complex designs at gate level without losing the accuracy provided from circuit level analysis ?"* is not appropriately addressed in the literature so far. To answer this question, we introduce a novel methodology to estimate the vulnerability of combinational designs to soft errors. This methodology starts with the synthesis of an RTL design into its gate level representation and then the layout of the design is extracted. Next, gates parasitics are extracted and gates timing details are characterized from the layout. These parameters are then employed to model and analyze SET propagation. A new model for SETs propagation is proposed, which captures the variations in the SET characteristics while propagating, such as the SET width attenuation and broadening. Moreover, this model includes the impact of all masking effects (logical, electrical, and temporal) and re-converging paths on SET propagation.

Furthermore, a new formalism modeling SET propagation into a Satisfiability problem utilizing Satisfiability Modulo Theories (SMTs) is proposed. The proposed methodology provides an exhaustive analysis of SETs propagation from each vulnerable node to each output using efficient well-known SMT solvers. It generates the required conditions for SET propagation

which are then used to compute the vulnerability of each node and to estimate the Soft Error Rate (SER) of the design. To the best of the authors' knowledge, this is the first time a SMT based technique exploits the layout details to provide an accurate estimation of the SER at the gate level.

In comparison with [51], [41], [5], experimental results reported later show that the proposed methodology significantly reduces the resource requirements while improving the accuracy of the computed SERs. For instance, it can analyze complex arithmetic circuits such as a 128-bit multiplier in about 70 minutes, while existing techniques ([51], [41]) require 499 minutes to analyze a 24-bit multiplier and fail to handle 32 bit multipliers.

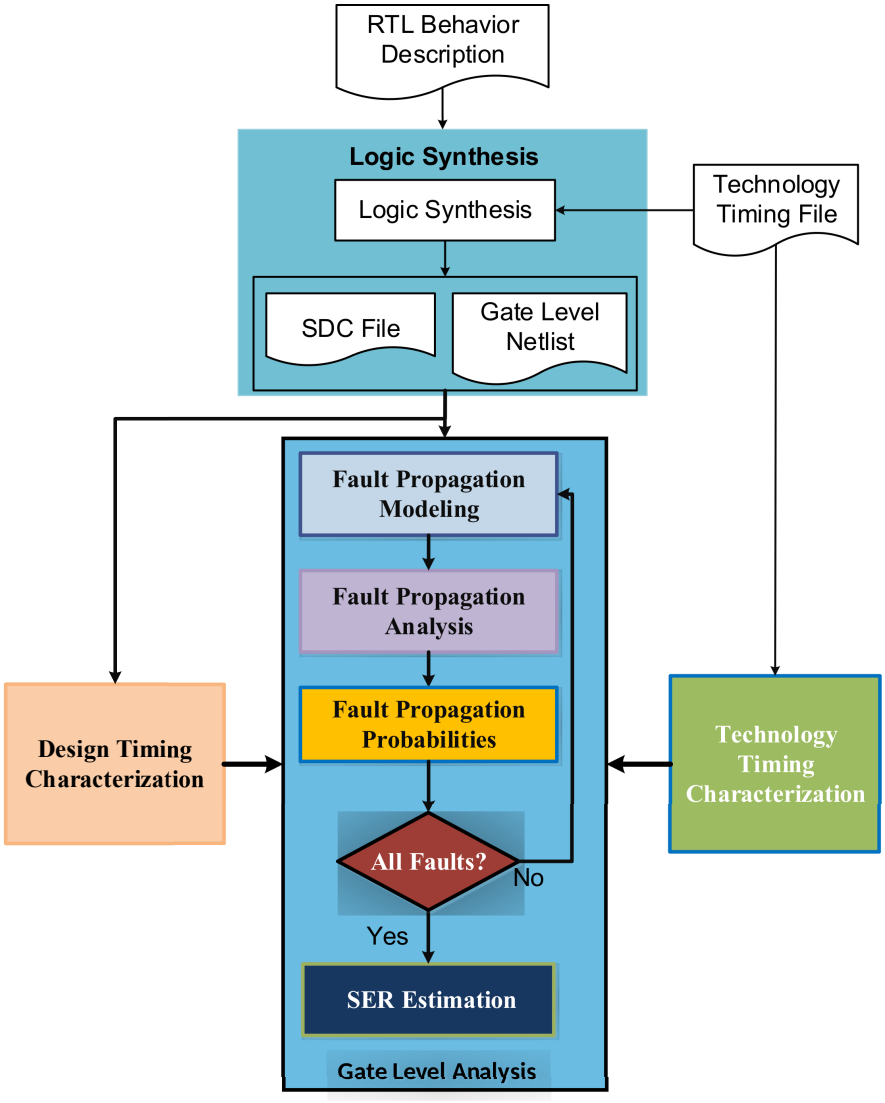


Figure 6.1 The Proposed Methodology for SET Modeling and Analysis.

6.2 Proposed Framework

An overview of the proposed methodology is shown in Fig. 6.1. It starts with the synthesis of the RTL design into its gate level representation using EDA synthesis tools. In this process, the design constraints are also generated. The gate level analysis starts by characterizing the timing details of each gate. This process consists on generating the complete mask layout of the intended design by following some well known VLSI design flow. Then, the exact parasitics of the layout are extracted, which are then employed to compute the exact timing for each gate using EDA static timing analysis tools. Moreover, transistor level analysis is performed to generate the Transistor Propagation Tables (TPTs). These tables report the technology parameters required to model SET propagation behavior for the technology node of interest. Thereafter, the extracted design timing, TPTs, and the gate level netlist are utilized in order to model SET propagation as an SMT problem. Next, the SET is injected (with different polarities) and its propagation is exhaustively analyzed, from each vulnerable node to each primary output. For each vulnerable node in the design, the proposed analysis generates : a) the set of input vectors that must be present at the primary inputs so that an injected SET is not logically masked ; b) the maximum SET strike time within the clock cycle so it is not temporally masked by the latching window of the register ; and c) the minimum SET width required at the strike time so it is not electrically masked. These results are then used to compute the vulnerability of each node and to estimate the SER of the design. In the following subsections, the main steps of the proposed methodology are explained in detail.

6.2.1 Design Timing Characterization

Without the layout, the modeling of electrical masking and SET characteristics variations cannot be fully accurate due to the lack of exact loading and exact timing details. This can lead to some SETs not being detected, thereby the calculated quantitative estimates are not accurate and they can serve only as approximations. Therefore, the proposed methodology involves generating and characterizing the layout of a design using EDA tools. The inputs of the post-layout characterization are (i) the gate level netlist generated from the synthesis tool, (ii) the target technology timing file (i.e., lib file), and (iii) the set of timing constraints used to drive the place and route process, which is reported as the Synopsys Design Constraints (SDC) file. Place-and-route tools create a layout by utilizing the layouts of the pre-defined standard cells such that the interconnections between the cells, as specified in the netlist, are preserved. Place-and-route tools also take into account the detailed timing issues that arise from the actual location of the various cells in the layout. In this work, the generation of the design layout and the extraction of its parasitics are done using the *SOC encounter*

tool from Cadence. It is preferable that the timing characterization step employs a highly accurate device-level simulator such as HSPICE [85] or some static timing software at the transistor level or gate level. In this work, timing characterization is performed at the gate level using the static timing analysis software from synopsys (i.e., *PrimeTime* [86]). To do that, (i) the gate netlist, (ii) the detailed layout parasitics extracted in the Standard Parasitic Exchange Format (SPEF) file, and (iii) the timing model lib file are required. The result of this process is the detailed layout timing of each gate, which is characterized into a Standard Delay Format (SDF) file.

6.2.2 Technology Node Characterization

SET propagation behavior varies based on the technology of interest. This variation is included in our model as a technology dependent parameter (a.k.a fitting paramter k) that impacts the SET width variation while propagating through each gate as proposed in [20]. This parameter is obtained through detailed transistor level analysis which investigates the impact of different design parameters on SET characteristics. The main steps of this analysis are summarized in Alg. 2. This analysis is performed using HSPICE [85] simulations of all standard cells and their combinations (chain of similar gates, chain of different gates, and diverging paths). This analysis starts by matching the transistors and finding good transistor sizes according to a suitable criteria. To characterize the impact of each design parameter, all others are fixed and the gate is simulated over a range of possible values of the target parameter. The variations in the SET characteristics due to variations in the design parameters (such as the node capacitance, the input pattern, and the fan-out) are characterized. For example, the impact of the node capacitance is characterized by performing different HSPICE simulations for a range of possible values of output capacitances and different fan-outs (*fan-out* of 1 to *fan-out* of 4) over a possible range of SET widths at the inputs. The results of all these analyses are stored in the Transistor-level Propagation Table (TPT) which contains the value of the fitting parameter (i.e., k) for each gate, for different loading and inputs. The transistor level analysis of standard cells is done beforehand and only once for each technology node.

6.2.3 Fault Propagation Modeling

The proposed modeling utilizes the characterized design layout timing details reported in the SDF file as explained in Section 6.2.1. Each gate is annotated with its propagation delay (i.e., tp) which is equal to tp_{LH} or tp_{HL} depending on the SET polarity. Each gate is also annotated with Δtp which is the difference between tp_{LH} and tp_{HL} or tp_{HL} and tp_{LH}

Algorithm 2 Transistor Level Timing Characterization

Inputs : Netlist.v, Tech.lib, LEF_lib.lef
Outputs : Transistor Propagation Table (TPT)
Tools : HSPICE Synopsys circuit simulator
procedure *TranTimingExtraction*
  Parameters ← ***Identify*** circuit parameters ;
  for each $p \in Paramters$;
    IdentifySize(PMOS(w,l), NMOS(w,l), C) ;
    Cir ←***Build*** example circuit ;
    Nodes ← ***Identify*** vulnerable nodes in *Cir* ;
    for each $n \in Nodes$;
      SET_in ←***InjectSET***(n, width, polarity) ;
      CircuitSimulation(Netlist, InPattern, SET_in) ;
      k ← ***CharactrizeResults***(SET_out, Tp) ;
      Update TPT ;

for each input to output transition depending on the SET polarity, as illustrated in Fig. 6.2. Moreover, the results of the transistor level characterization reported in the TPT are utilized to provide the technology dependent parameters (i.e., k), as explained in Section 6.2.2. The SET width propagation threshold for each gate (i.e., W_{thr}) is approximated as k, the technology dependent parameter, times tp, which is required to model electrical masking.

The proposed methodology models the SET propagation as a satisfiability problem based on the Satisfiability Modulo Theories (SMTs). In order for this model to fully capture the design functionality and the variation in SET characteristics, signals in the design possess four attributes : *logic* of *bit* type, *faulty* of *Boolean* type, *FaultWidth* of *real* type, and the SET arrival time (*FaultTime*) of *real* type, as follows :

$$Structure\ SET = [bit\ \textbf{logic}, bool\ \textbf{faulty}, real\ \textbf{FaultWidth}, real\ \textbf{FaultTime}]$$

The *logic* variable stores the original value of the signal. A signal is faulty if its *faulty* attribute is 'true'. The SET polarity is decided based on the *logic* attribute. A fault is considered to have a positive polarity if *faulty* is 'true' and *logic* is '0' and is considered to have a negative polarity if *faulty* is 'true' and *logic* is '1'. The fault time is taken with respect to the clock period. For example, at a certain node i, if there is a fault with a positive polarity injected at time x and with a width y, its signal would be described as follows :

$$Node\_i = SET(logic = 0, faulty = True, FaultTime = x, FaultWidth = y)$$

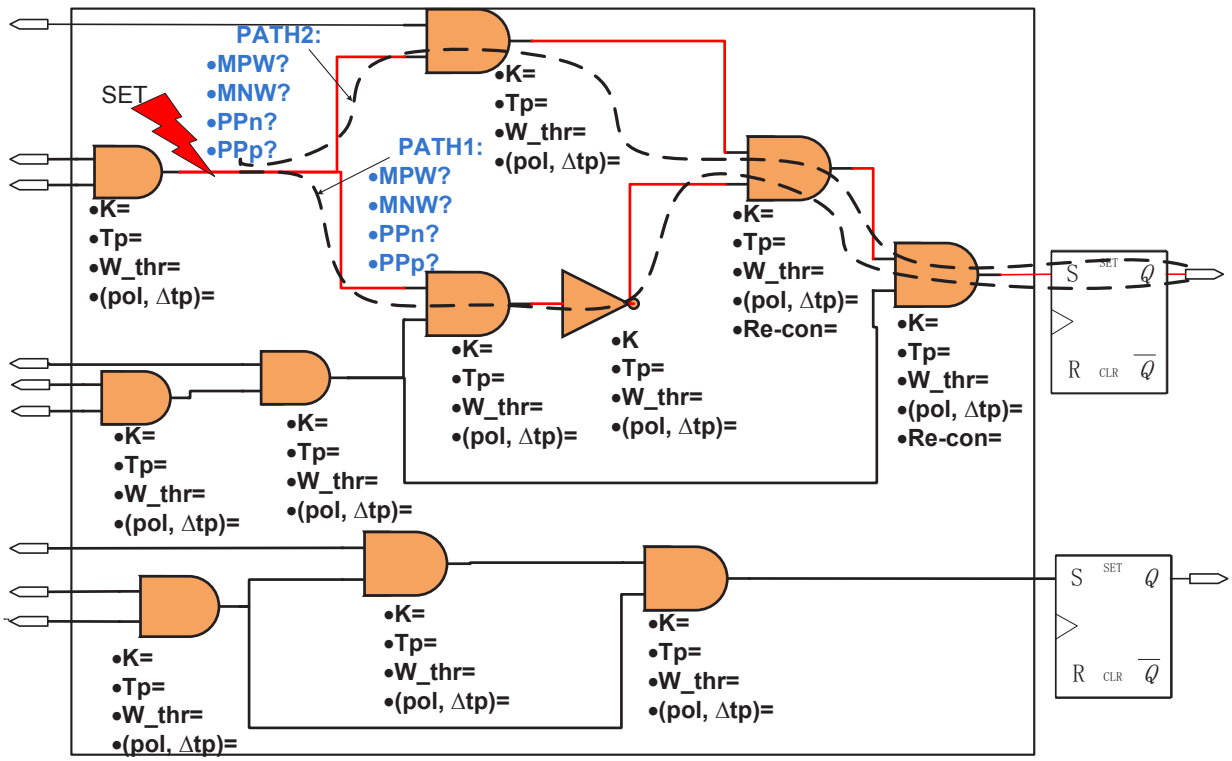


Figure 6.2 Modeling of Combinational Designs.

A library of the SMT models for all the standard logic gates was developed. These models include methods to evaluate the gate logic functionality and each masking effects. The effect of logical masking is modeled as a Boolean function over the *faulty* and the *logic* attributes of each input signal based on the gate functionality.

The possibility that an SET is electrically masked is modeled as a linear arithmetic constraints over the gate SET propagation threshold W_{thr}. The width of the SET at the input of the gate (τ_{set}) has to be greater than W_{thr} to cause an error at the output. In our model, if the SET is not logically and electrically masked, then its width can broaden or attenuate while propagating. In order to accurately model the variation in SET width while propagating, the models proposed in [20] and [12] are adapted. The variation in SET width is modeled based on τ_{set} and Δtp. It is important to note that Δtp is computed based on the input pattern i.e., each input pattern has its own delay and Δtp. For example, a NAND gate broadens the width of a positive SET by $k * \Delta tp$ and attenuates the width of a negative SET by $k * \Delta tp$ [12].

In the proposed temporal masking model, for an SET to be latched, it must arrive before the clock edge by the setup time of the register ($T_{clk} - T_s$). Moreover, an SET has to have a

steady state after the clock edge by the hold time of the register ($T_{clk} + T_h$), i.e., its width must be $\geq (T_h + T_s)$. To accurately model this, several factors are taken into consideration, such as the strike time (T_{st}), initial width (W_{in}), and the delay (tp) and width variation (Δtp) of all the gates in the SET propagation path. It is assumed that a register is connected to each primary output as shown in Fig. 6.2.

Contemporary studies are not accurately modeling the impact of re-converging paths. For instance, they assume that SETs never re-converge and always propagate separately [41], whereas in reality SETs at the inputs always logically mask each other in the overlap period for some gates [12]. In this work, we propose a new characterization of the different SETs re-converging scenarios based on their arrival times, widths, polarities, and the re-converging gate functionality. The proposed characterization for 2-input NAND/AND re-converging gates is summarized in Fig. 6.3. At the input of the re-converging gate we have : $D1$ and $D2$, which are the propagation delays of the re-converging paths, W_{SET1} and W_{SET2} which are the width of the two SETs. If there is an overlap between the two re-converging SETs $(((D1 \geq D2)\&(D2+W_{SET2} > D1))|((D2 \geq D1)\&(D1+W_{SET1} > D2)))$ then there are six different re-converging scenarios to be checked. In order to better explain the different scenarios for this re-convergence, Fig. 6.4 depicts an example of two paths re-converging through 2-input AND gate. Following, we explain each re-converging scenario for this example :

1. If both SETs have controlling values i.e., in the case of the AND gate, negative polarity SETs, the SET at the output is the result of the disjunction of both SETs $(min(D1, D2) < SET\_out < max(D1 + W_{SET1}, D2 + W_{SET2}))$, as shown in Fig. 6.4(a).
2. If both SETs have non controlling values i.e., in the case of the AND gate, positive polarity SETs, the SET at the output is the result of the conjunction of both SETs $(max(D1, D2) < SET\_out < min(D1 + W_{SET1}, D2 + W_{SET2}))$, as shown in Fig. 6.4(b).
3. If only one SET has controlling value and fully overlaps the other SET, then both SETs logically mask each other, as shown in Fig. 6.4(c).
4. If only one SET has controlling value and is fully overlapped by the other SET, the output SET is composed of the non-overlapping regions between the two SETs (see Fig. 6.4(d)). In this scenario, two shorter SETs are generated, i.e., SETs are attenuated due to re-converging paths by the overlap region between them.
5. If only one SET has controlling value and partially overlaps the other SET, then the width of the SET at the output equals the region where the non-controlling SET is propagating, as shown in Fig. 6.4(e), (f).

```
IF (((D1   D2)  &  (D2+W_SET2 > D1)) | ((D2   D1)  &  (D1+W_SET1 > D2))))THEN
  IF (SET1 & SET2 have controlling polarity) THEN
    min(D1, D2) < W_SET_out    max(D1+W_SET1, D2+W_SET2);
  ELSEIF (SET1 & SET2 do not have controlling polarity) THEN
       max(D1, D2) < W_SET_out    min(D1+W_SET1, D2+W_SET2);
  ELSEIF ((SET1 is controlling) & ((D1 < D2) & (D1+W_SET1 > D2+W_SET2 )))|
       ((SET2 is controlling) & ((D2 < D1) & (D2+W_SET2 > D1+W_SET1 ))) THEN
       Both SETs will mask each other;
  ELSEIF ((SET1 is controlling) & ((D1 > D2) & (D1+W_SET1 < D2+W_SET2 ))) |
            ((SET2 is controlling) & ((D2 > D1) & (D2+W_SET2 < D1+W_SET1 ))) THEN
       1- min(D1, D2) < W_SET_out    max(D1, D2);
       2- min(D1+W_SET1, D2+W_SET2)   W_SET_out   max(D1+W_SET1, D2+W_SET2);
  ELSEIF (((SET1 is controlling) & (D1 < D2)) | ((SET2 is controlling) &
       ((D2 < D1)))
        min(D1+W_SET1, D2+W_SET2 )   W_SET_out   max(D1+W_SET1, D2+W_SET2);
  ELSE
       min(D1, D2) < W_SET_out    max(D1, D2);
ELSE
       No Overlap; % no re-converging broadening/attenuation
ENDIF;
```

Figure 6.3 Proposed Characterization of Re-converging SETs.

If there is no overlap between the SETs, then the re-converging gate evaluates their propagation separately (as shown in Fig. 6.3).

6.2.4 Fault Propagation Analysis

The proposed analysis starts by identifying the Cone Of Influence (COI) for each output in order to evaluate its vulnerability to SETs. To do that, the technique recently proposed in [99] is adapted to compute all COIs in a single pass. The main steps of the COI evaluation are detailed in Alg. 4. The main idea is to assign a bit array called $BMP(n_i)$ to each node. The COI for each node is extracted using a backward depth-first traversal of unvisited nodes, i.e., whenever node n_j is reached by node n_i, the label of n_j is bitwise ORed with the label of n_i ($BMP(n_j) = BMP(n_j)|BMP(n_i)$). Thus, the set of nodes in the COI of a node correlates to the bits with value '1' in its bitmap. Thereafter, an SMT model of the design is built based on the developed standard gates SMT models. Next, an SET is injected at one gate at time

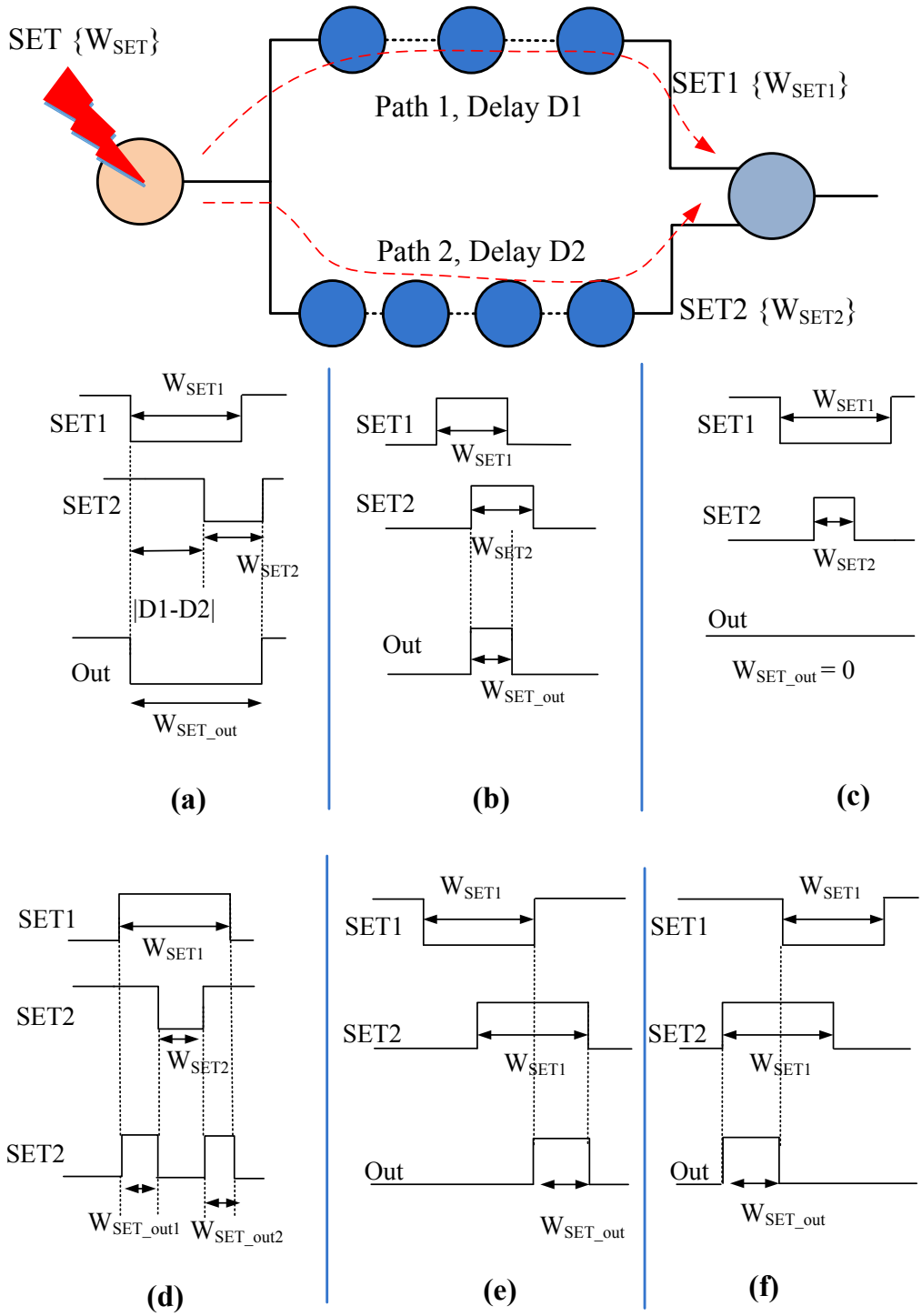


Figure 6.4 An Example on the SET Re-converging Scenarios.

and SMT solvers are utilized to exhaustively analyze the SET propagation from this gate to the primary output by verifying the following set of properties :

— Logical masking property : *is there an input vector which opens a sensitized path for*

the injected SET so it reaches the design output ?

— Temporal masking property : *what is the maximum strike time (T_{st}) for an SET within the clock cycle so it can be latched at a register ?*

— Electrical masking property : *what is the minimum width (MW) for an SET so it is not electrically masked and to be latched by a register ?* Electrical and temporal masking are mutually dependent and their effects should be considered simultaneously. The minimum width is computed based on the maximum strike time i.e., $MW \geq T_{clk} + T_h - \Delta tp(path) - Delay(path) - T_{st}$.

Based on the results of the verification of the first property, the SET propagation probability (that is not logically masked) is computed as follows :

$$PP = \frac{num(SAT\ instances)}{N} \tag{6.1}$$

Where $num(SAT\ instances)$ is the number of solutions in the randomly restricted search space N. This analysis is performed for both negative (i.e., PP_n) and positive (i.e., PP_p) SETs. The results of the verification of the second and the third property are the Minimum Positive Width (MPW) and the Minimum Negative Width (MNW) for positive and negative SETs, respectively. MNW and MPW are used to calculate the probability that an SET is not electrically and temporally masked as the ratio of these durations to the operated clock period duration T_{clk}. In case a fault has more than one MNW or MPW, then their disjunction is computed. Therefore, combining the impact of all masking effects and width variation, the probability of SET propagation from one node to the output is computed as follows :

$$P(z) = \frac{Pe_p(z) \cdot PP_p \cdot \frac{MPW}{T_{clk}} + Pe_n(z) \cdot PP_n \cdot \frac{MNW}{T_{clk}}}{2} \tag{6.2}$$

$Pe_n(z)$ and $Pe_p(z)$ denote the probabilities that a negative and a positive SET is injected at node z, respectively. These probabilities depend on the sensitive collection areas and the substrate or well voltage bias around the vulnerable node. In absence of these details as well as necessary details related to the energy distribution of the aggressor particles, it is assumed that $Pe_n(z)$ and $Pe_p(z)$ are equal. The vulnerability of an output node in the design can be estimated using the SET propagation probabilities from all the nodes in its COI as follows :

$$Vul(O_i) = \sum_{z \in allnodes} P(z) \tag{6.3}$$

This approach is accurate when the COIs of all outputs are mutually exclusive. By contrast, when COIs of different outputs overlap (or do not correspond to mutually exclusive events),

Algorithm 3 Modeling and Analysis of SET Propagation

CombinationlDesign{
Get technology node parameter ;
Sort gates topologically ;
Annotate each gate with its parameters ;
Compute_COI for each output ;
Generate SMT model **for each** gate ;
for each output
  for each gate
    Compute number of random tests N ;
    Generate all solution in search space ;
    Calculate propagation probabilities ;
  Compute output vulnerability ;
Compute SER ;
}

Compute_COI{
Clear all $visit\_flag$;
Clear all node bitmaps ;
Set all nodes bitmaps with their bithot ;
foreach $t_i \in T$
  $BitMap(t_i)$ =REACH_DF(t_i) ;
}
procedure REACH_DF(n)
  Set $visit\_flag$ of n ;
  foreach v_i in the adjacent list of n
    if (v is not visited)
      $BitMap(v)$ = REACH_DF(t_i) ;
      $BitMap(n) = BitMap(n)|BitMap(v)$;

---

then the probabilities for SET propagation to these outputs are correlated. In this case, if exact probabilities are required, then signal dependencies due to re-convergent fan-outs and/or correlated inputs have to be investigated. This leads to the path enumeration problem, where the number of paths that have to be enumerated independently can increase exponentially with the number of dependent re-convergent fanouts and correlated inputs [100]. Therefore, to avoid such complexity, COIs are assumed to be mutually exclusive as in Eq. 6.3 which can lead to safe over approximated probabilities. Finally, we estimate the SER of the design using the vulnerability of all outputs as follows :

$$\textbf{SER(comb)} = \sum_{O \in comb} Vul(O) \tag{6.4}$$

6.3 Experimental Results

The proposed methodology is fully automated. RTL synthesis to gate level, generation of the layout, and extraction of its details into an SDF file are automated using TCL scripting and EDA tools. An SMT model of the design is automatically built, from its gate level verilog netslit and our library of the standard gates SMT models, using Python scripting. The proposed analysis (outlined in Alg. 4) is fully performed using Python scripting and the *Yices* SMT solver. Experiments were conducted on an I7-3770K processor clocked at 3.50GHz with 16 GB RAM.

The proposed framework was implemented on the ISCAS85 benchmark circuits and different size array multipliers. The layouts of the analyzed designs were generated based on the FreePDK45 design kit [101]. This kit supplies technology files and layouts for a generic 45-nm process. It has been used to characterize the vulnerability of logic cells to soft errors [102]. Our first detailed analysis consisted on investigating the relationship between SET width variations and the SER. The results of the analysis of different designs that are depicted in Fig. 6.5 lead to the following observations :

1. SER increases as the width of the injected SET increases until it reaches a certain width threshold. After that threshold, no significant change in the SER is observed. This can be explained by the fact that different propagation paths have different SET width requirements. Therefore, increasing the width of the injected SET increases its probability to propagate through longer paths without being electrically masked. Nevertheless, if SET width is sufficiently large, then its width is large enough to propagate through all possible propagation paths i.e., logical and temporal masking dominate its propagation. Moreover, when taking into account the temporal masking, the SERs are reduced by factors proportional to the latching window and inversely proportional to the clock period.
2. It is essential to carefully model SET width broadening for accurate vulnerability evaluation. In Fig. 6.5, we investigate the impact of modeling the SET width broadening on the design SER. For example, when broadening is modeled, the SET width threshold for the *C499* design was evaluated to be around $300\,ps$. However, without modeling the broadening, the threshold SET width was evaluated to be around $800ps$. This is due to the fact that the SET width is only attenuated while propagating, i.e., larger widths are required for SETs to be able to reach the outputs and cause an error. The same results were observed for all other ISCAS85 circuits as shown in Fig. 6.5. To see how much inaccuracy can be introduced with such modeling, results shown in Fig. 6.5 demonstrate that it can cause significant underestimations of the SER.

The second detailed analysis consisted on investigating the applicability of the proposed framework to any combinational circuit. Table 6.1 shows the CPU time consumed by *Yices* to analyze all the ISCAS85 benchmark circuits. The reported times are the times consumed by the SMT solver to analyze all injection scenarios. It is observed that our framework's verifi-

Table 6.1 Comparison of Processing Times to Estimate SERs Between our Framework and Contemporary Techniques for ISCAS85 Benchmarks.

circuit	SSER [51]	SEAT [52]	MC [5]	**Proposed Methodology**	
	CPU (Sec)	CPU (Sec)	CPU (Sec)	CPU (Sec)	**SER**
c17	30.43	-	0.432	0.09	0.43
c432	269.71	6480	21.15	1.1	0.621
c499	36.90	12960	6.48	10.15	0.887
c880	273.20	6120	34.5	0.64	0.145
c1355	109.25	9720	-	15	0.315
c1908	120.23	64380	81	1.149	0.0555
c2670	309.53	32820	392.2	23.2	0.05
c3540	403.17	-	150.2	2.47	0.0254
c5315	4710.04	-	207.6	3.52	0.033
c7552	658.37	-	316.17	42.16	0.105

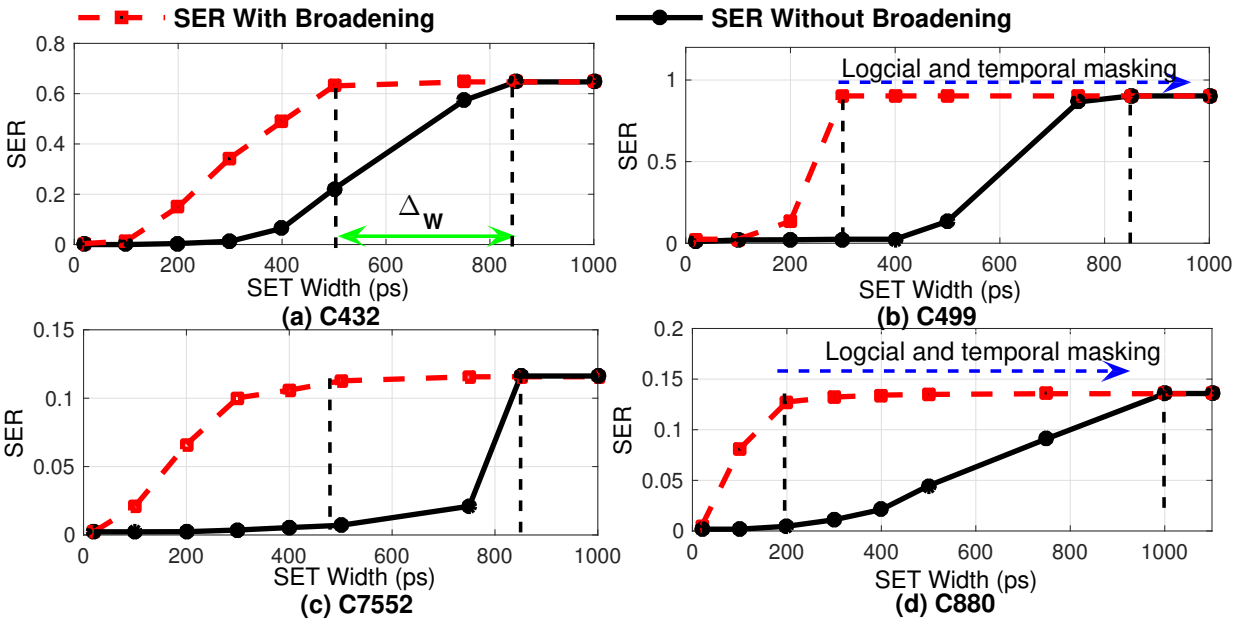


Figure 6.5 The Relationship Between SER and SET Width.

cation time for most of the circuits is much less than those reported with other contemporary methods.

6.3.1 SET Analysis For Multipliers

In general, multipliers are very complex benchmarks. This is due to the fact that increasing multiplier sizes quadratically increases the number of gates, but exponentially increases the number of re-converging paths. Therefore, the work on SER analysis of multipliers is rather limited.

In order to evaluate the scalability and the efficiency of the proposed methodology, different size array multipliers are modeled and verified. Their sizes range from *8* to *128* bits. The results of this analysis are reported in the sixth and the seventh columns of Table 6.2. Moreover, a comparison with state-of-the-art existing techniques (SSER [51] and SERA [41]) in terms of CPU time is also provided in Table 6.2. It is important to note that existing techniques do not take in consideration the impact of re-converging paths as explained before. When analyzing multipliers, we observed that such assumption has the following consequences :

1) it highly reduces the complexity of the SET propagation analysis. For example, when no re-convergence between SETs is considered, then each re-converging path is investigated separately. This means that the complexity of the analysis is directly related to the number of re-converging paths. However, when SET re-convergence is modeled, then the complexity of the analysis depends on the number of re-converging paths and their delays, SET widths, and SET arrival times as characterized in Section 6.2.3. However, as shown in Table 6.2, even with this assumption, SERA [41] requires 593 minutes to analyze a 32-bit multiplier, while SSER [51] requires 498 minutes to analyze a 24-bit multiplier. The proposed methodology is able to analyze a 128-bit multiplier in around 70 minutes, as shown in Table 6.2.

2) it highly affects the estimated SER. This is mainly because the number of re-converging paths in multipliers is huge. Moreover, re-converging paths can be short and the difference between their delays is very small due to the structure of the multiplier (such as the structure of array multipliers). Based on the extracted timing details from the layout it was observed that the difference between the delays of many re-converging paths was less than 200 *ps*. Moreover, based on the circuit level analysis performed in [102] injecting 15 MeV-cm^2/mg ion strike on FreePDK45 cells can lead to SET with widths range from 464 to 639 *ps*. Therefore, it is clear that SETs propagating in multipliers can have high chances to re-converge. In order to investigate the inaccuracy introduced by this assumption, the SER of the multipliers were computed with and without modeling SETs re-convergence. In the last column of Table 6.2, the percentage variation in the SERs ($|SER_{rec} - SER_{no\_rec}|/SER_{rec}$) is reported. Results

demonstrate that the errors in the SERs introduced by the lack of re-convergence modeling increases as the size of the multiplier increases. For example, the error in the SERs can be as high as 924% for the 128-bit multiplier.

Table 6.2 Comparison of SER Analysis Times for Different Multipliers with State-of-the-art Methods.

		SSER [51]	**SERA** [41]	**MC*** [5]	**Our Methodology**	
Size (bit)	**# of gates**	CPU (Sec)	CPU (Sec)	CPU (Sec)	CPU (Sec)	$\triangle$ SER %
4	124	34.85	6	27	1	4.24
8	568	271.03	78	TO	11.4	21.67
16	2.4K	5010.40	2472	TO	50.1	53.79
24	5.5k	29930.01	-	TO	381.12	90.17
32	10K	TO	35580	TO	690.249	131.84
64	44k	TO	TO	TO	1947.81	334.28
128	163K	TO	TO	TO	4170	924.08

* MC : Model Checking, 16-bit multiplier is the C6288 ISCAS85 benchmark
TO : Time Out.

6.4 Conclusion

This paper presents a comprehensive methodology to analyze and measure the vulnerability of combinational circuits to soft-errors due to SETs. A new model of SET propagation, that includes the impacts of all masking effects, the width variation (broadening and attenuation), and re-converging paths, is proposed. Moreover, a new formalism of SETs propagation models them as a Satisfiability problem by utilizing Satifiability modulo theories is proposed. An SMT-based exhaustive analysis of SET propagation is proposed. In the proposed methodology, gate level analysis is instantiated with the pre-characterized TPTs of the technology node and the exact gates timing extracted from the layout. The implementation of the proposed methodology on different combinational designs shows its accuracy, applicability, and scalability. For instance, it can analyze complex arithmetic circuits such as a 128-bit multiplier in about 70 minutes, while existing techniques fail to handle multipliers larger than 32 bits. Ongoing work promises to extend the proposed methodology to handle sequential logic in addition to combinational logic.

CHAPTER 7 ARTICLE 4 : TOWARDS FORMAL ABSTRACTION, MODELING, AND ANALYSIS OF SINGLE EVENT TRANSIENTS AT RTL

Summary of the Chapter

In this chapter, we introduce the proposed Register Transfer Level (RTL) abstraction and modeling approaches of the underlying behavior of SET propagation observed by gate level analysis (reported in Chapter 5 and Chapter 6) using Multiway Decision Graphs (MDGs). The proposed MDG-based modeling and verification (based on invariant checking) was introduced (and subsequently published) in a paper in the IEEE International Symposium on Circuits and Systems (ISCAS) on 2016. ***This published paper is reproduced in this chapter.***

Title : Towards Formal Abstraction, Modeling, and Analysis of Single Event Transients at RTL

Authors—Ghaith Bany Hamad, Otmane Ait Mohamed, and Yvon Savaria

Abstract—Soft errors due to Single Event Transients (SETs) have become one of the most challenging issues that impact the reliability of modern microelectronic systems at terrestrial altitudes. This is mainly due to the progressive shrinking of device sizes. Traditionally, the analysis of SETs has been carried out by simulations and experimental analysis. However, these techniques are resource hungry and require full details of the design structure and SET characteristics. This paper develops a hierarchical framework for formal analysis of SET propagation by (1) introducing Register Transfer Level (RTL) abstraction and modeling approaches of the underlying behavior of SET propagation using Multiway Decision Graphs (MDGs) ; and (2) investigating SET propagation conditions at RTL using a formal model checker. In order to illustrate the practical utilization of our work, we have analyzed different RTL combinational designs. Experimental results demonstrate the proposed framework is orders of magnitude faster than other comparable contemporary techniques. Moreover, for the first time, a decision graph based technique is developed to analyze multiplier designs.

Index Terms—Soft Errors, SET, MDG, multiway decision graphs, logical masking, CIC, invariant checking, model checking, multipliers, counterexamples.

7.1 Introduction

Single Event Transients (SETs) are becoming a major source of soft errors in digital designs [11]. SETs propagate more easily as the technology scales down. Moreover, the growing speed and complexity in new generation circuits increased the probability that SETs lead to soft errors [11]. Therefore, there is a growing need to analyze and estimate the impact of SETs on today's complex digital designs. Over the past two decades, several techniques have been proposed to analyze SET propagation at different abstraction levels. At gate level, researchers proposed different techniques such as fault injection [44] and formal verification methods [5]. However, these techniques are time consuming, resource hungry, and require full details (gate level net-list) of design structure. Thus, they are not applicable at early design stages.

Several methodologies have been proposed to perform the analysis at Register Transfer Level (RTL) such as ; fault simulation [74] and analytical techniques [75]. Other researchers have addressed this issue using formal verification methods such as Boolean Satisfiability solvers [6] and Probabilistic Model Checking (PMC) [56], [76, 4]. All these techniques suffer from the following shortcomings :

1. Contemporary formal verification based techniques are resource hungry and limited due to the *state explosion* problem. This is mainly due to intrinsic characteristics of their modeling technique. Indeed, in these techniques, SET propagation is modeled using concrete Boolean diagrams e.g., Binary Decision Diagrams (BDDs). With such techniques, a model checker rapidly runs out of memory, even when modeling moderate size designs e.g., 14-bit adder [56] or 15-bit multiplier [77].
2. Simulation based techniques (such as [74], [75]) have serious shortcomings as they are very time consuming for large designs with many primary inputs. Furthermore, these techniques have their drawbacks in terms of accuracy. This is mainly because their accuracy is determined by the ratio of the simulated sample size over the total vector space size.
3. At RTL, many of the details about design structure and SET characteristics are not available. Therefore, contemporary techniques make assumptions about SET propagation behavior. For instance, in [75, 56] and [76], SETs are modeled as bit flips. Such assumptions reduce the accuracy of the estimated Soft Error Rate (SER).

In this paper, a hierarchical framework is proposed to analyze SET propagation at RTL. This work is distinct from previous works in the following ways :

1. Abstraction is one of the most relevant techniques for addressing the *state explosion* problem [62]. Our proposed framework introduces a new abstraction approach. An

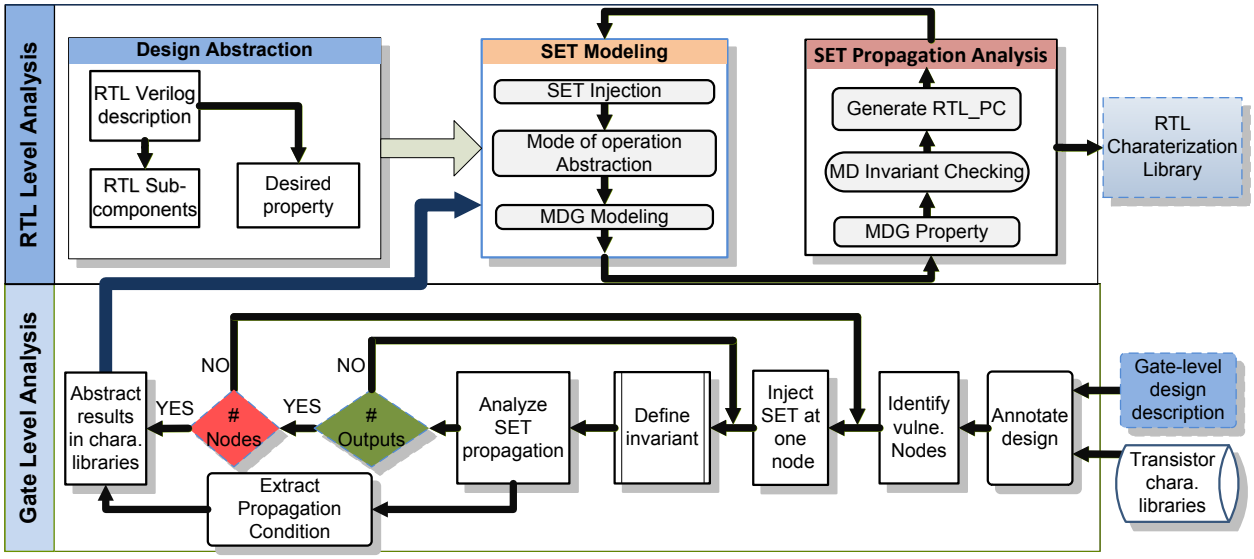


Figure 7.1 Steps of the Proposed Framework for the Investigation of SET Propagation at Gate and RTL Levels.

RTL component can have three modes of operation ; *Injection*, *Propagation*, or *Error-Free*. For each injection scenario, SET propagation at RTL is modeled based on the sub-components mode of operation and their gate level characterization libraries developed beforehand. Moreover, the proposed framework utilizes the Multiway Decision Graphs (MDGs) [57]. This decision graph provides different modeling options, rather than being limited to the boolean representation (such as [76]).

2. A new formal method to analyze SET propagation at RTL is proposed. The invariant checking tool from the MDG formal verification tool set [57] is adapted to perform this analysis. The results, which are SET propagation conditions for all injection scenarios, are reported as RTL characterization libraries.

Our results demonstrate that the CPU time and the memory required to analyze SET propagation are significantly reduced. Therefore, for the first time, a decision graph based technique is able to analyze fault propagation through complex arithmetic circuits such as large (16 by 16) multipliers.

7.2 Proposed Framework

The goal of this work is to develop new mechanisms to bridge the gap between design abstraction levels (gate and RTL). The verification process starts at gate level. Gate level libraries

which characterize SET propagation are developed to be utilized at RTL, as depicted in Fig. 8.1.

7.2.1 Gate Level SET Analysis and Characterization

At gate level, SET propagation is modeled by utilizing transistor level characterization libraries generated based on transistor level analyses such as [96]. The set of vulnerable nodes in a design is identified and an SET is injected at one of these nodes. Next, SET propagation is analyzed from this vulnerable node to each primary output. The results of this analysis are characterized. For each SET injection scenario, fault configurations that allow SET propagation to the output are reported. Different formal techniques can be adapted to perform this analysis such as [5], [71]. However, in this paper, the technique proposed in [5] is adopted, because it is the only technique that provides an exhaustive analysis and the generated results can be used to build the desired gate level libraries. As an example, SET propagation through the full adder shown in Fig. 8.12 is analyzed. Propagation conditions for each injection scenario are shown in Table 7.1. Generating such table for basic components is a one-time effort that can be done offline. Table 7.2 shows some of the basic combinational designs that have been characterized.

Table 7.1 Results of our Gate Level Analysis of a Full Adder

Vuln. Node	**Output Node**	**Propagation Conditions**
a=SET	S	$(b = 0/1) \wedge (c = 0/1) \wedge (S = SET)$
	Cout	$(c = 1) \wedge (b = 0) \wedge (Cout = SET)$
b=SET	S	$(a = 0/1) \wedge (c = 0/1) \wedge (S = SET)$
	Cout	$(a = 1) \wedge (c = 0) \wedge (Cout = SET)$
c=SET	S	$(a = 0/1) \wedge (b = 0/1) \wedge (S = SET)$
	Cout	$(a = 0) \wedge (b = 1) \wedge (Cout = SET)$
OG1=SET	S	$(c = 0/1) \wedge (S = SET)$
	Cout	$(c = 1) \wedge (b = 0) \wedge (Cout = SET)$
OG2=SET	S	NP
	Cout	$(a = 0) \wedge (Cout = SET)$
OG3=SET	S	NP
	Cout	$(c = 0) \wedge (Cout = SET)$
OG4=SET	S	AP
	Cout	NP
OG5=SET	S	NP
	Cout	AP

NP : No Propagation, AP : Always propagates

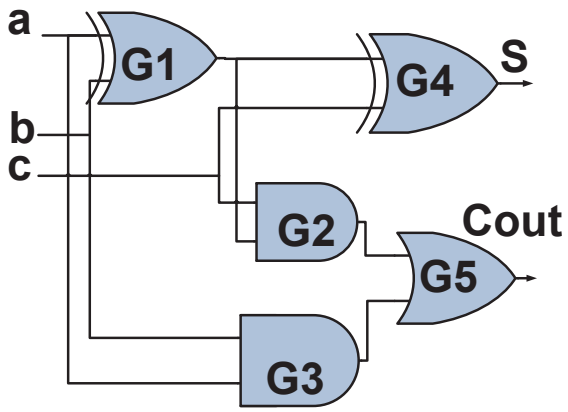


Figure 7.2 Gate Level Model of a Full Adder

Table 7.2 Benchmark Circuits Characterized at Gate Level

Circuit Name	Circuit Function	Total Gates	Input Lines	Output Lines
FA	Full adder	5	3	2
4-bit RCA	Adder	20	9	5
8-bit RCA	Adder	40	17	9
74182	CLA	19	9	4
74283	Fast Adder	36	9	5
C432	Priority Decoder	160 (18 EXOR)	36	7

7.2.2 Abstraction of SET Propagation at RTL

Contemporary techniques such as [103] proposed different abstraction approaches to improve the scalability of their analysis. However, these techniques are limited to certain design structures and tools. They eliminate details about the design structure which can be related to SET propagation. Thus, the accuracy of the analysis results can be affected.

Our framework abstracts irrelevant design details. In other words, we are only interested in modeling the design details that affect SET propagation and not the design's functionality. These details are defined based on three factors ; design structure, where the SET is injected, and the characterized gate level results. We abstract the RTL component behavior by its mode of operation : *Injection*, *Propagation*, or *Error-Free*. A component is in the *Injection* mode if a SET is injected inside this component. In this mode, the vulnerable nodes are the set of internal nodes. To abstract the behavior of this component, the Cone Of Influence (COI) for each output is identified as depicted in Fig. 8.3. The propagation conditions for each output is determined by disjuncting the gate level propagation conditions of all internal

nodes in its COI. A component is in the *Propagation* mode if it is in the propagation path of the injected SET (i.e., an SET propagates through its primary inputs). In this mode, the vulnerable nodes are the primary inputs of the component. A component is in the *Error-Free* mode if it is not in the propagation path of the injected SET. As depicted in Fig. 8.3, there are two types of *Error-Free* components :

1. Components which are outside the propagation zone of the injected SET (i.e., outside the COI of Out_i in Fig. 8.3). These components are eliminated to reduce the size of the design i.e., improving scalability while preserving accuracy.

2. Components which are in the COI of the injected SET. We preserve these components because their behavior affects SET propagation. However, if the component has different outputs then we eliminate all outputs which are not related to the propagation zone.

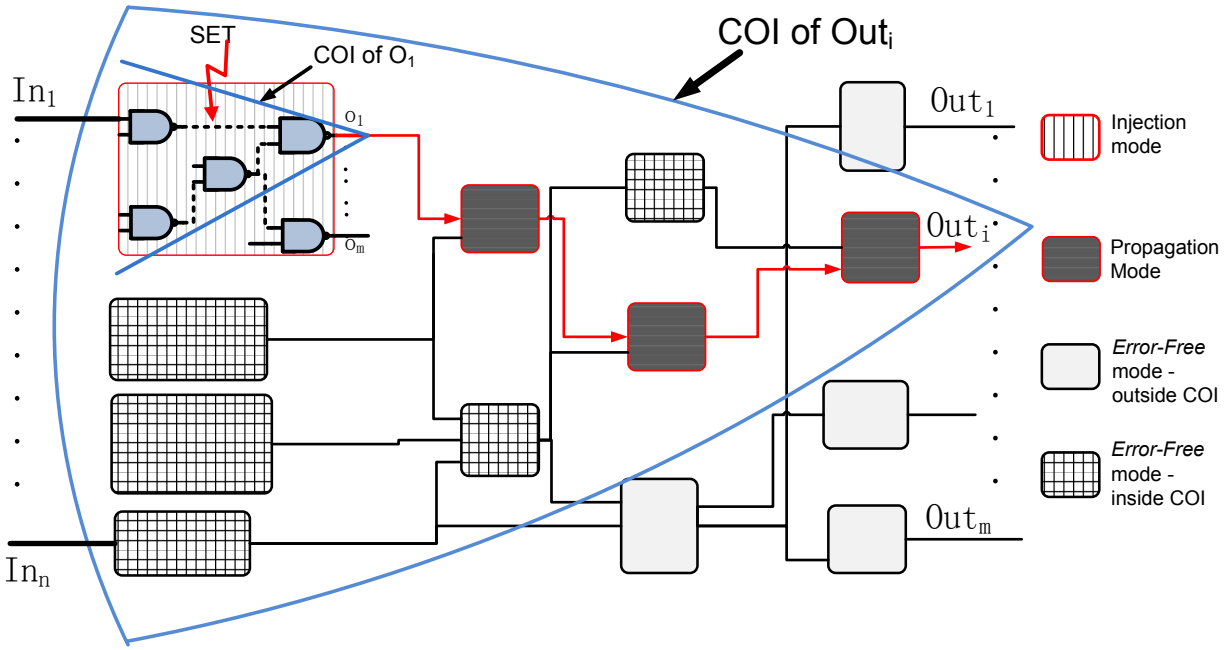


Figure 7.3 Decomposing an RTL Design and Deciding the Mode of Operation of its Sub-components Based on the Injection Scenario.

7.2.3 Formal RTL Modeling and Analysis

The inputs to the proposed analysis are the RTL structure description expressed in Verilog and the pre-characterized gate level results of basic digital components, as shown in Fig.

8.1. First, the RTL design is decomposed into a set of logic sub-functions, each of which is represented as sub-components.

For each injection scenario, the mode of operation for each sub-component is decided, where one component can be in *Injection* mode and all other components are either in *Propagation* or *Error-Free* mode, as depicted in Fig. 8.3. Next, each component is annotated with its pre-characterized gate level library. In the sequel, the corresponding RTL model of each component is built as a multiway decision graph (MDG). The formal logic underlying MDGs is a many-sorted first-order logic accommodating *concrete* and *abstract* sorts. *Concrete sorts* have enumerations of individual constants. Signals in components which operate in *injection* and *Propagation* mode are modeled as a *Concrete sort* that has values taken from the set $\{0, 1, SET\}$. The value SET is used when the signal is infected with a SET. Signals in components which operate in *Error-Free* mode are modeled as *Booleans* (0, 1). The proposed modeling allows the analysis of SET propagation in one version of the design without the need for two versions of the design (faulty and fault free) as required in BDD based techniques [76].

Next, we utilize the MDG verification tool set to analyze SET propagation at the RTL as it supports a mixture of structural and behavioral descriptions. SET propagation to a primary output (e.g., *po*) is evaluated by verifying the following properties over the MDG model of the design :

$$\texttt{RTL\_PC}? \quad [\ \mathcal{F}\ \ (\texttt{po} = \texttt{SET})\ ] \tag{7.1}$$

Which means : "*what are the fault configurations at the RTL (which we call RTL propagation conditions i.e.,* RTL_PC*) that allow the injected SET to eventually reach a po*". Our analysis demonstrates that if a SET has to propagate through n components to reach an output, then its RTL_PC is the combination of the gate level propagation conditions of all the components in the propagation path :

$$\texttt{RTL\_PC} = \texttt{PC}_{\texttt{m1}} \wedge \texttt{PC}_{\texttt{m2}} \ldots \wedge \texttt{PC}_{\texttt{mn}} \tag{7.2}$$

7.3 Experiments

In this section, we demonstrate how effectively SET propagation conditions can be evaluated directly at RTL using our framework. Our experiments are performed on a workstation with an Intel Core i7 running at 3 GHz and 24 GB RAM. To the best of the authors knowledge, this is the first time such abstraction and analysis of SET propagation at RTL is proposed. Therefore, for comparison purposes, we have implemented what we call the *Boolean* method. In this method, an RTL design is decomposed into sub components. Thereafter, propagation

of SETs through each component is modeled without any abstraction. The results of our proposed framework and the *Boolean* method for different ISCAS 85 benchmark designs are depicted in Fig. 7.4. The reported processing time and memory are the average CPU time and memory required to construct the decision graph and analyze SET propagation for one injection scenario. Results in Fig. 7.4 demonstrate that our framework significantly reduces the resources required to verify SET propagation by around 60%. Moreover, it can be observed that the CPU time and Memory are mainly consumed when constructing the MDG graph (around 9 times larger than verification time and around 4 times more than the memory consumed on verification). The proposed framework significantly reduces the size of MDG graphs. Hence, it reduces the CPU time and the memory required to construct MDG graphs by around 70% and 45%, respectively.

Decision graph based techniques (such as [56, 76, 4, 77, 104]) cannot handle some classes of circuits such as multipliers. This is mainly due to the combinatorial explosion in the number of nodes. For example, building a decision graph for a 15-bit multiplier requires over 12 million nodes (around 260 MB) [77, 104]. The number of nodes increases exponentially with the word size, and hence even much more powerful computers will have difficulty getting much beyond this point. As shown in Table 7.3, with the state-of-the-art techniques, model checkers run out of memory while constructing the graph for moderate size multipliers ($\geq$ 8-

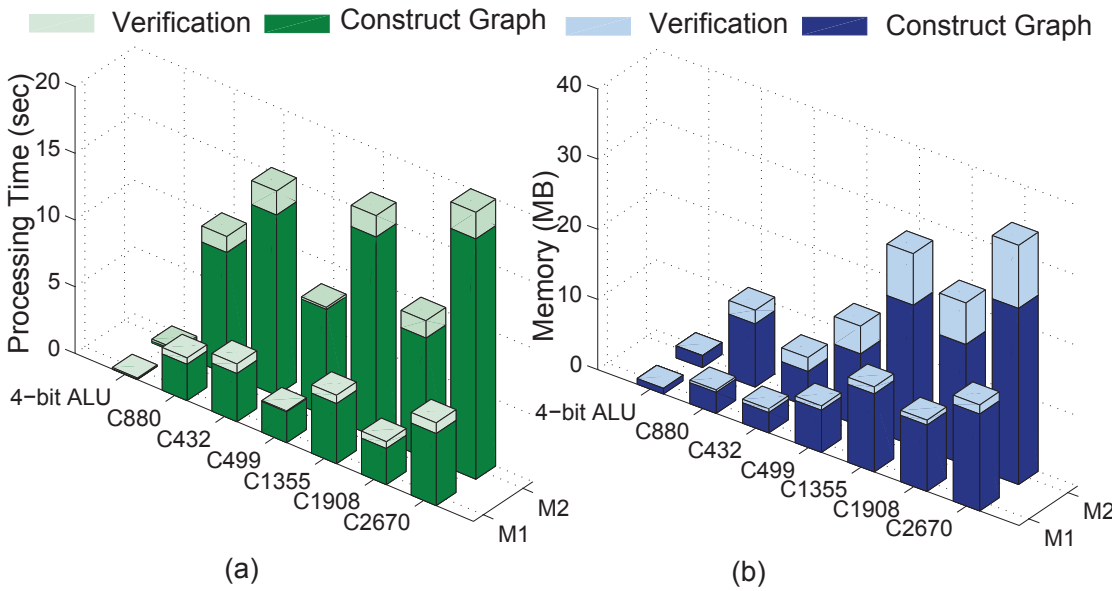


Figure 7.4 RTL Analysis of Combinational RTL Design, *M1* is Our Proposed Framework and *M2* is the *Boolean* Method. (a) Comparison Between *M1* and *M2* for the Processing Time. (b) Comparison Between *M1* and *M2* for the Memory Requirements.

Table 7.3 The Verification of SET Pulse Propagation for Multipliers

RTL Multiplier Design	**PI**	**PO**	**RTL Cells Count**	**Contemporary Techniques [56]– [4, 104]**	**Our Framework**
4-bit	8	8	12	√	√
6-bit	12	12	42	√	√
8-bit	16	16	56	×	√
10-bit	20	20	90	×	√
14-bit	28	28	182	×	√
16-bit	32	32	240	×	√

× : with this technique, the model checker runs out of memory while building the decision graph of this design.
√ : with this technique, it is possible to construct the decision graph and SET pulse propagation is verified.

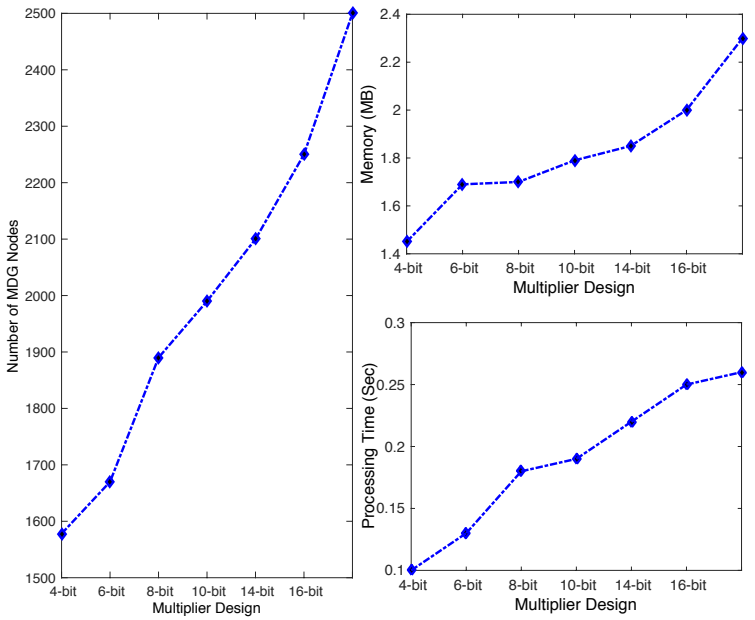


Figure 7.5 The Variation in the Average Processing Time, Memory, and Number of Decision Graph Nodes Required to Construct the MDG Graph and Analyze SET Propagation for one Injection Scenario.

bit). Our framework efficiently constructs the decision graph and analyzes SET propagation as shown in Table 7.3. Furthermore, the results in Fig. 7.5 demonstrate that with the proposed technique, the processing time, the memory, and the number decision graph nodes scale well with the size of the multiplier design. Therefore, the proposed methodology can replace some time consuming simulations to analyze larger designs at RTL. Moreover, we believe that

our formal framework can assist in building accurate system models by developing RTL characterization libraries. These libraries help circuit designers to evaluate the contribution of each vulnerable node on possible system failures. This allows identifying the nodes which need more protection if they are exposed to radiation-harsh environments or other related environments, such as those subject to crosstalk.

7.4 Conclusion

The Analysis of SET propagation at higher levels of abstraction is key to managing the complexity of today's VLSI chips. In this paper, we proposed a novel hierarchical framework to abstract, model, and investigate SET propagation. Gate level characterization libraries are utilized to model SET propagation at RTL. The proposed modeling and abstraction approaches significantly reduce the time and memory requirements required to model and analyze SET propagation at RTL. For instance, the CPU time and the memory required are reduced by more than 60%. For the first time, analysis of SET propagation through complex designs is possible e.g., 16-bit multiplier with less than *2 MB*. The probabilities of SET propagation can be computed and an accurate estimation of soft error rates can be developed based the results of the proposed RTL analysis.

CHAPTER 8 ARTICLE 5 : COMPREHENSIVE MULTILEVEL PROBABILISTIC ANALYSIS OF SINGLE EVENT TRANSIENTS PROPAGATION INDUCED SOFT ERRORS

Summary of the Chapter

In this chapter, we introduce a hierarchical multi-level probabilistic framework to model, analyze, and estimate SETs propagation in combinational designs expressed at different abstraction levels. Underlying probabilistic behavior of SET propagation is utilized to model this problem as a Markov Decision Process (MDP). PRISM model checker is adapted to analyze the probability of SET propagation for all vulnerable nodes. Furthermore, a new method to estimate the Soft Error Rate (SER) is proposed. The main idea behind this methodology was first introduced at RTL (and subsequently published) in a paper entitled "Efficient Multilevel Formal Modeling, Analysis, and Estimation of Design Vulnerability to Soft Error" which was published in the IEEE International On-Line Testing Symposium (IOLTS) on 2015. Thereafter, this methodology was extended to other abstraction levels and more accurate modeling is introduced based on the concept of fault space mapping. The general methodology was reported and submitted for publication. ***This submitted paper is reproduced in this chapter.***

Title : Comprehensive Multilevel Probabilistic Analysis of Single Event Transients Propagation Induced Soft Errors

Authors—Ghaith Bany Hamad, Otmane Ait Mohamed, and Yvon Savaria.

Abstract—Soft errors, induced by radiation, have a growing impact on the reliability of CMOS integrated circuits. The progressive shrinking of device sizes in advanced technologies leads to miniaturization and performance improvements. However, ultra-deep sub-micron technologies are more vulnerable to soft errors. In this paper, we propose a hierarchical multi-level methodology to model, analyze, and estimate Single Event Transients (SETs) propagation in combinational designs expressed at different abstraction levels (transistor to Register Transfer (RT) levels). Basic components are modeled and analyzed at low level and the results of this analysis are condensed into SET propagation tables. At high level, these tables are utilized to model the underlying probabilistic behavior of SET propagation as Probabilistic Automatas (PAs). Thereafter, the PAs of the different design components are

used to construct a Markov Decision Process (MDP) model for SET propagation through the complete design. A probabilistic model checker is adapted to analyze the probability of SET propagation for all vulnerable nodes. Furthermore, a new method to estimate the Soft Error Rate (SER) is proposed. Experimental results demonstrate that the proposed framework is orders of magnitude faster than contemporary techniques, while ensuring better accuracy. Moreover, it can handle designs as large as 256-bit adders.

8.1 Introduction

Despite considerable progress towards developing efficient methodologies for the functional verification of digital designs, advances in non-functional verification have been lagging. Non-functional verification is the type of verification that investigates the behavior of a Design Under Verification (DUV) in the presence of different uncertainties. The modeling and analysis of non-functional properties are more challenging than those of corresponding functional properties. This can be contributed to the difficulty encountered while characterizing and modeling the variation of the characteristics observed while transients due to single event effects [11, 66] propagate. Moreover, many details about the design structure and the uncertainty's characteristics may not be available at high abstraction levels. Thus, many assumptions about the uncertainty behavior are usually made, which impacts the accuracy of the generated results.

Soft errors, which started as a rather exotic failure mechanism causing anomalies in satellites, have become one of the most challenging types of uncertainties that impact the reliability of modern electronic systems. In the medical industry, for example, soft errors have been found responsible for the failure and the recall of many implantable cardiac pacemakers [7].

The exponential growth in the number of transistors per chip has brought tremendous progress in the performance of semiconductor devices ; however it increased the vulnerability of integrated electronics to soft errors. The expected Soft Error Rate (SER) per chip has been reported to increase 100-fold, from the 180nm to the 16nm CMOS technology nodes [105]. Therefore, there is a growing need to analyze and estimate the impact of soft errors on today's complex digital designs.

In general, single event effects (SEE) induced by radiation can have different effect such as Single Event Upsets (SEUs) and Single Event Transients (SETs) [11, 66]. This paper focuses on SETs. Several methodologies have been proposed to model the impact of SETs at different stages in the design cycle.

At transistor level, circuit simulation and experimental analysis have been performed [96, 3,

12, 20]. For instance, SET width broadening and attenuation while propagating have been investigated [96, 3, 12]. Some other studies investigated the effects of fan-out [20] and the impacts of input pattern and logic structure [96] on SET width. The analysis of SET propagation at transistor level can provide a certain level of accuracy for phenomena such as electrical masking and SET width variation. However, such analyses consume large amount of time and requires full details of the design structure and of SET characteristics. In other words, this type of analysis would be intractable at the chip level and is only tractable at the cell level (for hundreds of transistors at most) to get a certain level of accuracy. Hence, in order to investigate SET propagation in large scale designs, researchers proposed different methodologies to perform this analysis at higher abstraction levels. At gate level, researchers proposed different techniques such as fault injection [44] and formal verification methods [106]. At Register Transfer Level (RTL), SET propagation is analyzed using different techniques such as fault simulation [74], analytical techniques [75], and formal verification methods [56, 76, 6]. Contemporary formal based techniques model SET propagation as Discrete Time Markov Chains (DTMCs) [56, 76] or as Boolean Satisfiability problems [6]. Thereafter, a formal verification tool is adapted (such as SAT solvers [6] and Probabilistic Model Checkers (PMCs) [56, 76]) to investigate SET propagation. All aforementioned contemporary techniques operating at high abstraction levels suffer from the following shortcomings :

1. Techniques based on contemporary formal verification are resource hungry and suffer from a *state explosion* problem. This is mainly due to the intrinsic characteristics of their SET modeling technique. Indeed, in these techniques, each input vector is mapped to a unique state. Therefore, the corresponding Markov model has 2^M states (M primary inputs). Additionally, with these techniques, the formal model of the design size is doubled due to the requirement of two design versions, mainly a golden and a faulty version. For each injection scenario, in order to determine if a SET is propagating, the outputs of both the golden and the faulty versions are compared. With such modeling technique, any formal tool rapidly runs out of memory, even when trying to analyze moderate size designs e.g., a 14-bit adder [56].

2. Contemporary simulation and formal techniques have their drawbacks in terms of accuracy. This is mainly because these techniques explore a limited number of input vectors (random input assumption) to evaluate SET propagation probabilities, thus providing an incomplete analysis. Furthermore, at RTL and higher abstraction levels, many details about the design structure and SET characteristics are not available. Hence, assumptions are made about the SET propagation behavior. For instance, in [6, 56] SETs are modeled at RTL as bit flips. Such assumptions reduce the accuracy of the estimated SER. Our results demonstrate that the existing techniques, which

operate at gate and higher levels such as [6, 5, 4] generally provide an underestimated propagation probabilities.

In summary, the following important questions are not appropriately addressed in the literature so far :

1. *How to improve the usability of results of the transistor level analysis ?*
2. *How to measure the vulnerability of complex designs at high abstraction levels without losing the accuracy provided from low level analysis ?*
3. *Is it possible to improve scalability while preserving accuracy ?*

To answer these questions, we introduce a novel methodology to estimate the vulnerability of combinational designs to soft errors ; this work is distinct in the following ways :

1. For the first time, a multi-level formal verification framework is proposed to analyze SET propagation at gate and Register Transfer (RT) levels. SET propagation tables, which report the SET propagation behavior at lower abstraction levels are developed. These tables are utilized at higher abstraction levels to introduce awareness about the underlying probabilistic behavior of SET propagation.
2. Efficient probabilistic abstraction and modeling techniques of SET propagation are proposed. At high abstraction levels, we describe two efficient reduction methods, namely the Cone Of Influence (COI) and the component mode of operation methods. At any abstraction level, SET propagation is modeled based on the proposed fault space mapping technique. The propagation of high level faults for each sub-component is modeled as *Probabilistic Automatas (PAs)* based on the propagation probabilities of low level faults reported in the pre-characterized sub-component propagation table. The *PAs* of all sub-components are modeled as *Markov Decision Processes (MDPs)*. Thereafter, SET propagation is quantitatively analyzed using the proposed formal probabilistic verification technique that utilize the power of PMC. The results of this analysis are the SET propagation probabilities for all vulnerable nodes. Finally, theses probabilities are utilized to estimate SERs.

In this work, we implemented the proposed framework and applied it to different combinational designs modeled at gate and RT levels. Compared with existing techniques [6, 5, 4], our results demonstrate that the proposed framework significantly reduces the number of states (i.e., memory) and the CPU time required to analyze SET propagation. Furthermore, the proposed framework improves the accuracy of the measured design vulnerability (i.e., estimated SER). The results of the proposed analysis can be used to further improve the efficiency of the analysis of soft error mitigation techniques such as Triple Module Redundancy (TMR)

and retiming. Furthermore, it can improve the efficiency of the verification methodology by reducing verification time, thus enabling improved *time-to-market*.

The rest of this paper is organized as follows. Section 8.2, provides evidence of the problems we are addressing and gives an overview of probabilistic model checkers. Section 8.3 explains the main steps of the proposed framework. The proposed design reduction is explained in Section 8.4. In Section 8.5, the proposed SET propagation modeling and analysis are explained. In Section 8.6, we explain the implementation of the proposed framework for both gate and RT levels. In Section 8.7, our results are reported, discussed, and compared with the results produced by contemporary techniques. Section 8.8 concludes this work.

8.2 Background and Problem Formulation

8.2.1 Functional vs Non-Functional Verification

Designers and researchers found that the best way to build complex hardware designs is to start from high level descriptions and synthesize them all the way down to layout. This methodology is only applicable to synthesizable designs. With synthesis the code representing the design at one abstraction level can be translated into lower level implementations using pre-characterized rules and libraries. In other words, a design is synthesizable if the synthesis tool has the synthesis library (i.e., from which the low level implementation can be generated) for each part of the design. Therefore, the main concept in the design methodology is to utilize the lower level details from pre-characterized data to build large designs. In each synthesis phase between two abstraction levels more details about the design structure are added.

Unfortunately, when it comes to non-functional design verification, there is a different culture, and there is no unified approach. Design verification at one abstraction level relies on the information provided at this level. As shown in Table 8.1, low level analysis is very detailed, however, it is resource consuming and typically not applicable for large designs. On the other hand, higher level analyses are more time and resource efficient. However, their results have limited accuracy and may not provide much useful information to the designers about the design behavior in presence of different kinds of uncertainties. Therefore, there is a growing need to reduce the gap between the fault analyses at different abstraction levels. To achieve this goal, the usability of the results of the fault analysis at each abstraction level has to be improved. New methods to abstract the design details that directly affect fault propagation are required. However, we have to make sure that the additional overheads in terms of learning and verification are reasonable.

Table 8.1 Comparison Between High Level and Low Level Analysis

	Analysis at Gate and Higher Level	**Analysis at Transistor and lower Level**
Fault Modeling Accuracy	Very abstract, does not reflect actual fault behavior	Very detailed, reflects actual behavior
Complexity	Less complex	Very complex
Results Accuracy	Under/over estimation	More accurate
Memory	Less	More
CPU Verification Time	Less	Much more

8.2.2 Probabilistic Model Checking & PRISM

Probabilistic Model Checking (PMC) is a formal verification technique that can be applied to systems with stochastic behavior [107]. It does not only provide a Yes/No answer on whether a property holds. PMC can also quantify the probability (min/max) that the verified property is satisfied. Moreover, it has a wide range of applications in fields such as communication protocols and reliability analysis.

In this work, we use *PRISM* [82], which is an efficient probabilistic symbolic model checker. It employs efficient algorithms and data structures such as Binary Decision Diagrams (BDDs). In addition, *PRISM* supports different implementations of Markov chains, namely discrete-time and continuous-time Markov chains and MDP. It also supports a wide range of probabilistic temporal logic to specify the properties to be verified such as PCTL, PCTL*, and CSL.

8.3 Proposed Framework

This paper addresses the verification problem using a unified approach, which utilizes new mechanisms to bridge the gap between design abstraction levels. In the proposed framework, the verification process starts as early as possible, while providing the flexibility to move across different abstraction levels.

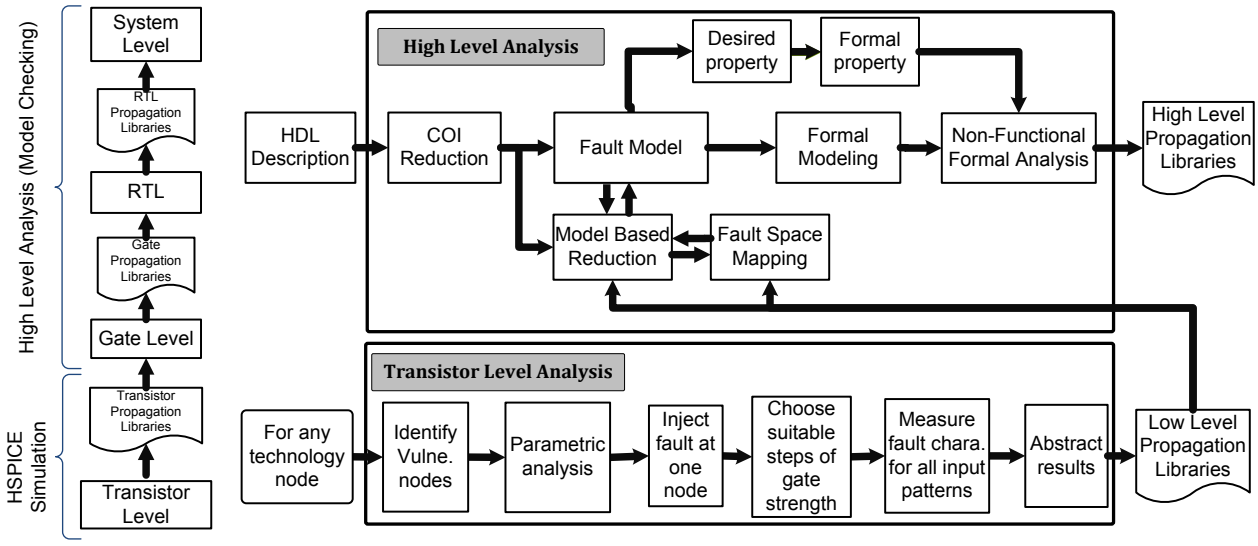


Figure 8.1 The Proposed Multi-Level Framework for Modeling and Investigating Fault Propagation.

8.3.1 Proposed Framework Steps

The proposed methodology is divided into two parts based on the abstraction level where it operates. At low level, it performs a transistor level analysis, but at high level it works with gate and higher level abstractions. As depicted in Fig. 8.1, the proposed multilevel analysis comprises the following main steps :

Step 1 : Performing detailed **transistor level analysis** of the fault propagation through standard logic CMOS gates such as AND and NAND using HSPICE for a certain technology node. Then, we characterize the results as transistor-level propagation tables.

Step 2 : Performing **design structure reduction at high level** for each primary output and fault injection scenario. In Section 8.4, both the cone of influence reduction and the component based reductions are explained in details.

Step 3 : Performing **high level formal modeling and analysis** of the probabilistic behavior of fault propagation at gate and higher abstraction level. This step is performed for standard digital designs. Then, we characterize the results as propagation tables which can be used at higher abstraction levels. in other words, the propagation table of one circuit at one abstraction level (e.g., gate-level) is utilized to model the fault propagation of this circuit (when it is part of a larger design) at higher levels (e.g., RTL). As an example, Fig. 8.2 depicts a chain of components which can be gates, RTL components, systems level components. A SET is injected at the input of *component 1*. The probability for this fault to propagate to

the output j is

$$P(j = F) = \prod_{i=1}^{3} P(C_i = F | C_{i-1} = F)$$

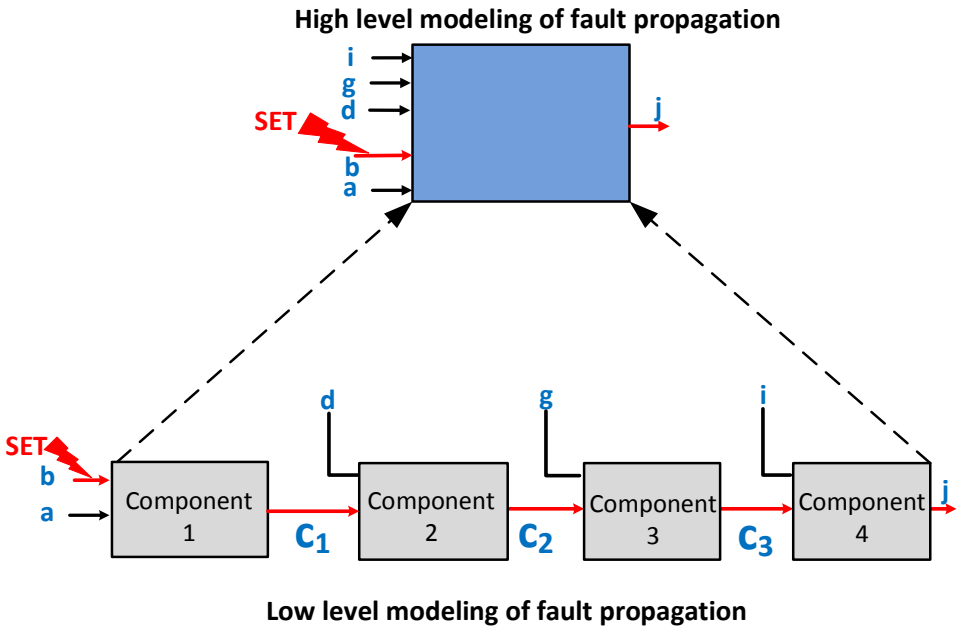


Figure 8.2 The Concept of Modeling SET Propagation at High Level Based on Low Level Propagation Details

Recently, different fault simulation [108] and formal model checking based techniques [5, 4] were proposed to generate such tables. In these techniques, propagation tables report the set of input vectors that allow fault propagation (called the Critical Input Combinations (CICs)). Later in this paper, we will explain in more details how the propagation tables developed in this paper provide more accurate information about fault propagation than other tables (developed in [5, 4, 108]).

8.3.2 Transistor-Level Analysis

In this analysis, the impacts of input patterns, fault polarity, and propagation paths on fault characteristics are investigated. Such analysis can be done by performing the following main steps :

— Fault propagation paths are divided into four main categories : chain of similar gates, chain of different gates, convergent paths, and divergent paths. Then, the vulnerable nodes in each category are identified.

— Thereafter, parametric analysis is performed to find out the threshold amplitude and duration for different gate strengths for a certain technology node.
— Next, representative circuits for each of category are analyzed using HSPICE circuit simulation. The purpose of this analysis is to measure the fault characteristics variation due to the input pattern and the logic structure. This step is done for all input patterns and for all vulnerable nodes.
— The last step of the analysis includes the abstraction of Transistor-level Propagation Tables (TPT) to describe SET propagation behavior at transistor level as depicted in Fig. 8.1. This detailed analysis is a one-time effort and can be done offline for a certain technology library.

In this work, we build our formal model at gate and higher abstraction levels using the results of our analysis of SET propagation at the transistor level as previously reported in [96], [109].

8.4 High Level Design Reduction

Reduction is one of the most relevant techniques for addressing the *state explosion* problem. The main purpose of reduction is to eliminate irrelevant details. However, what constitute relevant details is often left open to interpretation. In this work, the relevant details are defined as those which directly affect SET propagation. These details are identified based on three factors ; design structure, injection scenario (i.e., where the SET is injected), and low level propagation tables. To perform the proposed reduction, a design which is expressed at any suitable abstraction level is decomposed into main sub-components. Then, the set of vulnerable nodes are identified, which can be a gate, a wire, or a RTL component. The first step is to identify the Cone Of Influence (COI) for each output node. To do that, the framework recently proposed in [99] is adapted to compute all COIs in a single-pass. The main steps of the COI evaluation are detailed in Algorithm 4. All components in the design are associated with a visited flag and a bit array encoding, i.e., a bitmap. Each component is represented with one bit in this bitmap, i.e., the i-th bit in the bitmap correlates to the i-th component. The bitmap associated with a component n_i is denoted $BMP(n_i)$. Initially, all visited flags are set to *false*, and all components are labeled with a 0 bitmap. All components are labeled with a *one-hot* encoding of their variable index, i.e., $BMP(n_i) = BitHot(i)$. As proposed in [99], the COI for each node is extracted using a backward depth-first traversal from the output, i.e., whenever node n_j is reached by node n_i, the label of n_j is bitwise ORed with the label of n_i ($BMP(n_j) = BMP(n_j)|BMP(n_i)$). Thus, the set of nodes in the COI of a node correlates to the 1 bits in its bitmap.

In order to further reduce the size of the design representation, the component based reduction

Algorithm 4 Proposed Formal Modeling and Analysis

Inputs : $Prop\_Tables$, Structural description (R)
Outputs : SER, $Prop\_Tables$

```
SET_Model_Analysis{
Get Prop_Tables, R ;
Sort components topologically ;
Annotate each component with its Prop_Tables ;
Compute_COI(foreach output) ;
foreach output
    foreach component c_i
        Inject SET ; % Generation mode
        Extract SET probabilities from Prop_Tables ;
        Decide Mode_Of_Operation(all components) ;
        Build PA foreach component ;
        Build MDP model ;
        Investigate propagation probabilities ;
        Store propagation probabilities ;
    Compute output vulnerability ;
Compute SER ;
}
```

```
Mode_Of_Operation{
foreach (c_j not c_i)
    if (BitMap(c_i) ^ BitHot(c_j) == BitHot(c_i))
        c_j operates in propagation mode ;
        Extract SET probabilities for PIs from Prop_Tables ;
    else
        c_j operates in Error-Free mode ;
        Extract c_j Boolean behavior ;
        Calculate P(0) and P(1) ;
}
```

```
Compute_COI{
Clear all visit_flag ;
Clear all node bitmaps ;
set bitmap foreach node BitMap(i) = BitHot(i) ;
foreach  t_i ∈ T
    BitMap(t_i) =Reach_DF(t_i) ;
}
procedure Reach_DF(n)
    Set visit_flag of n ;
    foreach  v_i in the adjacent list of n
        if (v is not visited)
            BitMap(v) = Reach_DF(t_i) ;
            BitMap(n) = BitMap(n)|BitMap(v) ;
```

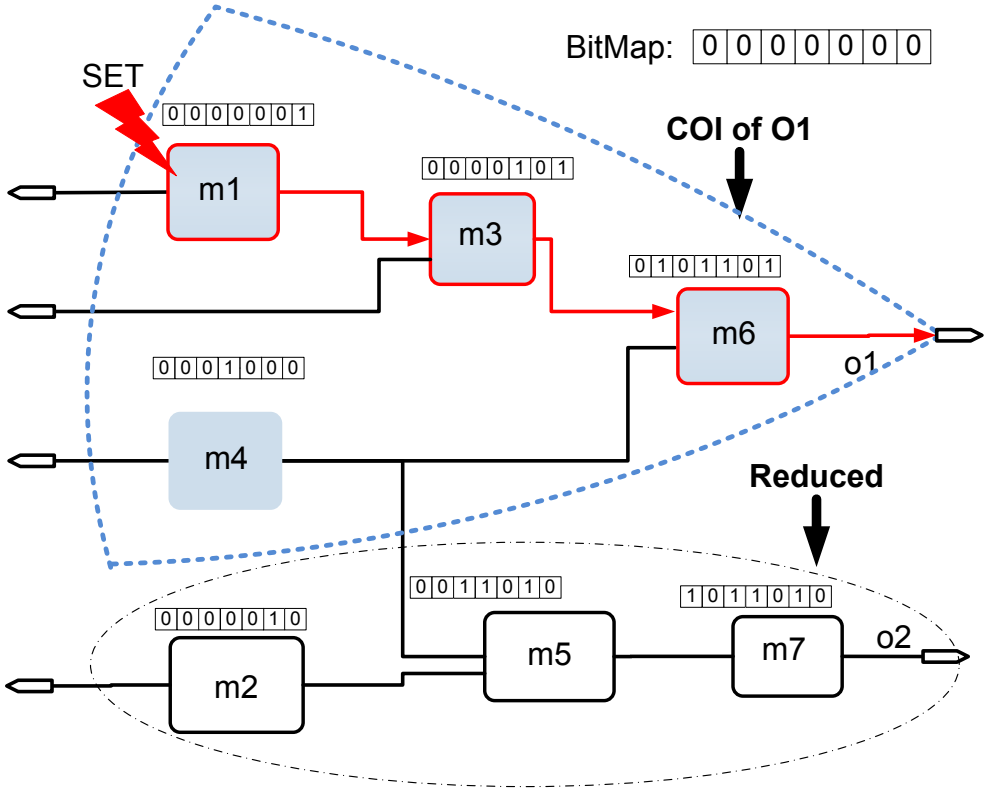


Figure 8.3 COI and Model Based Reduction.

technique at RTL (proposed in [4]) is adapted. After determining the COI of each output, the fault models for each COI are generated. Each fault model is the result of injecting a SET at one component in the COI. Each component in a fault model is modeled based on its modes of operation ; *Generation*, *Propagation*, or *Error-Free*. A component is in the *Generation* mode if an SET is injected internally in this component. A component is in the propagation mode if it is in the propagation path of an injected SET (i.e., the SET propagates through its primary inputs). A component in a COI is in the *Error-Free* mode if it is not in the propagation path of the injected SET. In this work, one component can be in *Generation* mode at a time and all other components are either in *Propagation* or *Error-Free* mode and this is decided based on their bitmaps. Let us assume that component c_j is in the *Generation* mode. If we need to decide the mode of operation of c_i, then if the bitwise AND of $BMP(c_i)$ and $BMP(c_j)$ is equal to $BitHot(cj)$ then c_i is in the propagation path of c_j. For example, consider the circuit illustrated structurally in Fig. 8.3. Assuming that an SET is injected at *m1*. A bitwise AND of $BMP(m_3)$ and $BMP(m_1)$ is equal to $BitHot(m_1)$, i.e., m_3 is in the *Propagation* mode. Similarly, it can be determined that m_6 is also in the *Propagation* mode, and *m4* is in the *Error-Free* mode. Next, the SET propagation behavior for the components

in the *Generation* and the *Propagation* mode is modeled based on the low level propagation probabilities. For components which operate in the *Error-Free* mode, the probabilities of having 0 or 1 at the output of these components are evaluated using the *Exhaustive Boolean Truth Table* or the *Signal Probability (SP)* technique [110]. These probabilities are used to compute the probabilities of SET propagation.

8.5 High Level Formal Modeling and Analysis

In this section, we explain in details the proposed modeling and analysis of SET propagation at high abstraction levels.

8.5.1 Fault Space Mapping

At each abstraction level, some faults can lead to design failure. Verification engineers define possible fault candidates based on the amount of details available about the design structure at one abstraction level. The number of faults in the design increases as it is modeled at lower abstraction levels. However, faults at one abstraction level do not have the same weight, i.e., possibility of occurrence. To demonstrate this concept, Fig. 8.4 depicts the Fault Tree (FT) of the fault space mapping between faults considered at different abstraction levels. In this FT, each fault at each abstraction level, which is considered as a Top Level Event (TLE), occurs due to faults at lower abstraction levels (i.e., Low Level Events (LLEs)). This FT can be characterized into *cut-sets* such that for each TLE at any abstraction level, there exists fault configurations down to the transistor level which makes this TLE true. For example, as

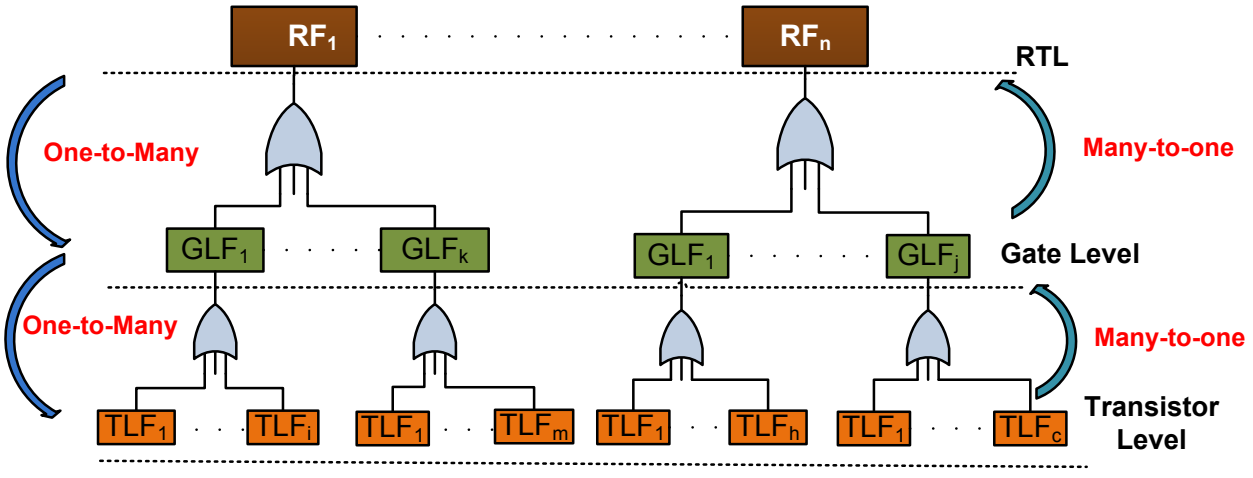


Figure 8.4 Fault Tree of our Proposed Fault Space Mapping.

shown in Fig. 8.4, RTL Faults (RFs) are the TLEs of the Gate Level Faults (GLFs) which are in turn the TLEs to the Transistor Level Faults (TLFs). Each High Level Fault (HLF) can be mapped through a one-to-many mapping to its corresponding set of Low Level Faults (LLFs) realization, which is defined as a correlation group. Therefore, the probability of an HLF is defined as follows :

$$\mathbf{P}(HLF) = \bigcup_{i=1}^{N} \mathbf{P}(LLF_i)$$

8.5.2 Proposed Formal Model Construction

It is evident that contemporary techniques do not fully utilize the power of formal verification methods. This is mainly because in spite of using the formal verification methods, the design is modeled similar to the fault simulation based method. For instance, these techniques test the response of both a faulty and a golden (i.e., *Error-Free*) version of the design for each test vector (input combination) as shown in Fig. 8.5(a). The proposed modeling is also limited to Boolean representation. Thus, in order to determine whether a fault is propagating to the design output (i.e., is this a faulty 0 or a faulty 1 ?) this output is compared with the *Error-Free* output. With such modeling scheme, the formal model size is doubled. Therefore, these technique are limited even when scalability improvement techniques are adapted as demonstrated in [56, 76].

In this work, the proposed modeling takes as input the design which is reduced using both the COI reduction and the model based reduction. All the details related to SET propagation are modeled in only one version of the design. This can be achieved thanks to the expressiveness

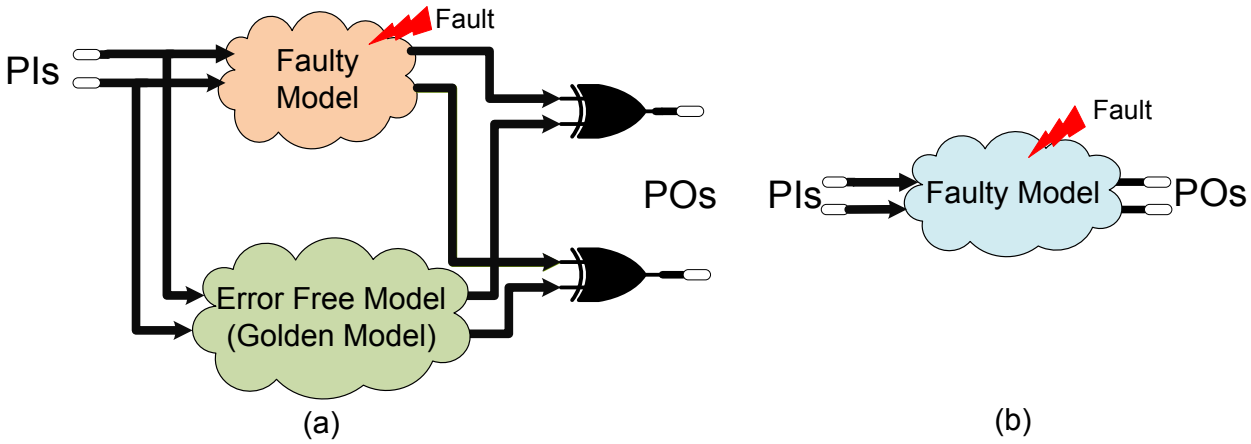


Figure 8.5 Comparison Between the Proposed Framework and the Contemporary Techniques.

power of formal methods. It is also of interest that our model is not limited to Boolean representation. In our model, signals which are in the propagation path of each injected fault take the values *{T, X_F}* instead of *{0, 1}*. A signal takes the value X_F if it is faulty i.e., $(1 \rightarrow 0)$ or $(0 \rightarrow 1)$. The *Error-Free* signal carries the value T. Components which operate in the *Error-Free* mode are modeled based on the probability of their output to be logic '1' or '0'. Signals in such component are modeled in Boolean. With such modeling, it is possible to determine *"the possibility for the output to eventually (at one state of the design) become faulty"*, which can be expressed as follows with a formal property : $F((Out = X\_F))$. This is equivalent to $AG((Out = T))$ which means *"for all possible paths and all possible states, the output (out) is equal to T"*. Furthermore, with the proposed model, it is not required to model the design behavior over all input combinations, because propagation tables abstracting low levels analysis are utilized as shown 8.5. Thus, the probability under which a signal at higher level will have the value T or X_F is pre-characterized. Table 8.2 provide a detailed comparison between our modeling approach and the contemporary methods which are adopted by most state-of-the art techniques.

The next step in the proposed framework is to build a probabilistic model for the fault propagation. In this process, the function *PA_gen* (line 16 in Algorithm 4) models the behavior of each component as a *PA* which is formally described as :

— S, a finite set of states,
— $\delta \subseteq (S \times S)$, a set of transitions between the states,
— PPr, a set of propagation probabilities,

Table 8.2 Comparison Between the Proposed Framework and the Contemporary Techniques

	(a) Contemporary Methods	**(b) Our Framework**
Model Size	Double the size	Original size
Modeling Technique	Modeling based on all input combination	Model based on fault propagation
Abstraction level	RTL and higher levels	Gate and higher
Reduction	PIs based partitioning	POs based partitioning
Abstraction	Eliminate design details which might be related to fault propagation, e.g., DTR abstraction	Model based abstraction by utilizing details from low level propagation tables
Preserve Accuracy	Loss accuracy to improve scalability	Preserved

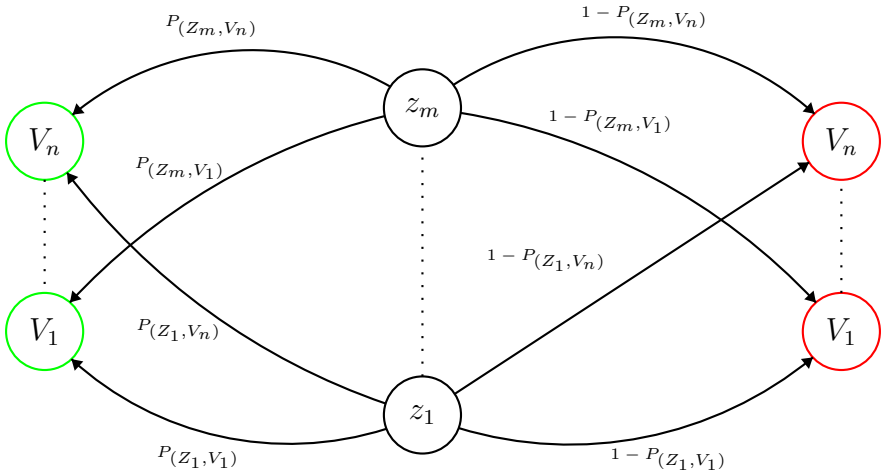


Figure 8.6 General Probabilistic Automata for any Component.

— $\pi : \delta(S \times S) \rightarrow (P, 1-P)$, a transition function assigning probability $P \in PPr$ for each transition.

The general probabilistic automata for fault propagation behavior is depicted in Fig. 8.6. Basically, a fault propagates from a node $\mathtt{z_i}$ to an output node $\mathtt{v_j}$ with probability $\mathtt{P_{(z_i,v_j)}}$ and it will be masked with probability $\mathtt{1-P_{(z_i,v_j)}}$. Next, these *PAs* are modeled as separate *PRISM* modules using the *PA_model* function, as shown in Algorithm 4. It is important to mention that the PA of each component is built based on its mode of operation for each injection scenario.

8.5.3 Proposed Markov Modeling of SET Propagation

In this paper, SET propagation is modeled as a Markov chain which is composed of the PAs of all the sub-components. With Markov chains, the set of probabilistic paths from the initial state and the desired state are identified. In these paths, the probability between any pair of states is identified such that the probability of a path ω is $P(\omega) = P(s, s_1) * \ldots * P(s_{n-1}, s_n)$. However, there are different implementations of Markov chains such as DTMC and MDP which can be implemented to model and analyze SET propagation. The type of Markov chain that should be used is decided based on the problem the verification engineer is trying to solve. Contemporary techniques (such as [76, 56]) model SET propagation through digital designs as DTMCs. However, in this work, the SET propagation is modeled as MDP due to the following reasons :

1) MDPs accurately models SET propagation through digital designs : MDPs allow nondeterministic choices while with DTMCs everything is probabilistic. This difference can be clearly

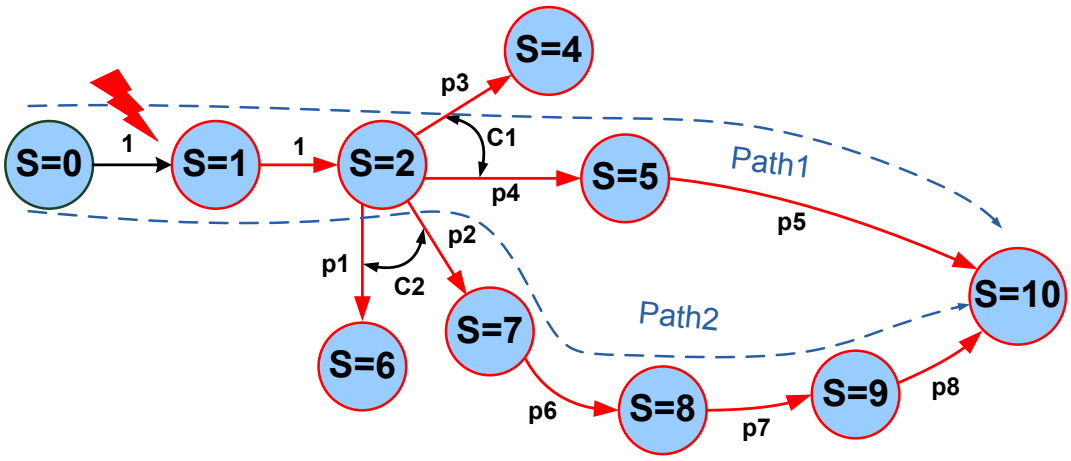


Figure 8.7 Illustrative Example for the Difference Between Modeling SET Propagation as MDP and DTMC.

seen when different probability distributions over a successor state exist. An example on this case of SET propagation in digital design is depicted in Fig. 8.7 S2, there are two probability distributions which depend on two actions *C1, C2*. This usually occurs at diverging node where each diverging path represents a probability distribution and all paths have the same triggering action. Modeling this behavior using DTMC means that we are assigning equal probability i.e., fault propagation probability through any diverging path is equal to $\frac{1}{\#\,of\,divergening\,path}$. In other words, fault propagation probability is reduced by this factor at each diverging node as shown in Table 8.3. However, experimental results at transistor and lower abstraction levels have proven that diverging paths may not reduce the fault propagation probability. On the contrary, for some design structural conditions, diverging paths can increase the probability of propagation by broadening the SET width as demonstrated in [96, 20]. On the other hand, when modeling such behavior with MDP, then the choice of the path through which the SET will propagate is non-deterministically determined which reflects more the actual SET propagation behavior.

2) MDPs allow modeling SET propagation through different paths while DTMCs do not : An injected SET can have different propagation paths to reach one output and cause a failure in the design operation. As shown in Table 8.3, with MDP, we can evaluate the minimum and the maximum probabilities to reach the desired state of a design. Since the goal is to analyze the design tolerance to SETs, then the path with the minimum probability is considered as the best propagation path. The path with the maximum probability is considered as the worst propagation path. Therefore, modeling SET propagation as MDP allows the analysis of worst case scenarios of SET propagation. However, by deploying the DTMC modeling approach, such quantitative evaluation is not possible. For instance, the probabilities evaluated with

Table 8.3 Illustrative Example for the Difference Between Modeling SET Propagation as MDP and DTMC

DTMC	**MDP**
What is the probability of eventually reaching the state **S=10** ? ? ?	
Conditions C1 & C2 are valid	
Path1 $\Rightarrow$ S=0 $\rightarrow$ S=1 $\rightarrow$ S=2 $\rightarrow$ S=5 $\rightarrow$ S=10	**Path1** $\Rightarrow$S=0 $\rightarrow$ S=1 $\rightarrow$ S=2 $\rightarrow$ S=5 $\rightarrow$ S=10
Path2 $\Rightarrow$ S=0 $\rightarrow$ S=1 $\rightarrow$ S=2 $\rightarrow$ S=7 $\rightarrow$S=8 $\rightarrow$ S=9 $\rightarrow$ S=10	**Path2** $\Rightarrow$S=0 $\rightarrow$ S=1 $\rightarrow$ S=2 $\rightarrow$ S=7 $\rightarrow$S=8 $\rightarrow$ S=9 $\rightarrow$ S=10
P(Path1)= $\mathbf{0.5} * (1 * 1 * p4 * p5)$	**P**(path1)= $1 * 1 * p4 * p5$
P(Path2)= $\mathbf{0.5} * (1 * 1 * p2 * p6 * p7 * p8)$	**P**(Path2)= $1 * 1 * p2 * p6 * p7 * p8$
$\mathbf{P}_{\texttt{DTMC}}?[\texttt{F(S = 10)}]$	$\mathbf{P}_{\texttt{min}}?[\texttt{F(S = 10)}]$ $\mathbf{P}_{\texttt{max}}?[\texttt{F(S = 10)}]$
$\mathbf{P}_{\texttt{DTMC}}$= 0.5 *P(Path2) + 0.5 *P(Path2) = 0.5 *(P(Path2) + P(Path2))	$\mathbf{P}_{\texttt{max}}$= P(Path1) $\mathbf{P}_{\texttt{min}}$=P(Path2)

DTMC provide an under-estimation of SET propagation by averaging all possible propagation probabilities as shown in Table 8.3.

In this paper, an MDP is constructed by parallel synchronization ($\|_S$) of the PAs of all components. The global states of the entire MDP are the interleaved states of each sub-module. At each state, a non-deterministic choice is done from a finite set of transitions as follows :

$$M_{MDP} = \{PA_1 \|_S PA_2 \|_S \dots \|_S PA_n\} \tag{8.1}$$

8.5.4 Proposed High Level Formal Analysis

The function *Analyze* in Algorithm 4 (line 24) implements the model checking process in which the state space of the MDP model of the design is exhaustively checked to verify the *satisfiability* of a property $p \in P$, returning a verification result $\varepsilon \in \Psi$. This analysis is performed using the *PRISM* probabilistic model checker. At any abstraction level, the maximum and the minimum probability that the injected SET will eventually reach a primary output (e.g., v) is evaluated by verifying the following properties :

$$\mathbb{P}_{\texttt{max}}?[\texttt{F(IS\_fault(v))}] \qquad \mathbb{P}_{\texttt{min}}?[\texttt{F(IS\_fault(v))}] \tag{8.2}$$

This analysis is performed for all vulnerable nodes. The output of this analysis is *OUT_RTL*, which is the set of all fault propagation probabilities. We assume that there is no correlation

between the signals in the design. Therefore, if a fault has to propagate through κ components to reach the output, then its probability to reach the output is calculated as follows :

$$\mathbb{P}_{max,min}(z) = \prod_{i=0}^{\kappa} PPr_i \tag{8.3}$$

where PPr_i is the fault propagation probability through each component in the propagation path of the injected fault. The values of $\mathbb{P}_{max}$ and $\mathbb{P}_{min}$ depend on the number of components in the propagation path and the fault propagation probabilities through each component, i.e., $\mathbb{P}_{max}$ and $\mathbb{P}_{min}$ can be different if there are different propagation paths with different probabilities. The maximum and the minimum probabilities that SETs will reach a primary output (e.g., O_v) of the design from all vulnerable nodes (i.e., Z) in its COI is calculated as follows :

$$\mathbb{V}ul_{min}(O_v) = \sum_{z=0}^{Z} Pe_z * \mathbb{P}_{min}(z) \tag{8.4}$$

$$\mathbb{V}ul_{max}(O_v) = \sum_{z=0}^{Z} Pe_z * \mathbb{P}_{max}(z) \tag{8.5}$$

where Pe_z is the injection probability at the vulnerable node z. $\mathbb{P}_{min}(z)$ and $\mathbb{P}_{max}(z)$ are evaluated based on Equation 8.3. This approach is accurate when the COIs of all outputs are mutually exclusive. In contrast, when COIs of different outputs overlap (or do not correspond to mutually exclusive events), then the probabilities for SET propagation to these outputs are correlated. In this case, if exact probabilities are required, then signal dependencies due to re-convergent fan-outs and/or correlated inputs have to be investigated. This leads to the path enumeration problem, where the number of paths that have to be enumerated independently can increase exponentially with the number of dependent re-convergent fanouts and correlated inputs [100]. Therefore, to avoid such complexity, COIs are assumed to be mutually exclusive as in Eq. 8.5 and Eq. 8.4, which can lead to safely over approximated probabilities. Finally, we estimate SER of the design using the vulnerability of all outputs as follows :

$$SER(comb) = \sum_{O \in comb} \mathbb{V}ol_{max}(O) \tag{8.6}$$

8.6 Implementation of the Proposed Framework

In this section, we illustrate the implementation of our proposed framework depicted in Fig. 8.1 at both gate and RTL levels. Our experiments were performed on a workstation with an Intel Core i7 running at 3 GHz and 24 GB RAM.

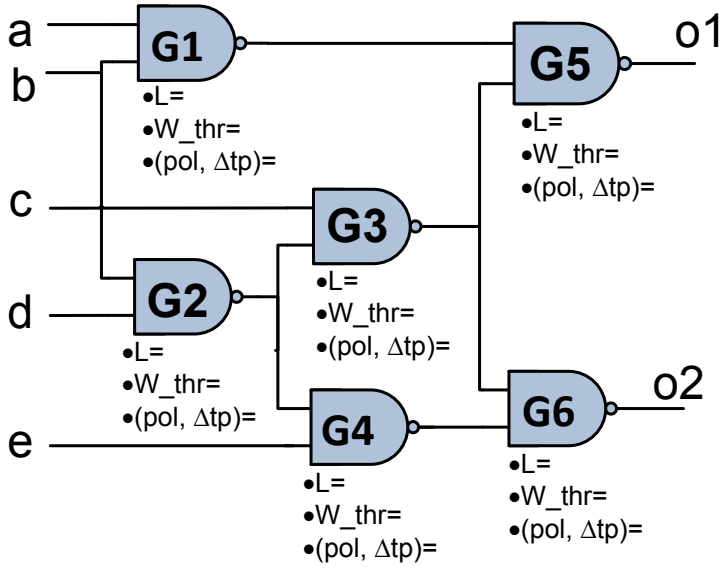
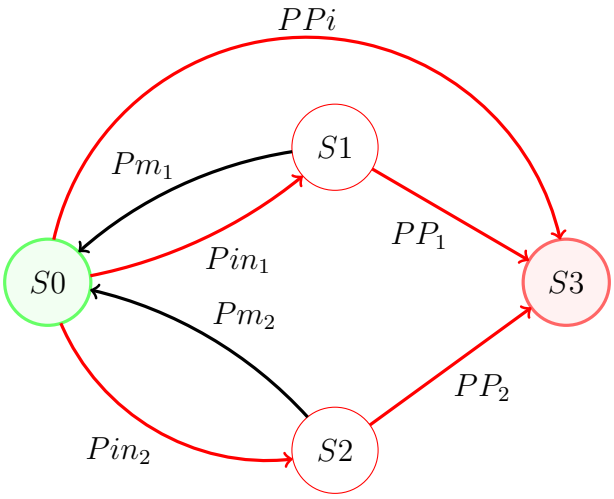


Figure 8.8 Annotated Gate Level Model of C17 (ISCAS85 Benchmark) Design.

The proposed model of SET propagation at gate level and RTL takes into account the impact of logical masking and SET width variation due to electrical masking and broadening phenomenon. In this work, we propose a mixed Boolean/linear integer encoding, in which logic variables and fault states are encoded as Boolean variables. The SET width is modeled as an integer variables. Thus, signals are composed of three elements : logic state (faulty or not) and SET polarity which are modeled as Boolean variables and the SET width which is modeled as an integer variable. The fault is considered to have positive polarity if a bit-flip changes the logical state of the signal from 0 to 1 (i.e., $0 \rightarrow 1 \rightarrow 0$) and a negative polarity if the signal changes from 1 to 0 (i.e., $1 \rightarrow 0 \rightarrow 1$).

8.6.1 Implementation at Gate Level

Starting from a gate level netlist, gate level analysis starts with the design COI reductions and then the mode of operation for all the gates is decided based on the injection scenario. The probabilistic models for SET propagation through each logic gate are developed based on their Transistor Propagation Tables (TPTs). In this paper, we adapted the TPTs developed in [111]. For example, consider the gate level netlist of the C17 benchmark shown in Fig. 8.8. Based on the TPT of each gate (based on its size and fan-out) is annotated with its width threshold (i.e., W_{thr}) and Δtp, which is the difference between tp_{LH} and tp_{HL}. These details are then utilized to build the probabilistic SET propagation behavior through each gate in this design. Let us assume we modeling 2-input NAND gate with in_1 and in_2 being



if $((in_1 = \text{SET}) \mid (in_2 = \text{SET}))$ & $(pol_f = \text{Pos})$ & $(W_f \geq W_{thr})$
  $\mathbf{P}(out = \text{SET}) = PP_1 \cdot Pin_1$; % For in_1
  $\mathbf{P}(out = \text{SET}) = PP_2 \cdot Pin_2$; % For in_2
  $W_f = W_f + \Delta tp$;
  $pol_f = not(pol_f)$;
else if $((in_1 = \text{SET}) \mid (in_2 = \text{SET}))$ & $(pol_f = \text{Neg})$ & $(W_f \geq W_{thr})$
  $\mathbf{P}(out = \text{SET}) = PP_1 \cdot Pin_1$; % For in_1
  $\mathbf{P}(out = \text{SET}) = PP_2 \cdot Pin_2$; % For in_2
  $W_f = W_f - \Delta tp$;
  $pol_f = not(pol_f)$;
else if $(((in_1 = \text{SET}) \mid (in_2 = \text{SET}))$ & $(W_f < W_{thr}))$
  $W_f = 0$; % Fault is Electrically masked
  $\mathbf{P}(out = \text{SET}) = 0$;

Figure 8.9 Probabilistic Model of SET Propagation Through a 2-input NAND Gate. PP_i is the Injection Probability for a NAND gate. PP_1 and PP_2 are the Propagation Probabilities for an SET Propagating Through in_1 and in_2, Respectively. Pin_1 and Pin_2 are the Probabilities an SET is Reaching in_1 and in_2, Respectively. Pm_1 and Pm_2 are the Probabilities That an SET is Masked While Propagating Through in_1 and in_2, Respectively.

the inputs and *out* being the output. The finite transition PA $(S; \overline{s}; P)$ for any 2-input NAND gate is depicted in Fig. 8.9. S is the set of states $S = (S0; S1; S2; S3)$, $\overline{s}$ is the initial state $\overline{s} = S0$. P is a transition probability matrix, $P_{i,j}$, such that $P_{S_0,S_1} = Pin_1$, $P_{S_0,S_2} = Pin_2$, $P_{S_0,S_3} = PPi$, $P_{S_1,S_3} = PP_1$, $P_{S_2,S_3} = PP_2$, $P_{S_1,S_0} = Pm_1$, $P_{S_2,S_0} = Pm_2$. Starting from the initial state S_0 (in_1 & in_2 are not faulty) there are two main scenarios for SETs :

— *Injection scenario* : if an SET is injected internally in the gate, then the next state is S_3 (output is faulty) with the injection probability PP_i. This probability is related to actual size of the gate, number of sensitive nodes, and the fanout of the gate which

are extracted based on the TPT.

— *Propagation scenario* : if an SET is propagating through the gate inputs, then the next state can be either S_1 (SET reached in_1) or S_2 (SET reached in_2), with probabilities Pin_1 and Pin_2, respectively. At this point, an SET can be masked i.e., go back to the error free state (S_0) with the probabilities Pm_1 and Pm_2 for in_1 and in_2, respectively. These transitions reflect the impact of both logical and electrical masking. For a NAND gate, an SET at one input is logically masked if the other input is 0. Moreover, if the width of the SET (W_f) at the input is less than the threshold (W_{thr}) of the gate, then it is electrically masked as shown in Fig 8.9. On the other hand, an SET can propagate and reach the output (S_3) with the propagation probabilities PP_1 and PP_2 for in_1 and in_2, respectively. In this case, a NAND gate broadens the width of a positive SET (in average by Δtp) and attenuate the width of a negative SET (in average by Δtp) as shown in Fig. 8.9.

When analyzing the SET propagation from different gates in the C17 circuit, this PA (shown in Fig. 8.9) is used to construct the behavior of the gates which operate in *Injection* and *Propagation* modes. Next, the MDP model of SET propagation through the design is constructed by parallel composition of gates PAs as explained in Section 8.5. PRISM is then employed to model this MDP as *Multi-Terminal BDDs (MTBDDs)* and to exhaustively analyze the probability of SET propagation from each vulnerable node to each primary output by verifying the following set of properties :

— P_{max} =?$[F((n_i = SET)\&(O_j = SET))]$: *If an SET is injected at node n_i, then what is the maximum probability that this SET will eventually propagate and reach output O_j ?*

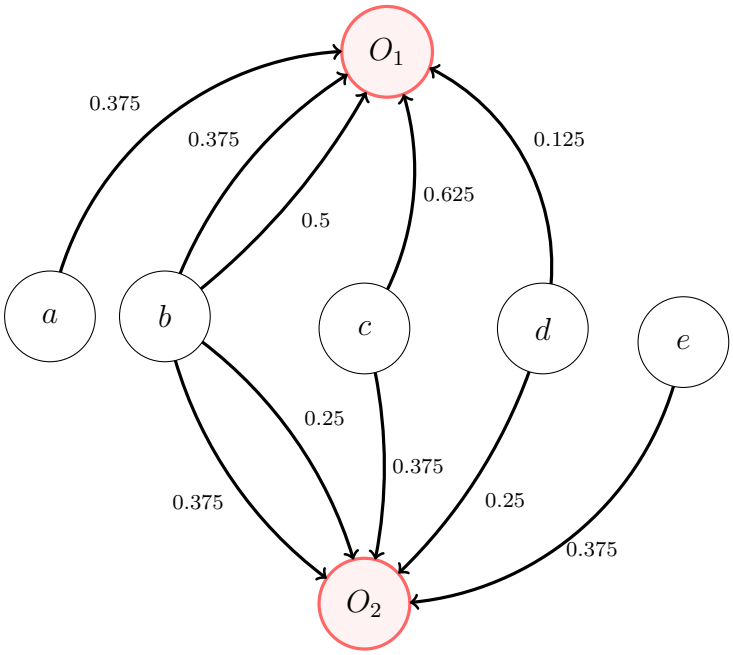
— P_{min} =?$[F((n_i = SET)\&(O_j = SET))]$: *If an SET is injected at node n_i, then what is the minimum probability that this SET will eventually propagate and reach output O_j ?*

If the injected SET has more than one propagation path to reach the output, then P_{max} and P_{min} provide the probabilities for the worst and the best propagation paths, respectively. As an example on this analysis, SET propagation probabilities for all vulnerable nodes in the C17 benchmark (shown in Fig. 8.8) are analyzed. Results are depicted in Table 8.4. If an SET propagates through node b, then it can reach O_1 with a maximum probability of 0.5 and a minimum probability of 0.375 and it will reach O_2 with a maximum probability of 0.375 and a minimum probability of 0.25.

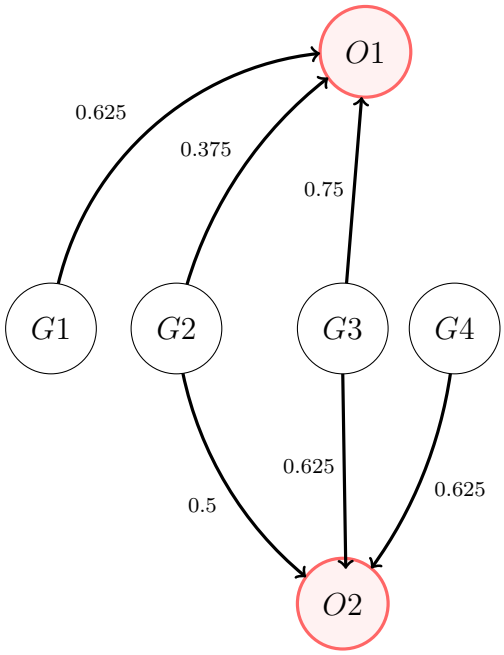
As explained in Section 8.3, these tables can be utilized to accurately model the SET propagation probabilities of this design at RTL i.e., RFs. For instance, if the C17 is a part of a larger design and an SET is injected somewhere outside the C17 then the *PA* depicted in

Table 8.4 Characterized Benchmark Circuits at Gate Level

	O1		O2	
Node	Pmin	Pmax	Pmin	Pmax
a	0.375	0.375	0	0
b	0.375	0.5	0.25	0.375
c	0.625	0.625	0.375	0.375
d	0.125	0.125	0.25	0.25
e	0	0	0.375	0.375
G1	0.625	0.625	0	0
G2	0.375	0.375	0.5	0.5
G3	0.75	0.75	0.625	0.625
G4	0	0	0.625	0.625



(a) PA for *Propagation* Mode



(b) PA for *Generation* Mode

Figure 8.10 Utilizing the Gate Level Table to Construct the Probabilistic Automata for SET Propagation for the C17 Benchmark Design.

Fig. 8.10(a) is constructed for the C17 when it is in the *propagation* mode. In other words, *PA* depicted in Fig. 8.10(a) describes the SET propagation probabilities through the primary inputs (*a, b, c, d,* and *e*). The *PA* depicted in Fig. 8.10(b) is constructed for the C17 when it is in the *Generation* mode i.e., an SET is injected internally in one of its gates (*G1, G2, G3,* and *G4*). For *G5* and *G6* the SET probabilities are 1 (because they are directly connected to the output).

Similarly, several combinational benchmark designs have been characterized such as basic adders and muxes. We have observed that the size of the MTBDD and the complexity of the gate level analysis is highly effected by the number of possible propagation paths (i.e., number of fan-outs and re-convergent gates in the propagation path). Therefore, this analysis is performed for moderate size designs. Generating gate level tables for the GLFs is a one-time effort that can be done offline.

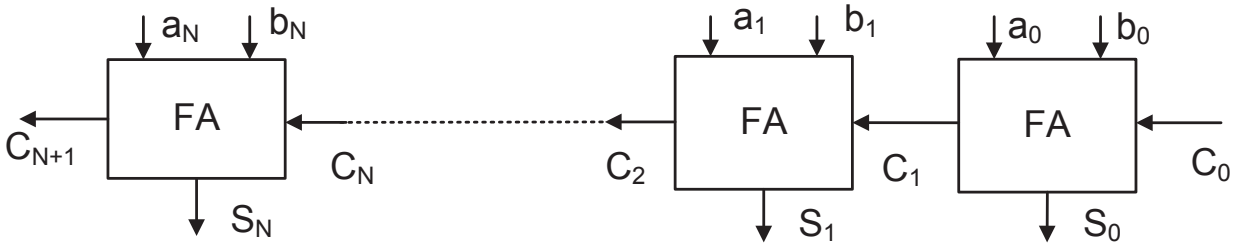


Figure 8.11 RTL of N-bit RCA and its SET Propagation Probabilities.

8.6.2 Implementation at RTL

In this section, we demonstrate how effectively SET propagation probabilities can be computed directly at RTL. Similar to the gate level analysis, this analysis starts with COI reduction of the model. Then, the modes of operation for all the components are decided based on the injection scenario and PAs are constructed for all components from their gate level tables, as explained in Section 8.3. Exhaustive analysis is performed over the MDP model to investigate SET propagation probabilities (i.e., P_{max} and P_{min}) from each component to each primary output.

As a first case study, our SET propagation analysis at RTL was performed on the Ripple Carry Adder (RCA) circuit. The RTL structure of a N-bit RCA is shown in Fig. 8.11, which is basically a chain of identical full adders. A N-bit RCA has $2N$ primary inputs and $N + 1$ primary outputs. Moreover, it can be observed that an SET which is present at one full adder can propagate to only one other full adder through only one propagation path which is the carry path.

In order to analyze any size RCA, the analysis of only one full adder at the gate level is required. The gate level representation of a full adder (depicted in Fig. 8.12) is analyzed as explained in Section 8.6.1. The results of this analysis are shown in Table 8.5, which are generated under the assumptions that the injection probability for an SET at any node is 1 and that all primary inputs are equally probable. It can be observed that the sum output

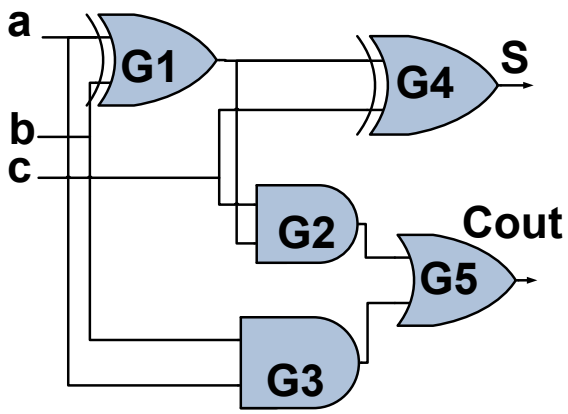


Figure 8.12 Gate Level Structure of the Analyzed Full Adder.

Table 8.5 SET Propagation Probabilities for Full Adder

	C		S	
Node	Pmin	Pmax	Pmin	Pmax
a	0.75	0.75	1	1
b	0.75	0.75	1	1
cin	0.5	0.5	1	1
G1	0.375	0.375	1	1
G2	0.75	0.75	0	0
G3	0.75	0.75	0	0

(S) of a full adder is very vulnerable as the SET propagation probabilities equal to one for all the vulnerable nodes in its COI. The propagation probabilities for the carry output are smaller and vary based on the injection scenario.

At RTL, if the SET is injected at one full adder, then this adder is in the *generation* mode. All the full adders before that full adder are in the *error-free* mode and all the full adders after that full adder are in the *propagation* mode. For example, in Fig. 8.13, an SET is injected at FA_0 i.e., it is in the *generation* mode. All other FAs ($FA_1 - FA_N$) are in the *propagation* mode. The SET probabilities reported in Table 8.5 are used to generate PAs for the *generation* and the *propagation* mode of the FA.

Fig. 8.14 reports the results of RTL analysis of a N-bit RCA. Fig. 8.14(a) depicts the SET propagation probabilities when it is injected at all the nodes in FA_0 (*c0, a0, b0, g1, g2,* and *g3* as shown in Fig. 8.12). Each curve represents the change in the SET probabilities while propagating from a certain node in FA_0 to the outputs of the RCA ($S_1, S_2, \ldots, S_N$ as

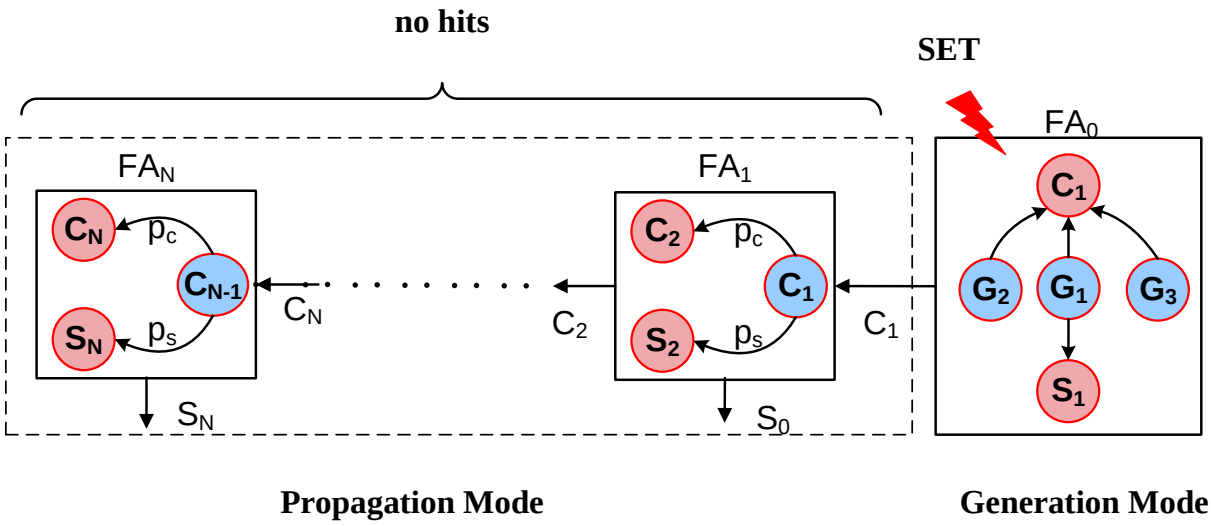
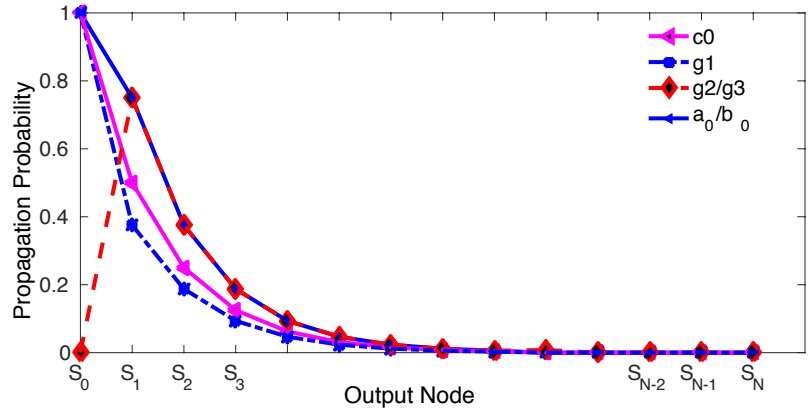


Figure 8.13 Modeling of SETs Propagation Probabilities in a N-bit RCA at RTL Based on the Injection Scenario.

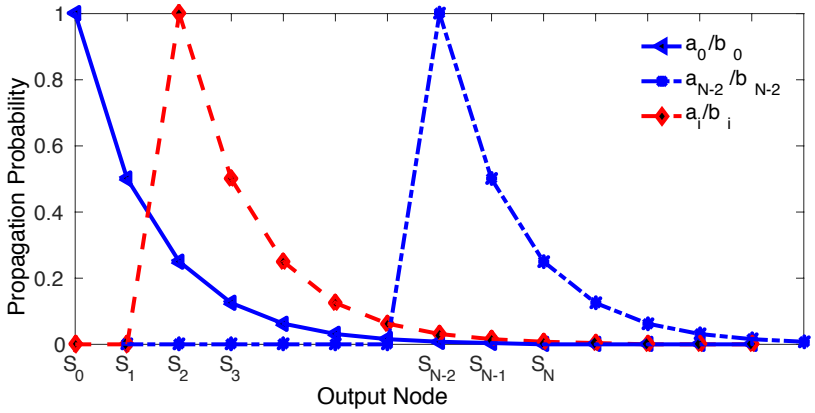
shown in Fig. 8.13). It can be observed that SET propagation probabilities vary based on the injection scenario for the next four stages. After that, SET propagation probabilities become almost the same for all the injection scenarios and is near to zero.

Fig. 8.14(b) depicts the results for injecting SET at the primary inputs (a_i/b_i) of different FAs. Each curve represents the probabilities of SET propagation from the full adder where it is injected to the outputs of the RCA ($S_1, S_2, \ldots, S_N$ as shown in Fig. 8.13). The same SET propagation probabilities are shifted to the stage of the full adder where the SET is injected. Due to the structure of the RCA, SET propagation probabilities are the same for any full adder from where it is injected to the Nth-stage. The proposed RTL modeling of the full adder allows us to model and analyze any adder implementation such as RCA, Carry Save Adder, and Carry Select Adder (CSA).

The 4-Bit ALU/function generator (74181 benchmark) depicted in Fig. 8.15 was also exhaustively analyzed. As explained before, at RTL one component can be in the *Generation* mode at a time, where we analyze SET propagation for all vulnerable nodes of this component, then another component is switched to the *Generation* mode. For example, in Fig. 8.15, if $M1$ is in the *Generation* mode, then $M2$ is in the *Error-Free* mode and both $M3$ and $M4$ are in the *Propagation* mode. The results of this analysis, which are the minimum and the maximum SET propagation probabilities from each vulnerable node to each primary output are depicted in Table 8.6. Based on these results, the contribution of each vulnerable node



(a)



(b)

Figure 8.14 The Results of the RTL Analysis of SET Propagation Probabilities of a N-bit RCA.

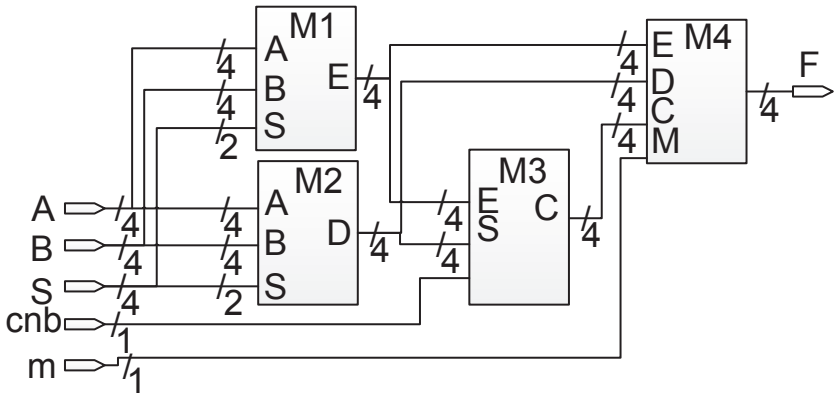


Figure 8.15 RTL Structural of the 4-bit ALU Circuit.

and each component to the failure of the design can be characterized. For example, as shown in Table 8.6 node $z20, z21$, and $z22$ have the least contribution to the failure of the 4-bit ALU, because if an SET is generated at these nodes, then it can only propagate to F_3 with maximum probability 0.0903. Such information can be very helpful for designers when applying some soft error tolerance techniques based on hardware redundancy such as TMR. In this experiment, it is assumed that the injection probabilities (Pe) for all vulnerable nodes are equiprobable and equal to $1/(number\ of\ nodes)$. The values of $\mathbb{V}ul_{min}$ and $\mathbb{V}ul_{max}$ are reported in the last row of Table 8.6. Therefore, the output nodes can be ranked based on their average vulnerability to soft error as $F_2 \rhd F_1 \rhd F_3 \rhd F_0$ where F_2 is the most vulnerable. These results can be used as a measure of the fault observability of each output. Such information can be very helpful for designers when applying a soft error tolerance techniques which is based on retiming. The SER of the design can be computed based on Eq. 8.6 which is equal to 0.679.

8.7 Discussion

At gate level, different formal methods based techniques have been proposed to investigate fault propagation conditions. In the closest related work [5, 4], the authors employ formal methods to identify one CIC for each injection scenario. A CIC is basically an input vector that allows propagation of a SET to an output. In [4, 5], it is assumed that the SET propagation probability is equal to the probability of having the generated CIC at the input of

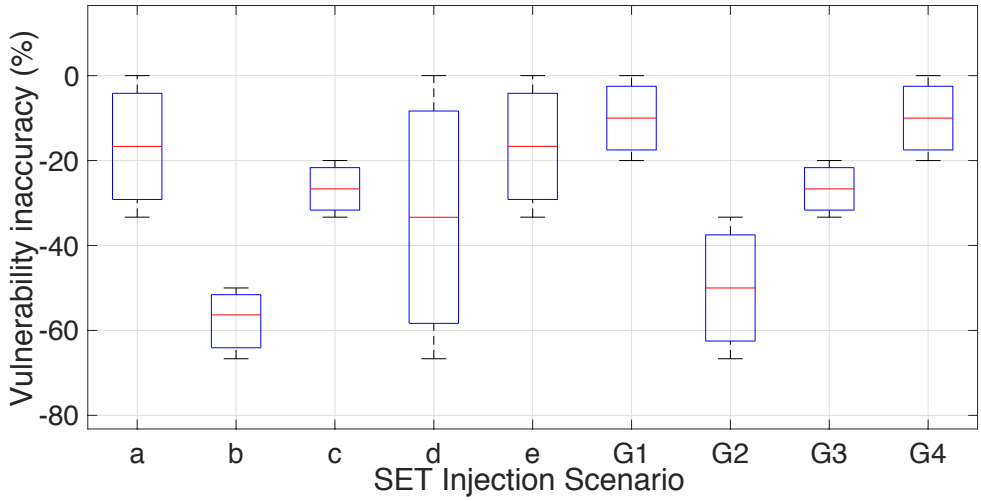


Figure 8.16 Inaccuracy in the Evaluation of the SET Propagation Probabilities in Related Works [4, 5] for the C17 Benchmark.

Table 8.6 Propagation Probabilities for a 4-bit ALU Circuit.

		F0		F1		F2		F3	
		min	max	min	max	min	max	min	max
M1	z1	0	.875	0	.156	0	.09	0	.06
	z2	0	0	0	.875	0	.304	0	.176
	z3	0	0	0	0	0	.75	0	.14
	z4	0	0	0	0	0	0	.5	.5
	z5	0	.875	0	.156	0	.09	0	.06
	z6	0	0	0	.875	0	.304	0	.176
	z7	0	0	0	0	0	.75	0	.14
	z8	0	0	0	0	0	0	.5	.5
M2	z9	0	.375	0	.094	0	.043	0	.025
	z10	0	0	0	.375	0	.16	0	.025
	z11	0	0	0	0	0	.304	0	.025
	z12	0	0	0	0	0	0	.25	.25
	z13	0	.375	0	.094	0	.043	0	.025
	z14	0	0	0	.375	0	.16	0	.025
	z15	0	0	0	0	0	.304	0	.025
	z16	0	0	0	0	0	0	.25	.25
M3	z17	0	0	.438	.438	0	0	0	0
	z18	0	0	0	0	.175	.175	0	0
	z19	0	0	0	0	.281	.28	0	0
	z20	0	0	0	0	0	0	.09	.09
	z21	0	0	0	0	0	0	.09	.09
	z22	0	0	0	0	0	0	.09	.09
M4	z23	1	1	0	0	0	0	0	0
	z24	0	0	1	1	0	0	0	0
	z25	0	0	0	0	1	1	0	0
	z26	0	0	0	0	0	0	1	1
	z27	1	1	0	0	0	0	0	0
	z28	0	0	1	1	0	0	0	0
	z29	0	0	0	0	1	1	0	0
	z30	0	0	0	0	0	0	1	1
$\mathbb{V}ul$		.067	.15	.081	.181	.082	.192	.126	.156

design. However, this probability provides an under-estimation of the actual fault propagation probability. This is mainly because in practice different CICs can allow the propagation of a fault. To demonstrate how much inaccuracy can be introduced with this assumption, we applied the technique proposed in [4] on the C17 circuit. We compare those vulnerabilities obtained with the values evaluated using the proposed technique. Fig. 8.16 shows that estimating propagation probability using a single CIC causes 27.4 % inaccuracy in the evaluated vulnerabilities, and the maximum deviation reaches around 67 %. For example, as shown in

Fig. 8.16, when analyzing SET injected at gate *G2* (as proposed in [4, 5]) then the inaccuracy in the result varies from -33 % to -68 %.

It is important to note that the techniques in [4, 5] provided accurate results (0% inaccuracy) as the case for *a, d, e, G1* and *G4* only in two scenarios : 1) if there is no propagation for the injected SET to reach a certain output ; or 2) the injected SET can propagate under only one input vector. Therefore, the only way that the technique proposed in [5] can lead to the same level accuracy as our technique is if it generates all the CICs. Thus, in that case the SET propagation probability will be equal to $P(\bigcup_{all} CIC)$. However, for large designs it is not practical nor possible to generate all the CICs for all the injection scenarios. Further detailed comparison between these techniques and the proposed framework is shown in Table 8.7.

Table 8.7 Comparison Between the Proposed Framework and the Contemporary Techniques at Gate Level

	RASVAS [4]	**MDG [5]**	**This work**
Logical Masking	Included	Included	Included
Electrical Masking	Not Included	Included	Included
SET width Attenuation	Not Included	Enumerated Data type – 4 stages	Integer data type–for n stage
SET width Broadening	Not Included	Not Included	Included
Formal Tool Used	MDG model checker	MDG model checker	PRISM model checker
Results of the analysis	CIC for each fault	CIC for each fault	Propagation Probability for each fault

At RTL, our results demonstrate that the proposed framework outperforms contemporary formal verification techniques ([112, 56]) in the following aspects :

1) Accuracy : Our framework provides more accurate results than contemporary methods. This is mainly because the proposed probabilistic analysis explores all possible transitions of the MDP model. By contrast, contemporary techniques (such as [6]) typically explore a limited number of input vectors (random input assumption) to evaluate SET propagation probabilities, thus providing an incomplete analysis. Therefore, such analysis can provide inaccurate estimation of the SET propagation probabilities. For instance, based on the tech-

nique reported in [112], SET propagation probabilities for any size RCA is 1. Results in Fig. 8.11 demonstrate that the probability of SET propagation is not 1 for all outputs. Moreover, probability of SET propagation through more than 29 FAs is almost zero ($< 1.38 \times 10^{-17}$). Furthermore, contemporary techniques (such as [4]) model SET propagation at the RTL using the CICs computed at the gate level. To see how much inaccuracy can be introduced from relying only on the gate level CICs, we compare the probabilities reported in [4] with the values evaluated using the proposed framework for the 4-bit ALU. Fig. 8.17 shows using CICs on average causes up to 55.84 % inaccuracy.

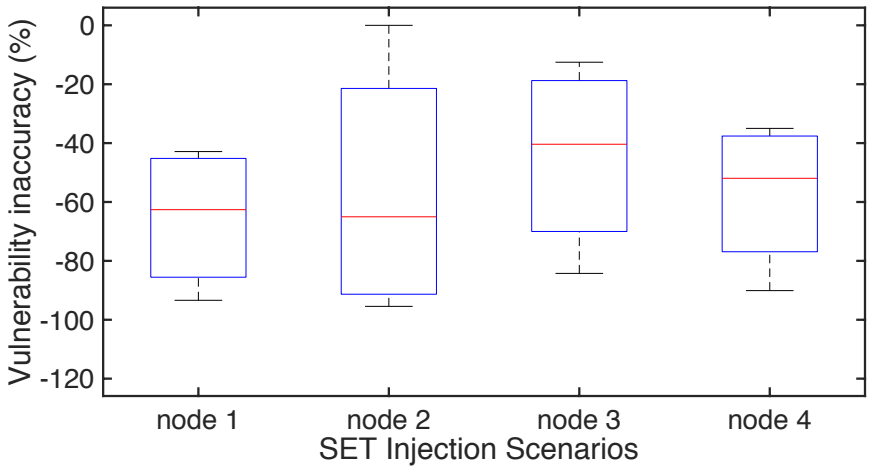


Figure 8.17 Inaccuracy in the Evaluation of the SET Propagation Probabilities in Related Works [6, 4] by Relying Only on the Gate Level CICs for the 4-bit ALU Circuit.

2) Scalability : State-of-the-art techniques are limited due to the state explosion problem when dealing with large designs. In Table 8.8, we compare our methodology with the DTMC RTL approach (proposed in [56]) by considering different sizes of RCA adders as a case study. In [56], each input vector is mapped to a unique state. The corresponding Markov model has 2^{2N} states. Thus, *PRISM* runs out of memory while constructing RTL models for adders ($>$ 14-bit). Therefore, in [56] the authors try to partition large RTL designs into smaller sized blocks to minimize the total runtime. However, even with the proposed partitioning, the verification time is growing exponentially with the design size. For example, the time required to analyze a 64-bit RCA with partitioning is 223.58 *sec* as shown in Table 8.8. With the proposed framework, it is possible to analyze a 256-bit RCA within 0.601 *sec*. The analysis time in Table 8.8 is the time required to evaluate SET propagation for any injection scenario.

Table 8.8 Digital Designs Analyzed at RTL

RTL Design	PI	RTL Cells Count	DTMC RTL [56]				Our Method	
			Without Partitioning		With Partitioning			
				Time (sec.)		Time (sec.)		Time (sec.)
16-bit	33	16	$\surd$	2.141	$\surd$	1.24	$\surd$	0.016
32-bit	65	32	$\times$	$> 10hrs$	$\surd$	41.84	$\surd$	0.102
64-bit	129	64	$\times$	$> 10hrs$	$\surd$	223.58	$\surd$	0.119
128-bit	256	128	$\times$	-	$\times$	-	$\surd$	0.308
256-bit	512	256	$\times$	-	$\times$	-	$\surd$	0.601

$\times$: with this technique *PRISM* runs out of memory while building the model.
$\surd$: with this technique *PRISM* builds the Markov model and verifies the property.

8.8 Conclusion

In this paper a hierarchical framework to quantitatively model, analyze, and estimate the effects of soft errors at different abstraction levels is proposed. This is achieved by reducing the design size, utilizing propagation tables generated from lower abstraction level models, and adapting a probabilistic model checker. Proposed framework achieves significant speedup compared to statistical fault injection and contemporary formal techniques with more precise estimated vulnerability.

At gate level, the proposed framework was implemented on different combinational benchmarks. SET propagation probabilities at gate level are utilized to model SET propagation through larger designs at RTL. Results demonstrate that our proposed framework can handle larger and more complex designs (e.g., 256-bit RCA), while the best previously reported techniques run out of memory for 14-bit RCAs.

CHAPTER 9 GENERAL DISCUSSION

Recent radiation ground testing campaigns of digital designs have demonstrated that the probability for Single Event Transients (SETs) propagation is increasing in advanced technologies. Classical models have been found to underestimate the Soft Error Rate (SER) due to SETs. This thesis addresses the verification problem using a unified approach, which utilizes new mechanisms to bridge the gap between design abstraction levels. In the techniques proposed in this thesis, the verification process starts as early as possible, while providing the flexibility to move across different abstraction levels.

9.1 Discussion of the Proposed Transistor Level Analysis

SETs become prevalentas as geometric dimensions scale down [11]. As our results demonstrate, SET characteristics are dependent on the propagation paths, the input patterns, the strike time, the converging node and the diverging node. Therefore, it is predicted that all these factors will continue to impact SET characteristics with technology scaling. New insights of the impact of the PIPB phenomenon on SET propagation are provided.

In order to bridge the gap between the different analysis of SET propagation at different abstraction levels, based on our understanding of SET propagation behavior at transistor level, I proposed different methods to improve the usability of the results of this analysis. The main constraints related to the design structure, SET timings, and SET characteristics are abstracted.

9.2 Discussion of the Proposed Gate Level Analysis Methods

One of the main challenges in analyzing SET propagation at higher abstraction level is to accurately model all the SET propagation scenarios observed at the transistor level. For example, contemporary techniques (such as [48], [6]) are not sufficiently accurate, as these techniques omit the possibility of SET broadening while propagating, which is significant because a relatively short pulse, just sufficient to propagate, can become arbitrarily long, provided the existence of a sufficient logic depth. Moreover, state-of-the-art techniques at gate or higher abstraction levels analyze the susceptibility of digital circuits to soft errors by only modeling the masking effects that can prevent SETs from propagating [11]. Nonetheless, existing state-of-art techniques are unable to model the effects of propagation paths characteristics, re-converging paths, and input patterns on SET characteristics at high abstraction

levels. Deficiencies in conventional models lead to inaccurate estimation of soft error rates (SERs). Hence, there is a growing need to better abstract and characterize SET propagation at gate and higher abstraction levels.

During my Ph.D., I investigated possible solutions to address these shortcomings. Table 9.1 depicts a comparison between the proposed modeling and analysis techniques of SETs at gate level using model checking with MDG [111] and Satisfiability verification using SMTs [113].

In all these techniques, details from transistor level are utilized to accurately model SET characteristic width variation at gate level. In each technique, I developed different methods to better abstract SET propagation scenarios and to characterize the impact of pulse polarity, the logic structure, and the input patterns on the propagating SET width. These methods are mainly different based on the modeling capabilities for the adapted formal formulation. As discussed in details in Chapter 5, based on certain assumptions, the proposed MDG modeling include the impact of different masking effects using the abstract and the enumerated data types. However, I have observed that such modeling is not very efficient to model complex behaviors such as SET width variation and temporal masking using only enumerated data types as it is not possible to enumerate all possible values of all the variables in the model. Furthermore, I faced a major scalability issue when trying to implement this technique on complex combinational designs such as multipliers. For example, the MDG model checker runs out of memory when trying to analyze an 8-bit array multiplier.

Hence, I started looking for a better formal formulation of these complex phenomena at gate level. I proposed a new formulation of SETs propagation as a Satisfiability problem by utilizing Satisfiability modulo theories. Using different SMT theories (e.g., the theory of real numbers, the theory of integers, and the theory of difference logic) I was able to model all the making effects and the SET width variation. The proposed modeling relies on details extracted from the pre-characterized TPTs of the technology node and the gates timing extracted from the layout. An SMT-based exhaustive analysis of SET propagation is proposed. As shown in Table 9.1, the implementation of the proposed methodology on different combinational designs shows its applicability and scalability. For instance, it can analyze complex arithmetic circuits such as a 128-bit multiplier in about 70 minutes, while existing techniques fail to handle multipliers larger than 32 bits. Our results demonstrate that the complexity of the proposed analysis varies for different faults. In other words, while most SET injection scenarios are solved in a reasonable amount of time, some SETs are harder to analyze than others. The complexity of the proposed non-functional analysis depends on the number of paths that lead from a node to the outputs. Furthermore, results indicate that the performance greatly varies by the used SMT solver. We implemented the proposed SMT

modeling on different SMT solvers in order to compare the performance of each and decide on an optimal verification technique and solver. The solvers we used are *Z3* [58], *Yices* [59], *Mathsat* [60], and *CVC4* [61]. *Yices* performance was found to be much better than that of other solvers. These results agree with the results published in the SMT competition [114], since *Yices* is known to have the best performance in the verification of *quantifier free theory of arrays* and the theory of *linear integer arithmetic*. However, if the results of this analysis are to be used to estimate the vulnerability of the design then one problem rises when dealing with large designs is generating the exact probabilities. This is mainly because (as explained in Chapter 6) the propagation probabilities of the injected SET are computed based on the solutions generated by the SMT solvers. Therefore, for exact probabilities, all solutions are needed. However, this present a complex issue when dealing with large designs with large number of primary inputs. To handle this issue, we can utilize of *all-solutions* SMT solvers to provide an estimate of the actual number of solutions.

Table 9.1 Detailed Comparison Between the SET Gate Level Analysis Techniques Proposed in This Thesis

	Accuracy of the Model				**Model Size**	**Scalability**	**Analysis Results**	**Results Accuracy**	
	Includes Logical Masking ?	Includes Electrical Masking ?	Includes Temporal Masking ?	Models SET Width Variation ?				For Each Injection Scenario	Estimated SER
Analysis Using MDG	YES	YES, very approx. model	NO	NO	Large	Not Scalable	One input assignment	Accurate	High Over/Under approx.
Analysis Using SMTs	YES	YES	YES	YES	Small	Highly Scalable	Multiple Soultions	Accurate	Moderate Over or under approx.

9.3 Discussion of the Proposed RTL Analysis Methods

The analysis of SET propagation at higher levels of abstraction is key to manage the complexity of today's VLSI chips. In this thesis, I have proposed different modeling and analysis techniques of SET propagation at RTL based on MDGs and Markov decision process.

As demonstrated in Chapters 7, 8, these RTL techniques outperform the existing modeling and analysis techniques. In Table 9.2, a comparison is made between all the techniques I proposed to analyze SETs propagation at RTL namely :

1. RTL modeling and analysis based on multiway decision graphs (MDGs) and gate level propagation tables of the CICs. This is referred to in Table 9.2 as *MDG*.

2. RTL probabilistic modeling and analysis based on Markov Decision Process (MDP) and gate level propagation tables of the CICs. This is referred to in Table 9.2 as *PMC1*.
3. RTL probabilistic modeling and analysis based on MDP and gate level characterization libraries developed by our probabilistic gate level analysis. This is referred to in Table 9.2 as *PMC2*.

By utilizing MDG, I developed a hierarchical abstraction and modeling approaches of SET propagation. The RTL model I built here relies on gate level propagation tables which I developed based on the proposed MDG gate level analysis in [111]. As explained before, in these tables, a CIC for each SET injection scenario is reported. In this analysis, two reduction techniques are used to significantly reduce the time and memory requirements required to model and analyze SET propagation at RTL : 1) the cone of influence reduction techniques proposed in [99] ; and 2) cross-level modeling of the RTL sub-component based on their mode of operation. For each injection scenario, reduced version of the design is built and analyzed using the invariant checking tool from the MDG tool-set. The reported results demonstrate that due to the proposed modeling, the CPU time and the memory requirements are reduced by more than 60%. The proposed analysis investigates SET propagation and returns a counterexample which reports a CIC that can propagate the injected SET to the output. The generated CICs are then utilized to estimate the SER of the design at the RTL. For complex RTL design each SET injection scenarios can have a large number of different CICs which allow its propagation. However, with the proposed MDG RTL analysis, it is not possible to generate multiple CICs for the same injection scenario. We have observed that this limitation led to large discrepancy (under-approximation or over-approximation) in the computed SER at RTL. Therefore, I worked on developing a hierarchical probabilistic framework to quantitatively model, analyze, and estimate the effects of SETs at RTL (PMC1 in Table 9.2). In this framework, I utilized the same gate level tables. Moreover, similar to the MDG based RTL analysis I utilized the COI reduction technique and the mode of operation cross-level modeling. In this work, the SET propagation at the RTL is modeled as DTMC model. This modeling allows the probabilistic analysis (using PRISM probabilistic model checking) of each injection scenario and estimate the overall design SER. With this framework a significant speedup was observed compared to the MDG RTL analysis (MDG in Table 9.2). Furthermore, the computed SERs based on this framework are more accurate than the SER computed based on the MDG RTL model. However, after further investigation, I concluded that there is still a certain amount of discrepancy (under-approximation or over-approximation) in the computed SERs. I observed that this inaccuracy is introduced at the component level by the gate level propagation tables. This is mainly because these tables reports only one CIC for each injection scenario in this component (i.e., one input vector

that allows the propagation of an SET to an output). However, it is possible to have many CICs for each injection scenario. The accumulative inaccuracy from all the sub-components in the SET propagation path can lead to high inaccuracy in the probability computed based on our MDP model.

Table 9.2 Detailed Comparison Between the SET RTL Analysis Techniques Proposed in This Thesis

	Multilevel Model	**Cross-level Model**	**Formal Model**	**Reduction**	**Results Accuracy**			**Scalability**	**Results Usability**	**Notes**
					For Each Injection Scenario	**SER**	**Results**			
MDG [115]	Underlying CICs from MDG gate analysis	YES	Multiway decision Graph	COI & Lower Details	Varies For Each Node	Weak	One RTL CIC	moderate	Low	Best for LOF
PMC1 [4]	Underlying CICs from MDG gate analysis	YES	Markov Decision Process	COI & Lower Details	Varies For Each Node	Good	Max and Min prob.	Moderate	Moderate	Best for LOF
PMC2 [116]	Underlying PPs from PMC gate analysis	YES	Markov Decision Process	COI & Lower Details	Accurate	Accura	Max and Min prob.	High	High	For LOF and HOF

It is important to note that the techniques in Table 9.2 (*MDG* and *PMC1*) can provide accurate results (0% inaccuracy) in three cases : 1) if it is not possible for the SET injected to reach the output ; 2) the injected SET propagates under only one CIC ; or 3) if theses techniques are able to generate all CICs that allow SET propagation. Thus, in that case the SET propagation probability will be equal to $P(\bigcup_{all} CICs)$. However, for large designs, it is not practical nor possible to generate all the CICs for all the injection scenarios. Therefore, in order to provide more accurate estimations using this multilevel and cross-level approach I developed new modeling at each abstraction level based on the fault space mapping. At each abstraction level, there is a certain number of faults which lead to design failure. Each fault at each abstraction level, which is considered as a Top Level Event (TLE), occurs due to faults at lower abstraction levels (i.e., Low Level Events (LLEs)). Each High Level Fault (HLF) can be mapped through a one-to-many mapping to its corresponding set of Low Level Faults (LLFs) realization, which is defined as a correlation group. Based on this concept, a new gate level modeling was developed based on transistor propagation tables. The PA for each gate is generated based on its mode of operation. Thereafter, an MDP of SET propagation through the design is constructed by parallel composition of gates PAs as

explained in Section 8.5. If the injected SET has more than one propagation path to reach the output, then P_{max} and P_{min} provide the probabilities for the worst and the best propagation paths, respectively. All these probabilities are characterized and made available to model SET propagation probabilities at higher abstraction levels. At RTL, *PMC2* approach models SET propagation is modeled as an MDP. The SET propagation probabilities can be computed and an accurate estimation of soft error rates can be developed based the results of the proposed RTL analysis. As shown in Table 9.2, this framework is the most efficient in comparison with *MDG*, *PMC1* and all other existing techniques.

CHAPTER 10 CONCLUSION

10.1 Conclusion

Soft errors, induced by radiation, have a growing impact on the reliability of CMOS integrated circuits. The progressive shrinking of device sizes in advanced technologies leads to miniaturization and performance improvements. However, ultra-deep sub-micron technologies are more vulnerable to soft errors. In this thesis, we propose a hierarchical multi-level methodology to model, analyze, and estimate Single Event Transients (SETs) propagation in combinational designs expressed at different abstraction levels (transistor to Register Transfer (RT) levels). Basic components are modeled and analyzed at low level and the results of this analysis are condensed into SET propagation tables. At high level, these tables are utilized to model the underlying probabilistic behavior of SET propagation. The methods proposed and explored in this thesis are validated through case studies and the reported results confirm their accuracy, scalability, and applicability.

Chapter 1 explained the context of this thesis, introduced the problem of soft errors due to SETs, and reviewed the main shortcoming of the existing techniques. The main objectives are identified and the research contributions presented in this thesis were highlighted.

In Chapter 3 a background is provided about the main sources of single event transients in digital circuits. Thereafter, formal verification methods utilized in this thesis to model and analyze SET propagation at high abstraction levels are introduced namely ; MDG model checking, probabilistic modeling checking using PRISM, and Satisfiability formulation based on SMTs. In Chapter 2, existing related SET propagation analysis at transistor level as well as at gate and RT levels are discussed.

Chapter 4 presented the article entitled "*New Insights Into the Single Event Transient Propagation Through Static and TSPC Logic*". In this chapter, we investigate the variations of SET characteristics while propagating due to propagation paths, input patterns, and pulse polarity in both static and TSPC logic. We demonstrate that these factors can aggravate the SET pulse broadening phenomenon. Worst and best propagation paths (WPPs and BPPs) were identified for the analyzed designs. New insights on the propagation induced pulse broadening (PIPB) phenomenon in different combinations of static and TSPC logic are reported. Moreover, timing constraints related to SET propagation in TSPC logic such as the strike time and clock period are identified. Our results demonstrate that SET pulses propagation can lead to Byzantine faults as they propagate through diverging paths.

Chapter 5 presented the paper entitled "*Modeling, Analyzing, and Abstracting Single Event Transient Propagation at Gate Level*". In this chapter, the proposed transistor level analysis is leveraged to offer a mechanism to abstract at gate level the PIPB effects, width attenuation, and electrical masking and their implications on the soft error rate. For example, the impact of the applied input pattern and the gate fan-out on the SET pulse width is abstracted using the Load and Input Combination Factor (LICF). At the gate level, we analyzed SET pulse propagation by utilizing the MDG model checker and the delay degradation model. We proposed a novel method to identify paths that can propagate SET pulse causing soft errors in digital designs. Finally, we proposed new gate level characterization libraries which can be used to accurately analyze SET pulse propagation and estimate the SER at the RTL.

Chapter 6 presented the paper entitled "*Efficient and Accurate Analysis of Single Event Transients Propagation Using SMT-Based Techniques*". In this chapter, the problem of SET propagation was formalized as an SMT problem. This model captures all the details related to : 1) all masking effects (logical, electrical, and temporal) ; and 2) variations in the SET characteristics (attenuation and broadening). By solving the SMT model of the design under specific assertions, the following results are obtained : a) the set of input vectors to be present at the primary inputs so that SET is not logically masked ; b) the window of vulnerability within the clock cycle and the minimum SET width such that it is not temporally and electrically masked. Thereafter, based on these results, the SET propagation probabilities and the Soft Error Rate (SER) of the design are estimated.

Chapter 7 presented the paper entitled "*Towards Formal Abstraction, Modeling, and Analysis of Single Event Transients at RTL*". In this chapter, a hierarchical formal modeling and analysis of SET propagation at register transfer level by introducing new abstraction and modeling of the underlying behavior of SET propagation using MDGs. Invariant checking tool from the MDG tool set is utilized to formally validate the SET propagation for each injection scenario which is designed to return a CIC that can propagate the injected SET to the output. In order to illustrate the practical utilization of our work, we have analyzed different RTL combinational designs. Experimental results demonstrate that the proposed MDG formulation significantly reduces the time and memory requirements to model and analyze SET propagation at RTL. For instance, the CPU time and the memory required are reduced by more than 60% which enabled the analysis of SET propagation through complex designs e.g., 16-bit multiplier.

Chapter 8 presented the article entitled "*Comprehensive Multilevel Probabilistic Analysis of Single Event Transients Propagation Induced Soft Errors*". In this chapter, our hierarchical probabilistic framework to quantitatively model, analyze, and estimate the effects of soft er-

rors at RTL is presented. First, for each injection scenario, the design is reduced based on the proposed reduction techniques and the propagation tables generated from lower abstraction level models. Then, SET propagation through the reduced design is modeled as a Markov decision process based on the probabilistic automatas of all the RTL sub-components. PRISM is adapted to analyze the probability of SET propagation for all vulnerable nodes. Furthermore, a new method to estimate the SER is proposed. Results demonstrate that the proposed framework achieves significant speedup compared to statistical fault injection and contemporary formal techniques with more precise estimated vulnerability. For example, with this framework, we were able to analyze larger and more complex designs (e.g., 256-bit RCA), while the best previously reported techniques run out of memory for 14-bit RCAs.

10.2 Future Work Directions

10.2.1 Layout-Based Multiple Events Transients (METs) SMT-based Analysis

As demonstrated in this thesis, the progressive miniaturization of device sizes in advanced technologies increases the probability for a high energy particle strike to cause a transient fault in several adjacent cells in a circuit resulting in Multiple Event Transients (METs) in combinational gates. Different radiation experiments (such as [117], [118]) demonstrated that a considerable fraction of the soft errors were contributed by single particle strikes in random logic results in METs. Additionally, the distribution of affected error sites and the number of affected cells depend on the target technology node, and the injected particle type, energy, strike angle, cell structure, cell size, cell capacitance which are not available at high abstraction levels.

The proposed framework in this thesis could be easily extended to include the impact of METs. First, both the design layout timing of the cells can be extracted as done in Chapter 6. The physically adjacent error sites are extracted from the circuit layout based on the sensitive area size. The SMT modeling and analysis proposed in Chapter 6 is then extended to include METs instead of SETs. It is expected that high accuracy and scalability will be observed when analyzing METs.

10.2.2 SMT-Based Reliability-Aware Synthesis

The results in this thesis demonstrate the applicability and efficiency of the proposed framework. Such framework can be very useful when developing mitigation techniques at different abstraction levels. For instance, in order to develop efficient mitigation techniques, the vulnerability of each node in the design is needed. Such vulnerability can be estimated using

our framework. A hierarchical reliability-aware synthesis framework to design combinational circuits at gate level for soft error tolerance with minimal area overhead can be proposed. The main idea is based on utilizing the results of our SMT-based analysis to harden sensitive cells, paths, or sub-circuits, whose SET propagation probability is relatively high, until the desired SER is achieved or a given area overhead constraint is met.

In this context, we already exploited the techniques presented in this thesis toward developing an efficient SETs reliability-aware synthesis framework. Initial steps of this directions are presented in [119], [120].

REFERENCES

[1] B Gill, N Seifert, and V Zia. Comparison of alpha-particle and neutron-induced combinational and sequential logic error rates at the 32nm technology node. In *2009 IEEE International Reliability Physics Symposium*, pages 199–205. IEEE, 2009.

[2] NN Mahatme, S Jagannathan, TD Loveless, LW Massengill, BL Bhuva, S-J Wen, and R Wong. Comparison of combinational and sequential error rates for a deep submicron process. *IEEE Transactions on Nuclear Science*, 58(6) :2719–2725, 2011.

[3] M. Bellido-D., J. Juan-C., A. Acosta, M. Valencia, and J. Huertas. Logical modelling of delay degradation effect in static cmos gates. In *IEEE Proc. Circuits Devices Syst.*, pages 107–117. IET, 2000.

[4] G. Bany Hamad et al. Efficient multilevel formal analysis and estimation of design vulnerability to single event transients. In *IEEE 21st International On-Line Testing Symposium (IOLTS)*, pages 1–6, 2015.

[5] G. Bany Hamad et al. Characterizing, modeling, and analyzing soft error propagation in asynchronous and synchronous digital circuits. *Microelectronics Reliability*, 55(1) :238–250, 2015.

[6] S. Shazli and M. Tahoori. Using boolean satisfiability for computing soft error rates in early design stages. *Microelectronics Reliability*, 50(1) :149–159, 2010.

[7] T. Zaremba et al. Radiotherapy in patients with pacemakers and implantable cardioverter defibrillators : a literature review. *Europace Journal*, pages 135–145, 2015.

[8] International technology roadmap for semiconductors. available :. `http://public.itrs.net`.

[9] Tanay Karnik and Peter Hazucha. Characterization of soft errors caused by single event upsets in cmos processes. *IEEE Transactions on Dependable and Secure Computing*, 1(2) :128–143, 2004.

[10] Shubu Mukherjee. *Architecture design for soft errors*. Morgan Kaufmann, 2011.

[11] Véronique F. C., Lloyd W M., and Pascale G. Single event transients in digital cmos a review. *IEEE Trans. Nucl. Sci.*, 60(3) :1767–1790, 2013.

[12] V. Ferlet-Cavrois et al. Investigation of the propagation induced pulse broadening (PIPB) effect on single event transients in soi and bulk inverter chains. *IEEE Trans. Nucl. Sci.*, 55(6) :2842–2853, 2008.

[13] V. Ferlet-Cavrois, P. Paillet, D. McMorrow, N. Fel, J. Baggio, S. Girard, O. Duhamel, J. Melinger, M. Gaillardin, J. Schwank, et al. New insights into single event transient propagation in chains of inverters—evidence for propagation-induced pulse broadening. *IEEE Trans. Nucl. Sci.*, 54(6) :2338–2346, 2007.

[14] Pascale Gouker, Jim Brandt, Peter Wyatt, Brian Tyrrell, Anthony Soares, Jeffrey Knecht, Craig Keast, Dale McMorrow, Balaji Narasimham, Matthew Gadlage, et al. Generation and propagation of single event transients in 0.18-< formula formulatype=. *IEEE Trans. Nucl. Sci.*, 55(6) :2854–2860, 2008.

[15] D.G. Mavis and P.H. Eaton. Soft error rate mitigation techniques for modern microcircuits. In *Reliability Physics Symposium Proceedings, 2002. 40th Annual*, pages 216–225. IEEE, 2002.

[16] David G Mavis and Paul H Eaton. Seu and set mitigation techniques for fpga circuit and configuration bit storage design. In *Proceedings MAPLD Conference*, 2000.

[17] Paul E Dodd, Marty R Shaneyfelt, James A Felix, and James R Schwank. Production and propagation of single-event transients in high-speed digital logic ics. *IEEE Trans. Nucl. Sci.*, 51(6) :3278–3284, 2004.

[18] V Ferlet-Cavrois, P Paillet, M Gaillardin, D Lambert, J Baggio, JR Schwank, G Vizkelethy, MR Shaneyfelt, K Hirose, EW Blackmore, et al. Statistical analysis of the charge collected in soi and bulk devices under heavy lon and proton irradiation—implications for digital sets. *IEEE Trans. Nucl. Sci.*, 53(6) :3242–3252, 2006.

[19] Lloyd W Massengill and Paul W Tuinenga. Single-event transient pulse propagation in digital cmos. *IEEE Trans. Nucl. Sci.*, 55(6) :2861–2871, 2008.

[20] G. Wirth et al. Single event transients in logic circuits load and propagation induced pulse broadening. *IEEE Trans. Nucl. Sci.*, 55(6) :2928–2935, 2008.

[21] Paul W Tuinenga and Lloyd W Massengill. Circuit modeling of single-event transient pulse stretching in digital cmos. *IEEE Trans. Nucl. Sci.*, 56(6) :3165–3171, 2009.

[22] Matthew J Gadlage, Jonathan R Ahlbin, Vishwanath Ramachandran, Pascale Gouker, Cody A Dinkins, Bharat L Bhuva, Balaji Narasimham, Ronald D Schrimpf, Michael W McCurdy, Michael L Alles, et al. Temperature dependence of digital single-event transients in bulk and fully-depleted soi technologies. *IEEE Trans. Nucl. Sci.*, 56(6) :3115–3121, 2009.

[23] D Truyen, J Boch, B Sagnes, J-R Vaille, N Renaud, E Leduc, M Briet, C Heng, S Mouton, and F Saigne. Temperature effect on heavy-ion-induced single-event transient propagation in cmos bulk 0.18< formula formulatype=. *IEEE Trans. Nucl. Sci.*, 55(4) :2001–2006, 2008.

[24] Pascale M Gouker, Brian Tyrrell, Matthew Renzi, Chenson Chen, Peter Wyatt, Jonathan R Ahlbin, Stephanie Weeden-Wright, Nick M Atkinson, Nelson J Gaspard, Bharat L Bhuva, et al. Set characterization in logic circuits fabricated in a 3dic technology. *IEEE Trans. Nucl. Sci.*, 58(6) :2555–2562, 2011.

[25] Matthew J Gadlage, Ronald D Schrimpf, Joseph M Benedetto, Paul H Eaton, David G Mavis, Mike Sibley, Keith Avery, and Thomas L Turflinger. Single event transient pulse widths in digital microcircuits. *IEEE Trans. Nucl. Sci.*, 51(6) :3285–3290, 2004.

[26] Matthew J Gadlage, Paul H Eaton, Joseph M Benedetto, and Thomas L Turflinger. Comparison of heavy ion and proton induced combinatorial and sequential logic error rates in a deep submicron process. *IEEE Trans. Nucl. Sci.*, 52(6) :2120–2124, 2005.

[27] J Benedetto, P Eaton, K Avery, D Mavis, M Gadlage, T Turflinger, Paul E Dodd, and G Vizkelethyd. Heavy ion-induced digital single-event transients in deep submicron processes. *IEEE Trans. Nucl. Sci.*, 51(6) :3480–3485, 2004.

[28] JM Benedetto, PH Eaton, DG Mavis, M Gadlage, and T Turflinger. Variation of digital set pulse widths and the implications for single event hardening of advanced cmos processes. *IEEE Trans. Nucl. Sci.*, 52(6) :2114–2119, 2005.

[29] JM Benedetto, PH Eaton, DG Mavis, M Gadlage, and T Turflinger. Digital single event transient trends with technology node scaling. *IEEE Trans. Nucl. Sci.*, 53(6) :3462–3465, 2006.

[30] Jan M Rabaey, Anantha P Chandrakasan, and Borivoje Nikolic. *Digital integrated circuits*, volume 2. Prentice hall Englewood Cliffs, 2002.

[31] Neil HE Weste and Kamran Eshraghian. Principles of cmos vlsi design : a systems perspective. *NASA STI/Recon Technical Report A*, 85 :47028, 1985.

[32] RH Krambeck, Charles M Lee, and H-FS Law. High-speed compact circuits with cmos. *Solid-State Circuits, IEEE Journal of*, 17(3) :614–619, 1982.

[33] Nelson F Goncalves and H De Man. Nora : A racefree dynamic cmos technique for pipelined logic structures. *Solid-State Circuits, IEEE Journal of*, 18(3) :261–266, 1983.

[34] Yuan Ji-Ren, Ingemar Karlsson, and Christer Svensson. A true single-phase-clock dynamic cmos circuit technique. *Solid-State Circuits, IEEE Journal of*, 22(5) :899–901, 1987.

[35] Jiren Yuan and Christer Svensson. High-speed cmos circuit technique. *Solid-State Circuits, IEEE Journal of*, 24(1) :62–70, 1989.

[36] M Jung, J Fuhrmann, A Ferizi, G Fischer, R Weigel, and T Ussmueller. A 10 ghz low-power multi-modulus frequency divider using extended true single-phase clock (e-

tspc) logic. In *Microwave Integrated Circuits Conference (EuMIC), 2012 7th European*, pages 508–511. IEEE, 2012.

[37] Huimin Liu, Xiaoxing Zhang, Yujie Dai, and Yingjie Lv. Low power cmos high speed dual-modulus 15/16 prescaler for wireless communications. In *Communications and Mobile Computing (CMC), 2011 Third International Conference on*, pages 397–400. IEEE, 2011.

[38] S Pellerano, S Levantino, C Samori, and AL Lacaita. A 13.5-mw 5-ghz frequency synthesizer with dynamic-logic frequency divider. *Solid-State Circuits, IEEE Journal of*, 39(2) :378–383, 2004.

[39] Xiao Peng Yu, Manh Anh Do, Wei Meng Lim, Kiat Seng Yeo, and Jian-Guo Ma. Design and optimization of the extended true single-phase clock-based prescaler. *Microwave Theory and Techniques, IEEE Transactions on*, 54(11) :3828–3835, 2006.

[40] Philip C Murley and GR Srinivasan. Soft-error monte carlo modeling program, semm. *IBM Journal of Research and Development*, 40(1) :109–118, 1996.

[41] Ming Zhang and Naresh R Shanbhag. Soft-error-rate-analysis (sera) methodology. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 25(10) :2140–2155, 2006.

[42] MP Baze, WG Bartholet, TA Dao, and S Buchner. An seu analysis approach for error propagation in digital vlsi cmos asics. *Nuclear Science, IEEE Transactions on*, 42(CONF-950716–), 1995.

[43] J. Arlat, M. Aguera, L. Amat, Y. Crouzet, J. Fabre, J. Laprie, and E. Martins. Fault injection for dependability validation : A methodology and some applications. *IEEE Trans. Software Eng.*, 16(2) :166–182, 1990.

[44] D. Holcomb, W. Li, and S. Seshia. Design as you see fit : System-level soft error analysis of sequential circuits. In *DATE*, pages 785–790, 2009.

[45] Hungse Cha, Elizabeth M Rudnick, Janak H Patel, Ravishankar K Iyer, and Gwan S Choi. A gate-level simulation environment for alpha-particle-induced transient faults. *Computers, IEEE Transactions on*, 45(11) :1248–1256, 1996.

[46] Yuvraj Singh Dhillon, Abdulkadir Utku Diril, and Abhijit Chatterjee. Soft-error tolerance analysis and optimization of nanometer circuits. In *Design, Automation, and Test in Europe*, pages 389–400. Springer, 2008.

[47] Chong Zhao, Xiaoliang Bai, and Sujit Dey. A scalable soft spot analysis methodology for compound noise effects in nano-meter circuits. In *Proceedings of the 41st annual Design Automation Conference*, pages 894–899. ACM, 2004.

[48] B. Zhang, W. Wang, and M. Orshansky. Faser : Fast analysis of soft error susceptibility for cell-based designs. In *Intl. Symp. on Quality Electronic Design (ISQED)*, pages 755–760, 2006.

[49] N. Miskov-Zivanov and D. Marculescu. Mars-c : modeling and reduction of soft errors in combinational circuits. In *DAC*, pages 767–772, 2006.

[50] Feng Wang and Yuan Xie. Soft error rate analysis for combinational logic using an accurate electrical masking model. *Dependable and Secure Computing, IEEE Transactions on*, 8(1) :137–146, 2011.

[51] Y. Kuo et al. Accurate statistical soft error rate (sser) analysis using a quasi-monte carlo framework with quality cell models. In *ISQED*, pages 831–838, 2010.

[52] R Rajaraman et al. Seat-la : a soft error analysis tool for combinational logic. In *19th International Conference on VLSI Design*, pages 4–8. IEEE, 2006.

[53] Y. Savaria. *The Design of Digital Machine Tolerant to Soft Errors*. PhD thesis, Department of ECE, McGill University, 1985.

[54] L. Massengill and P. Tuinenga. Single-event transient pulse propagation in digital cmos. *IEEE Trans. Nucl. Sci.*, 55(6) :2861–2871, 2008.

[55] L. Sterpone, N. Battezzati, F. Kastensmidt, and R. Chipana. An analytical model of the propagation induced pulse broadening (pipb) effects on single event transient in flash-based fpgas. *IEEE Trans. Nucl. Sci.*, 58(5) :2333–2340, 2011.

[56] J. Kumar. *Statistical guarantees of performance for RTL designs*. PhD thesis, University of Illinois at Urbana-Champaign, 2012.

[57] Francisco Corella, Zijian Zhou, Xiaoyu Song, Michel Langevin, and Eduard Cerny. Multiway decision graphs for automated hardware verification. *Formal Methods in System Design*, 10(1) :7–46, 1997.

[58] L. De Moura and N. Bjørner. Z3 : An efficient smt solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer, 2008.

[59] B. Dutertre. Yices 2.2. In *Computer Aided Verification*, pages 737–744. Springer, 2014.

[60] R. Bruttomesso et al. The mathsat 4 smt solver. In *Computer Aided Verification*, pages 299–303. Springer, 2008.

[61] C. Barrett et al. Cvc4. In *Computer aided verification*, pages 171–177. Springer, 2011.

[62] Christel Baier, Joost-Pieter Katoen, et al. *Principles of model checking*. MIT press Cambridge, 2008.

[63] R Velazco et al. Heavy ion test results for the 68020 microprocessor and the 68882 coprocessor. *Nuclear Science, IEEE Transactions on*, 39 :436–440, 1992.

[64] F Bezerra et al. Seu and latch-up results on transputers. *IEEE Transactions on Nuclear Science*, 43(CONF-9509107–), 1996.

[65] YF Li, M Li, JY Zhao, RD Schrimpf, DM Fleetwood, B Zhang, JQ Wang, DL Wang, and Y Wang. Characterizing, modeling, and simulating soft error susceptibility in cell-based designs in highly scaled technologies. In *Radiation and Its Effects on Components and Systems (RADECS), 2011 12th European Conference on*, pages 353–358. IEEE, 2011.

[66] Zhao Yuanfu, Yue Suge, Zhao Xinyuan, Lu Shijin, Bian Qiang, Wang Liang, and Sun Yongshu. Single event soft error in advanced integrated circuit. *Journal of Semiconductors*, 36(11) :111001, 2015.

[67] Matthew J Gadlage, Ronald D Schrimpf, Balaji Narasimham, Bharat L Bhuva, Paul H Eaton, and Joseph M Benedetto. Effect of voltage fluctuations on the single event transient response of deep submicron digital circuits. *Nuclear Science, IEEE Transactions on*, 54(6) :2495–2499, 2007.

[68] MC Casey, OA Amusan, SA Nation, TD Loveless, A Balasubramanian, BL Bhuva, RA Reed, D McMorrow, RA Weller, ML Alles, et al. Single-event effects on combinational logic circuits operating at ultra-low power. *Nuclear Science, IEEE Transactions on*, 55(6) :3342–3346, 2008.

[69] Natasa Miskov-Zivanov and Diana Marculescu. Circuit reliability analysis using symbolic techniques. *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, 25(12) :2638–2649, 2006.

[70] Bin Zhang and Michael Orshansky. Symbolic simulation of the propagation and filtering of transient faulty pulses. In *Workshop on system effects of logic efforts*, 2005.

[71] N. Miskov-Z. and D. Marculescu. Mars-s : modeling and reduction of soft errors in sequential circuits. In *Intl. Symp. on Quality Electronic Design (ISQED)*, pages 893–898, 2007.

[72] L. Chen, M. Ebrahimi, and M. Tahoori. Quantitative analysis of soft error propagation at rtl. In *MEDIAN*, 2013.

[73] Enrico Costenaro, A Evans, D Alexandrescu, Liang Chen, Mehdi Tahoori, Michael Nicolaidis, et al. Towards a hierarchical and scalable approach for modeling the effects of sets. In *Proc. of IEEE Workshop on Silicon Errors in Logic-System Effects (SELSE)*, 2013.

[74] J. Baraza, J. Gracia, S. Blanc, D. Gil, and P. Gil. Enhancement of fault injection techniques based on the modification of VHDL code. *IEEE Trans. on Very Large Scale Integration (TVLSI)*, 16(6) :693–706, 2008.

[75] X. Li, S. Adve, P. Bose, and J. Rivers. Softarch : an architecture-level tool for modeling and analyzing soft errors. In *IEEE International Conference on Dependable Systems and Networks (DSN).*, pages 496–505, 2005.

[76] L. Chen, M. Ebrahimi, and M. Tahoori. Formal quantification of the register vulnerabilities to soft error in RTL control paths. *Journal of Electronic Testing*, 31(2) :193–206, 2015.

[77] H. Ochi et al. Breadth-first manipulation of very large binary-decision diagrams. In *ICCAD*, pages 48–55, 1993.

[78] Subhashini Balakrishnan and Sofiene Tahar. A hierarchical approach to the formal verification of embedded systems using mdgs.

[79] S. Tahar, X. Song, E. Cerny, Z. Zhou, M. Langevin, and O. Ait-Mohamed. Modeling and formal verification of the fairisle atm switch fabric using mdgs. *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, 18(7) :956–972, 1999.

[80] Md Hasan Zobair. *Modeling and formal verification of a telecom system block using MDGs.* PhD thesis, Concordia University, 2001.

[81] Y. Xu, X. Song, E. Cerny, and O. Ait Mohamed. Model checking for a first-order temporal logic using multiway decision graphs (mdgs). *The Computer Journal*, 47(1) :71–84, 2004.

[82] M. Kwiatkowska, G. Norman, and D. Parker. Prism 4.0 : Verification of probabilistic real-time systems. In *Computer aided verification*, pages 585–591. Springer, 2011.

[83] Marta Kwiatkowska, Gethin Norman, and David Parker. Probabilistic symbolic model checking with prism : A hybrid approach. *International Journal on Software Tools for Technology Transfer*, 6(2) :128–142, 2004.

[84] C. Barrett, R. Sebastiani, S. Seshia, and C. Tinelli. Satisfiability modulo theories. *Handbook of satisfiability*, 185 :825–885, 2009.

[85] Synopsys HSPICE. Inc. *Version E-2010.12*, Dec. 2010.

[86] A. Daga et al. Automated timing model generation. In *Proceedings of Design Automation Conference (DAC)*, pages 146–151, 2002.

[87] Y Savaria, J Hayes, N Rumin, and V Agarwal. A theory for the design of soft-error-tolerant vlsi circuits. *Selected Areas in Communications, IEEE Journal on*, 4(1) :15–23, 1986.

[88] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3) :382–401, 1982.

[89] K. Driscoll, B. Hall, H. Sivencrona, and P. Zumsteg. Byzantine fault tolerance, from theory to reality. In *Computer Safety, Reliability, and Security*, pages 235–248. Springer, 2003.

[90] Pierre-Louis Aublin, Sonia Ben Mokhtar, and Vivien Quéma. Rbft : Redundant byzantine fault tolerance. In *In Proceedings of the 33rd International Conference on Distributed Computing Systems*, pages 297–306. IEEE, 2013.

[91] Gilson I Wirth, Michele G Vieira, Egas H Neto, and Fernanda Lima Kastensmidt. Modeling the sensitivity of cmos circuits to radiation induced single event transients. *Microelectronics reliability*, 48(1) :29–36, 2008.

[92] J. Palau, M. Calvet, P.E. Dodd, F. Sexton, and Roche Ph. Contribution of device simulation to ser understanding. In *In Proceedings of the International Reliability Symposium*, pages 71–75. IEEE, 2003.

[93] J-M Palau, G Hubert, K Coulie, B Sagnes, M-C Calvet, and S Fourtine. Device simulation study of the seu sensitivity of srams to internal ion tracks generated by nuclear reactions. *IEEE Trans. Nucl. Sci.*, 48(2) :225–231, 2001.

[94] K Castellani-Coulié, J-M Palau, G Hubert, M-C Calvet, PE Dodd, and F Sexton. Various seu conditions in sram studied by 3-d device simulation. *IEEE Trans. Nucl. Sci.*, 48(6) :1931–1936, 2001.

[95] Ginette Monté, Bernard Antaki, Serge Patenaude, Yvon Savaria, Claude Thibeault, and Pieter Trouborst. Tools for the characterization of bipolar cml testability. In *Proceedings of the 19th IEEE VLSI Test Symposium*, page 388. IEEE Computer Society, 2001.

[96] G. Bany Hamad et al. New insights into the single event transient propagation through static and tspc logic. *IEEE Transactions on Nuclear Science*, PP(99) :1–1, 2014.

[97] G. Bany Hamad, S. R. Hasan, O. Ait Mohamed, and Y. Savaria. Abstracting single event transient propagation characteristics to support gate level modeling. In *ISCAS*, 2014, (to be published).

[98] K. Basu et al. Observability-aware directed test generation for soft errors and crosstalk faults. In *VLSID*, pages 291–296, 2013.

[99] C. Loiacono et al. Fast cone-of-influence computation and estimation in problems with multiple properties. In *DATE*, pages 803–806, 2013.

[100] J. Han et al. Reliability evaluation of logic circuits using probabilistic gate models. *Microelectronics Reliability*, 51(2) :468–476, 2011.

[101] J. Stine et al. Freepdk : An open-source variation-aware design kit. In *IEEE ICMS*, pages 173–174, 2007.

[102] D. Limbrick et al. Reliability-aware synthesis of combinational logic with minimal performance penalty. *IEEE Transactions on Nuclear Science*, 60(4) :2776–2781, 2013.

[103] Marco Bozzano, Alessandro Cimatti, and Cristian Mattarei. Efficient analysis of reliability architectures via predicate abstraction. *Hardware and Software : Verification and Testing*, pages 279–294, 2013.

[104] K. Brace, R. Rudell, and R. Bryant. Efficient implementation of a bdd package. In *Proceedings of the 27th ACM/IEEE design automation conference*, pages 40–45. ACM, 1991.

[105] H. Liu, M. Cotter, S. Datta, and V. Narayanan. Soft-error performance evaluation on emerging low power devices. *IEEE Transactions on Device and Materials Reliability*, 14(2) :732–741, 2014.

[106] G. Bany Hamad et al. Segp-finder : Tool for identification of soft error glitch-propagating paths at gate level. In *IEEE ICECS*, pages 358–361, 2011.

[107] V. Forejt, M. Kwiatkowska, G. Norman, and D. Parker. Automated verification techniques for probabilistic systems. In *Formal Methods for Eternal Networked Software Systems*, pages 53–113. Springer, 2011.

[108] Shahrzad Mirkhani, Subhasish Mitra, Chen-Yong Cher, and Jacob Abraham. Efficient soft error vulnerability estimation of complex designs. In *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, pages 103–108. EDA Consortium, 2015.

[109] G. Bany Hamad et al. Investigating the impact of input patterns, propagation paths, and re-convergent paths on the propagation induced pulse broadening. In *Radiation and Its Effects on Components and Systems (RADECS), 14th European Conference on*, pages 358–361. IEEE, 2013.

[110] K. Parker and E. McCluskey. Probabilistic treatment of general combinational networks. *IEEE Trans. on Computers*, 100(6) :668–670, 1975.

[111] Ghaith Bany Hamad, Syed Rafay Hasan, Otmane Ait Mohamed, and Yvon Savaria. Modeling, analyzing, and abstracting single event transient propagation at gate level. In *2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 515–518. IEEE, 2014.

[112] D. Alexandrescu and E. Costenaro. Towards optimized functional evaluation of see-induced failures in complex designs. In *IEEE IOLTS*, pages 182–187, 2012.

[113] Ghaith Bany Hamad, Ghaith kazma, Otmane Ait Mohamed, and Yvon Savaria. Efficient and accurate analysis of single event transients propagation using smt-based techniques. In *International Conference On Computer Aided Design (ICCAD)*, 2016.

[114] Smt-competition. cav. http://smtcomp.sourceforge.net/2015, 2015.

[115] Towards formal abstraction, modeling, and analysis of single event transients at rtl. In *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, pages 2166–2169. IEEE, 2016.

[116] Ghaith Bany Hamad, Otmane Ait Mohamed, and Yvon Savaria. Comprehensive multilevel probabilistic analysis of single event transients propagation induced soft errors. *Submitted to IEEE Transactions on Very Large Scale Integration (TVLSI)*.

[117] Ryo Harada, Yukio Mitsuyama, Masanori Hashimoto, and Takao Onoye. Neutron induced single event multiple transients with voltage scaling and body biasing. In *Reliability Physics Symposium (IRPS), 2011 IEEE International*, pages 3C–4. IEEE, 2011.

[118] Daniele Rossi, Martin Omana, Fabio Toma, and Cecilia Metra. Multiple transient faults in logic : An issue for next generation ics ? In *20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)*, pages 352–360. IEEE, 2005.

[119] Ghaith Bany Hamad, Ghaith kazma, Otmane Ait Mohamed, and Yvon Savaria. Smt-based reliability-aware synthesis for single event transients tolerant combinational circuits. In *IEEE Conference on Radiation Effects on Components and Systems (RADECS)*, 2016.

[120] Ghaith Bany Hamad, Ghaith kazma, Otmane Ait Mohamed, and Yvon Savaria. Single event transients reliability and area-aware synthesis of combinational circuits utilizing satisfiability modulo theories. *Submitted to IEEE Transactions on Nuclear Science*, 2016.