Reliability analysis methods and improvement techniques applicable to digital circuits

Samuel Nascimento Pagliarini

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2013-ENST-0060

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Communications et Electronique »

présentée et soutenue publiquement par

Samuel NASCIMENTO PAGLIARINI

le 15 octobre 2013

Méthodes d'Analyse et Techniques d'Amélioration de Fiabilité pour les Circuits Numériques

Directrice de thèse : **Lirida NAVINER**
Co-encadrement de la thèse : **Jean-François NAVINER**

Jury

M. Jean-Luc LERAY, Directeur de Recherches, CEA	Rapporteur
M. Matteo SONZA REORDA, Professeur, Politecnico di Torino	Rapporteur
M. François MARC, Maître de Conférences, IMS	Examinateur
M. Emmanuel CASSEAU, Professeur, IRISA	Examinateur
Mme. Lirida NAVINER, Professeur, Télécom ParisTech	Directrice de Thèse
M. Jean-François NAVINER, Maître de Conférences HDR, Télécom ParisTech	Directeur de Thèse

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

THÈSE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2013-ENST-0060

EDITE - ED 130

Samuel NASCIMENTO PAGLIARINI

October 15, 2013

Reliability Analysis Methods and Improvement Techniques Applicable to Digital Circuits

THÈSE

Advisor : Lirida NAVINER
Co-advisor : Jean-François NAVINER

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

"In the days of my youth,
I was told what it means to be a man
Now I've reached that age,
I've tried to do all those things the best I can
No matter how I try,
I find my way into the same old jam"

L. Z.

Acknowledgements

This thesis and its results were only obtained thanks to the support of plenty of people from Télécom ParisTech. My deepest gratitude goes out to you all, but especially to my advisors Lirida Naviner and Jean-François Naviner. I would also like to take the time to thank Jean-Luc Leray and Matteo Sonza Reorda, members of the jury, for taking the role of rapporteurs and to the examinateurs François Marc and Emmanuel Casseau. My sincere thankfulness to the whole jury for I know their comments and suggestions will help to improve this thesis.

None of this would be possible without the support of my family. They have helped me to get here in so many ways that this thesis would not be possible without them. Once I tried to explain to my father what circuit placement was. He thought I was talking about soldering work, putting wires together. Multiple times I have tried to explain to my mother what a paper is and how conferences work. Pointless. Nevertheless they support it. Because that is what family should do and that is what they do best. Mom, dad, and little sis, thank you.

I would also like to thank my colleagues and friends Gutemberg, Arwa and Chadi. You are all part of this accomplishment and part of the Paris experience. And so are many others. You know who you are.

Abstract

With the current advances achieved in the manufacturing process of integrated circuits, a series of reliability-threatening mechanisms have emerged or have become more prominent. For instance, physical defects originating from poorly lithographed wires, vias and other low-level devices are commonly seen in nanometric circuits. On the other hand, circuits have also become more sensitive to the strikes of highly energized particles. Both mechanisms, although essentially different, can cause multiple faults that contribute for lower reliabilities in integrated circuits. Multiple faults are more troubling than single faults since these are more severe and also because they can overcome fault tolerance techniques.

Digital circuits are used in most electronic systems nowadays, but there is a specific context in which they are required to be reliable. Such context comprises high-dependability applications, e.g., circuits that are designed targeting medical, aerospace and/ or military use and therefore cannot fail. Although all digital circuits can potentially be affected by faults, the effect of a fault is not as critical in consumer electronic products intended for everyday use. Thus, it is imperative to be able to assess the level of reliability of those dependable circuits and, in case of an unsatisfactory level, to be able to harden those circuits.

This is the scenario in which this thesis is conceived. It's goals are twofold : (a) to propose methods to assess the reliability of digital circuits, and (b) to propose techniques for reliability improvement. Concerning the first goal, several methods have been proposed in the literature and the text shows how these methods present limitations with respect to circuit size (number of gates), circuit type (sequential or combinational) and fault profile (single versus multiple faults). The accuracy obtained when using these methods is also a concern.

This thesis proposes two methods for reliability assessment. The first method is termed SPR+ and its targeted at the analysis of combinational logic only. SPR+ is an analytical approach targeted at estimating the effects of circuit reconvergence. SPR+ improves the average analysis accuracy by taking into account the effect of each fanout reconvergent node to the overall circuit reliability.

Another method, termed SNaP, is also proposed in this thesis. It is a hybrid approach since it is partially based on simulation. SNaP can be used for combinational and sequential logic and can also be emulated in an FPGA device for faster analysis. Both SPR+ and SNaP can cope with multiple faults, a phenomena that is more and more common due to technology scaling.

Another branch of this thesis deals with the improvement of circuit reliability by means of fault tolerance techniques. Such techniques usually have hardening costs that are not negligible. Being so, selective hardening is used instead, and only a few critical parts of the target circuit are hardened. This type of approach allows for a cheaper harde-

ning solution that is able to respect the limitations imposed by tight hardening budgets, either in terms of area, power or timing.

Different approaches for choosing those critical parts have been used and a thoroughly study of the potentials behind selective hardening has also been conducted. Among these approaches, it was studied how selective hardening can be used together with a full circuit-level triplication technique (global TMR) and how the choices of critical parts change in the presence of it. Another approach studied in this thesis is how to limit the effect of multiple faults by using a locality bias. Using benchmark circuits, the savings obtained by applying selective hardening are highlighted in the obtained results.

French Abstract

Introduction

Au cours des dernières années, un développement continu a été observé dans les domaines des systèmes électroniques et des ordinateurs. Ces systèmes sont généralement constitués par un grand nombre de petits systèmes dits circuits intégrés (CIs). La technologie utilisée pour produire ces CIs a changé au cours des dernières décennies dans un processus connu sous le nom de scaling. La figure 1 présente l'évolution de la surface des circuits intégrés sur les 20 dernières années ainsi qu'une projection jusqu'en 2045 (obtenue à partir de International Technology Roadmap for Semiconductors).

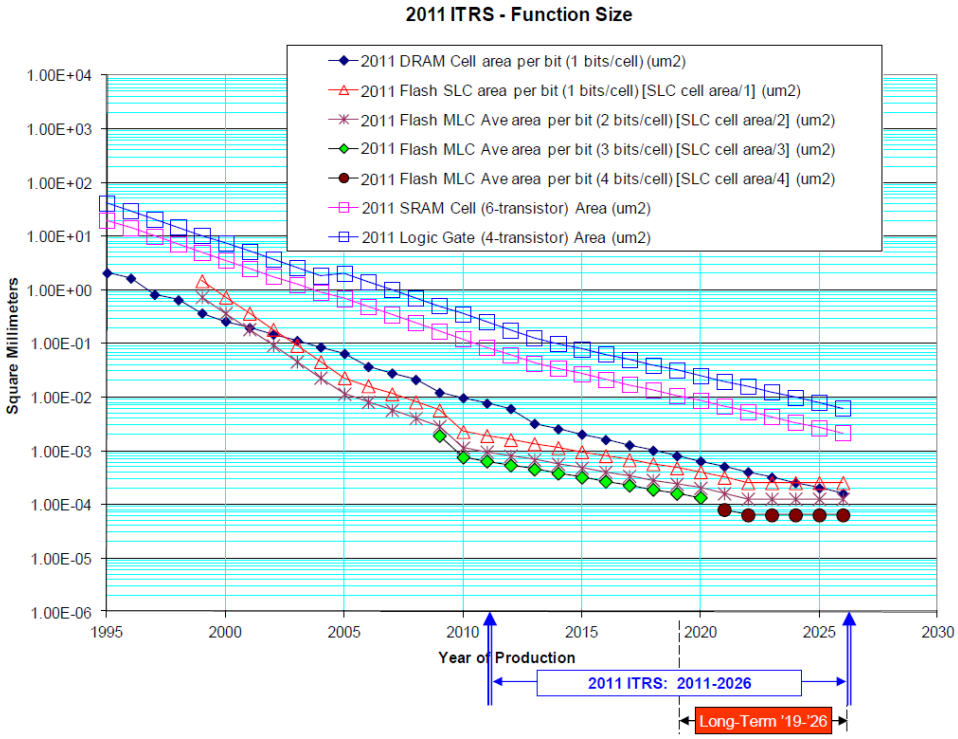


FIG. 1 – Les tendances d'évolution des différents dispositifs.

Une question d'importance vitale est la fiabilité de ces circuits et systèmes, en particulier ceux qui sont utilisés dans des environnements sensibles. Ces environnements sont caractérisés par des exigences strictes d'un attribut donné. Des exemples de cet attribut sont la fiabilité, la disponibilité, la sécurité, survivabilité et la maintenabilité. Ces attributs

sont décrits en détail dans le texte qui suit.

Dans cette thèse, nous nous focaliserons sur les circuits numériques. Ce type de circuit est utilisé dans la plupart des appareils présents dans notre vie quotidienne, comme les téléphones mobiles, ordinateurs, appareils photo, etc. La figure 2 illustre une des approches possibles pour construire le diagramme de blocs d'un circuit numérique :

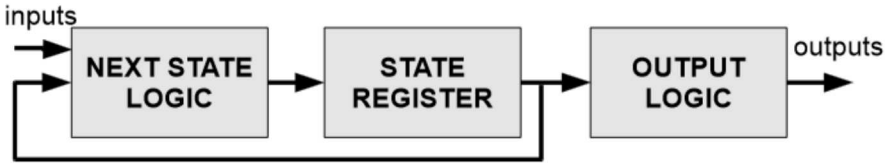


FIG. 2 – Schéma d'un circuit numérique avec ses parties séquentielles et combinatoires.

L'illustration de la figure 2 montre les entrées et sorties d'un circuit, ainsi que la logique interne de l'état actuel et l'état suivant. La logique d'état actuel est stockée dans les éléments de mémoire et est appelée comme logique séquentielle. La logique de l'état suivant ne stocke pas les données, elle calcule les données basées sur les entrées et l'état actuel ; ce type de logique est dit combinatoire. Un système comme celui-ci, en utilisant la logique séquentielle et combinatoire, est répliqué de nombreuses fois pour construire des circuits plus complexes. L'information pertinente ici est que, quels que soient leurs types, les éléments logiques ne sont pas totalement fiables. Ceci sera expliqué en détail plus loin dans ce manuscrit.

Sûreté de fonctionnement

Selon Avizienis, un système électronique peut être caractérisé par quatre propriétés : la fonctionnalité, le performance, le coût et la sûreté de fonctionnement. Les trois premières propriétés sont naturellement liées les unes aux autres, donc un compromis entre ces propriétés est établie. Ce compromis est bien connu parmi les designers. Néanmoins, la fiabilité doit également être considérée dans certains scénarios, ce qui ajoute un élément à une équation qui est déjà assez complexe.

La sûreté de fonctionnement d'un système informatique est sa capacité à offrir un service qui peut être digne de confiance. Une taxonomie complète de la sûreté de fonctionnement et de ses concepts connexes est représentée sur la figure 3. Ces concepts sont divisés en menaces (*threats*), attributs (*attributes*) et moyens (*means*).

Il existe une relation entre les menaces. Cette relation est illustrée sur la figure 4. En termes simples : une faute peut activer une erreur, alors qu'une erreur peut se propager et provoquer une défaillance. Une telle défaillance pourrait alors représenter une faute dans un système plus vaste. Ainsi, le processus d'activation et propagation continue jusqu'à un point où il peut en fait obtenir une visibilité dans l' ensemble du système, ce qui provoque un fonctionnement erroné ou non satisfaisant.

Fiabilité dans les circuits numériques

Les progrès dans l'industrie des semi-conducteurs ont amélioré significativement la performance des circuits numériques. La grande partie de ce gain est attribuable aux petites dimensions et basse tension, qui ont conduit à des architectures complexes avec un grand parallélisme combiné à une haute fréquence.

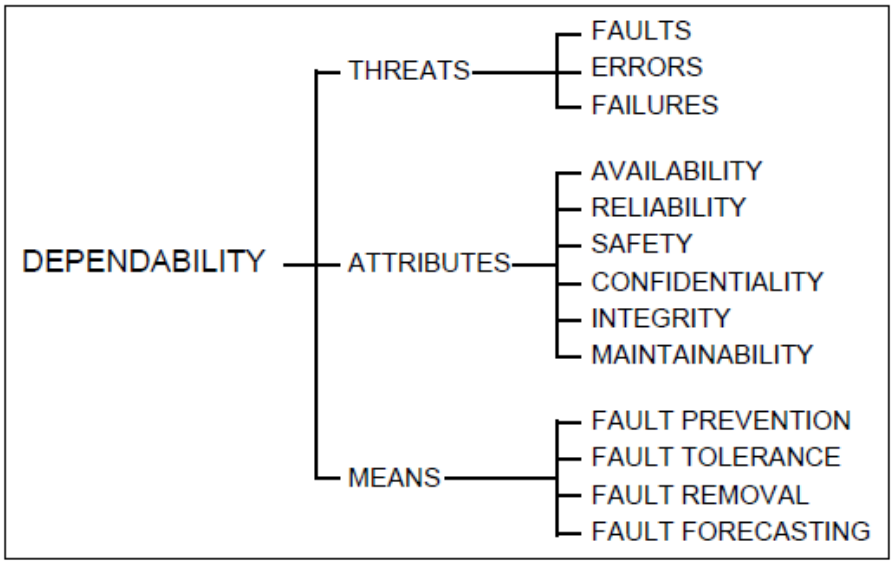


FIG. 3 – Taxonomie de la sûreté de fonctionnement et de ses concepts connexes.

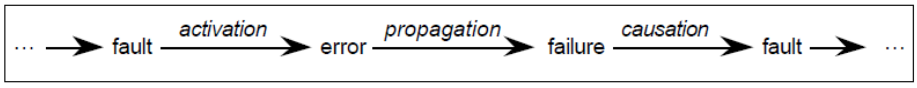


FIG. 4 – Chaîne des menaces et leurs propagation.

Cependant, le même progrès technologique qui a rendu tout cela possible, a également réduit la fiabilité des transistors. En réduisant la tension de seuil et en réduisant la marge de bruit, les transistors sont plus sensibles aux défauts de différentes sources. Tous les éléments typiques d'un circuit numérique sont construits en utilisant des réseaux de transistors. En conséquent, un transistor à faible fiabilité réduit la fiabilité du circuit complet.

Les fautes qui affectent un circuit numérique sont classées en trois catégories : permanente, intermittente ou transitoire. Pour chaque type de faute, différentes stratégies sont appliquées pour détecter et corriger (quand et si possible).

Les défauts de fabrication sont un exemple de fautes permanentes. Dans le processus de fabrication de circuits intégrés, une grande quantité de dispositifs électroniques est produite simultanément dans une série d'étapes très complexes. La probabilité que l'ensemble de ces dispositifs (et aussi de leurs interconnexions) fonctionnera correctement dépend du degré de contrôle exercé dans leur fabrication. La fraction de puces qui, à la fin de la fabrication, peuvent satisfaire un ensemble d'exigences de test est appelée le rendement.

Un exemple d'un défaut de type open est représenté sur la figure 5, qui montre la vue de haut (a) et la section transversale (b) d'un défaut dans la couche M2 (métal 2).

La figure 6 montre l'effet d'une particule ionisante quand elle traverse une jonction de silicium, en créant ainsi une faute transitoire. Il est montré comment la charge générée dans le substrat de silicium est collectée. Plusieurs mécanismes de transport de charge peuvent être impliqués en fonction de la technologie utilisée et de la conception du circuit. L'image montre deux mécanismes différents, qui sont appelés drift et diffusion. Le

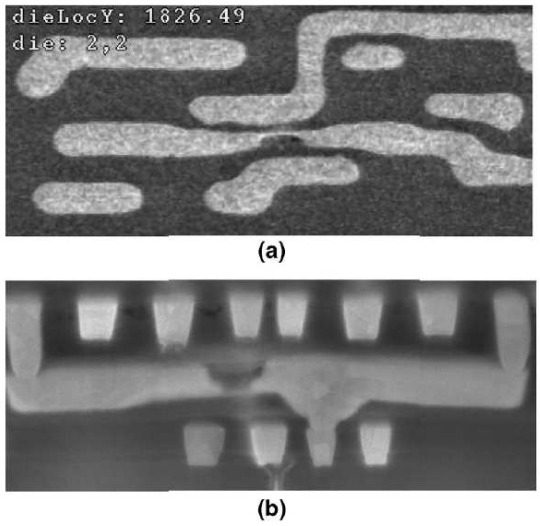


FIG. 5 – Vue de haut (a) et section transversale (b) d'un défaut.

premier mécanisme est entraîné par un champ électrique et se produit très rapidement, tandis que le deuxième n'est pas aussi rapide.

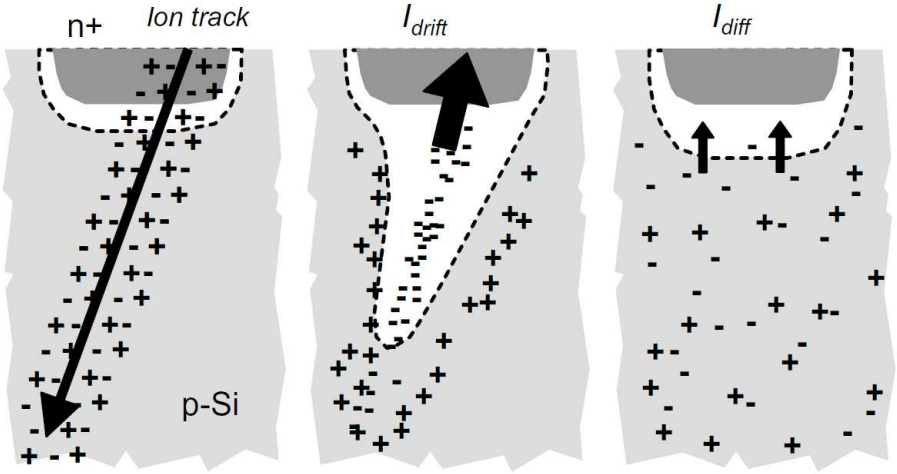


FIG. 6 – Effet d'une particule ionisante dans le silicium.

Si la charge collectée est plus grande que la charge critique (montant minimal de charge qui doit être déposé par une particule afin de produire une transition capable de changer une valeur logique), elle est alors perçue par le circuit comme valide. Cela est représenté sur la figure 7.

La figure 7 représente l'instant de collision de la particule ainsi que les deux mécanismes de transport de charge. Puisque le mécanisme de drift est relativement rapide (de l'ordre de la picoseconde), une impulsion de courant rapide est générée. Lorsque le mécanisme de diffusion commence, il n'est pas aussi rapide (ordre de la nanoseconde). Ainsi, l'impulsion de courant générée change en une forme de queue.

Le Soft Error Rate est la probabilité qu'un un dispositif (ou système) subisse des soft er-

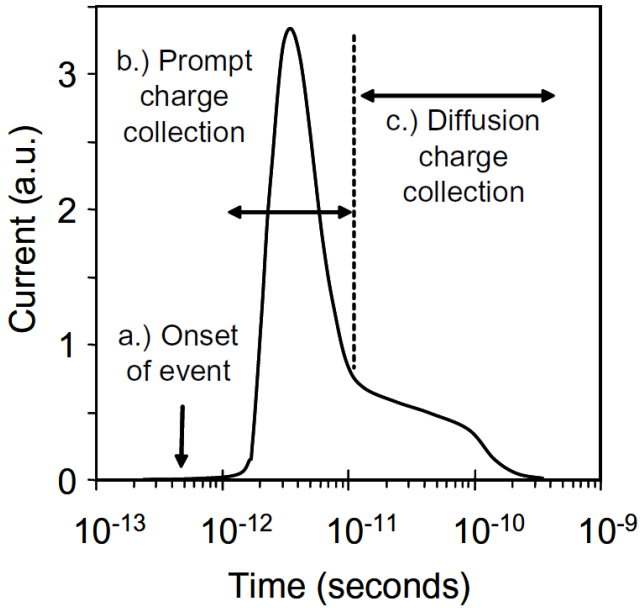


FIG. 7 – Courant à la jonction et les mécanismes de collecte concernés.

rors comme celles décrites ci-dessus. Des recherches antérieures montrent que le SER par puce est en augmentation, principalement en raison de la croissance rapide du nombre de transistors sur une seule puce. Lorsqu'on examine des technologies de pointe, une seule particule peut affecter plusieurs nœuds dans un circuit. Ce processus est connu sous le nom de charge sharing. À cause de cela, des modèles à plusieurs fautes doivent être pris en compte afin d'analyser le taux d'erreurs dans les circuits intégrés. Les calculs des taux d'erreur peuvent être significativement plus petits que ceux observés dans le circuit réel si les modèles multi-fautes transitoires ne sont pas utilisés.

Etat de l'Art

L'analyse de fiabilité des composants électroniques généralement aborde deux aspects très différents : la prédiction de fiabilité et évaluation de la fiabilité. L'objectif du travail présenté dans cette thèse est surtout la prédiction de la fiabilité, c'est à dire, on suppose qu'il existe un autre processus utilisé pour caractériser la fiabilité des éléments individuels d'un circuit.

Certaines techniques utilisées pour estimer la fiabilité d'un circuit numérique s'appliquent uniquement à la logique combinatoire alors que d'autres sont plus générales. En ce qui concerne la logique combinatoire, cette étude se concentre sur les propriétés de masquage logique. Il est bien connu que l'estimation du masquage logique est beaucoup plus complexe que l'estimation du masquage électrique ou masquage temporel.

Injection de fautes par simulation

L'injection de fautes est une approche très simpliste et intuitive pour estimer la fiabilité d'un circuit. En raison de sa simplicité, elle a reçu une grande attention de la part des chercheurs. Le processus commence par le choix d'un noeud (un bloc, une cellule ou un

transistor, en fonction de la granularité de l'analyse) et on procède ensuite à décaler la valeur de sortie pour un temps donné. Habituellement deux versions d'un même circuit sont simulées au même temps : une version sans erreur (golden version) et une version sujette à des fautes. La simulation vérifie ensuite si les sorties des deux circuits sont égales.

Si possible, le processus décrit ci-dessus est répété pour tous les noeuds. Une métrique est ensuite appliquée pour mesurer la fiabilité du circuit. Par exemple, on peut prendre le rapport entre les erreurs détectées et non détectées. Ce rapport est une mesure de la capacité de masquage du circuit, par conséquent, reflète la fiabilité du circuit.

Il faut préciser que l'injection de fautes basée sur la simulation est coûteuse en terme de temps de calcul. Le problème est que pour une analyse complète, il est nécessaire de simuler tous les scénarios possibles, y compris tous les sites de défaut et tous les vecteurs d'entrée possibles. Il est clair que cette combinaison peut conduire à un nombre de scénarios intraitable. Le nombre de scénarios augmente encore plus si plusieurs fautes doivent être considérées. Ainsi, des évaluations partielles sont habituellement effectuées. Sélectionner les parties du circuit qui devraient être évaluées et celles qui peuvent être ignorées est également un problème. Pour faire face aux contraintes de temps de la simulation, des techniques d'émulation ont été créées. Ces techniques sont explorées dans la section suivante.

Injection de fautes par emulation

L'idée de base de l'injection de fautes, soit par émulation ou par simulation, est exactement la même. Néanmoins, les solutions d'émulation utilisent une plateforme de support telle qu'un FPGA. La plupart des solutions font usage d'un "off-the-shelf commercial" FPGA. Ces plateformes offrent beaucoup de ressources et ont été utilisées avec succès pour obtenir des résultats plus rapides (par rapport à des approches fondées sur la simulation). Malheureusement, l'utilisation de ces plateformes apporte aussi un inconvénient considérable : l'observabilité est généralement faible, c'est à dire, l'utilisateur n'a pas d'accès direct à tous les signaux du circuit en cours d'analyse.

Dans les dernières années, les FPGA ont évolué de telle sorte que la reconfiguration partielle est possible. Certaines cartes permettent la reconfiguration dynamique, c'est à dire, tandis que les pièces du circuit fonctionnent d'autres parties peuvent être reconfigurées. Ceci permet d'effectuer une analyse légèrement différente : d'abord, une configuration du circuit sans défaut est faite. Cette première exécution est analysée et l'état de chaque bascule du circuit est connu à tous les cycles. Ensuite, un second passage du circuit suit, dans lequel la reconfiguration est appliquée pour modifier l'état d'une bascule à la fois, recréant ainsi l'effet d'un défaut.

Injection de fautes par des moyens physiques

Plusieurs techniques entrent dans la catégorie de l'injection physique. En général, ces techniques utilisent une certaine forme de source de faute accélérée. L'accès à ces sources peut être coûteux et compliqué dans certains cas. Un échantillon du circuit souhaité est nécessaire pour ce type de technique, qui a également un coût associé. Ainsi, ce type de technique est utile pour la caractérisation après fabrication, un processus utilisé pour confirmer qu'un circuit intégré est conforme à une certaine norme comme un taux de défaillance maximal.

Plusieurs auteurs ont utilisé la haute altitude pour des expériences en temps réel.

L'idée est que le flux de particules est plus dense à haute altitude, ce qui peut rendre la caractérisation d'un dispositif plus rapide et tout aussi fiable. Il y a plusieurs détails relatifs à toutes ces techniques d'injection physiques qui comportent la physique de bas niveau et ne sont pas l'objet de ce travail. Une expérience avec des neutrons pour injecter des fautes dans un FPGA est raportée dans l'annexe B.

Approches analytiques

Les approches analytiques ont été développées en raison de l'inefficacité ou de l'incapacité des méthodes d'injection de fautes traditionnelles pour manipuler de grands circuits. Ces approches ont leurs propres limites, mais en général elles ne présentent pas les coûts élevés de techniques physiques, ni les longs délais d'exécution de solutions basées sur la simulation. Les méthodes analytiques peuvent estimer la fiabilité de la logique combinatoire seulement.

La première méthode d'analyse qui sera présentée est la Probabilistic Transfer Matrices. Il s'agit d'une approche simple qui modélise, grâce à l'utilisation de matrices, les portes logiques et la topologie d'un circuit. L'idée principale de la méthode est de définir la corrélation entre les motifs de sortie et les motifs d'entrée d'un circuit. Pour ce faire, chaque porte logique est également représentée comme une matrice PTM. Cette représentation est obtenue par l'utilisation de deux éléments auxiliaires : la Ideal Transfer Matrix et le paramètre q, comme illustré sur la figure 8. La matrice ITM représente la table de vérité de la porte logique.

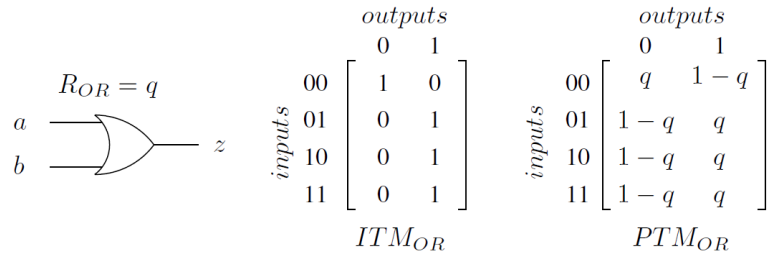


FIG. 8 – La représentation PTM d'une porte logique.

En sachant que chaque porte logique dans un circuit est représentée par une matrice PTM, il est alors nécessaire de calculer la matrice PTM de l'ensemble du circuit en prenant en compte la topologie. Ce calcul est effectué par le leveling du circuit ciblé, comme illustré sur la figure 9.

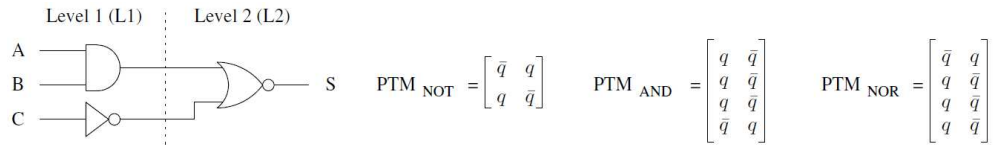


FIG. 9 – Leveling du circuit ciblé.

Bien que la méthode PTM soit capable d'estimer la fiabilité d'un circuit avec précision, elle souffre d'un temps de simulation intraitable, même pour les circuits de taille moyenne. Ceci est dû au fait que la complexité croît d'une manière exponentielle. La PTM est la base de toutes les méthodes de la famille Signal Probability Reliability. Lors de l'application de

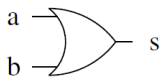
la méthode PTM, la taille des matrices intermédiaires augmente à un rythme rapide. La méthode SPR tente d'éviter ce problème en représentant chaque signal dans le circuit par une matrice 2x2. Une telle matrice est illustrée sur la figure 10.

$$P_{2\times 2}(signal) = \begin{bmatrix} P(signal = correct\ 0) & P(signal = incorrect\ 1) \\ P(signal = incorrect\ 0) & P(signal = correct\ 1) \end{bmatrix}$$

FIG. 10 – Représentation SPR.

Dans la représentation SPR, il est supposé que le signal peut avoir quatre valeurs distinctes. Ces valeurs sont : un '0 correct', un '0 incorrect', un '1 correct' et un '1 incorrect'. La matrice SPR contient la probabilité d'un signal d'être l'une des valeurs mentionnées. Considérons une porte OR, comme représenté sur la figure 11. Supposons aussi que ses entrées sont déjà représentées comme matrices SPR A4 et B4. Pour le calcul des sorties de la matrice SPR, il est nécessaire de prendre en compte les entrées, la fonction logique et également la fiabilité de la porte.

$$A_4 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \qquad B_4 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$



$$q_{OR} = 0.95 \qquad S_4 = \begin{bmatrix} s_0 & s_1 \\ s_2 & s_3 \end{bmatrix}$$

$$\begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.25 \end{bmatrix} \times \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \\ 0.05 & 0.95 \\ 0.05 & 0.95 \end{bmatrix} = \begin{bmatrix} 0.2375 & 0.0125 \\ 0.0125 & 0.2375 \\ 0.0125 & 0.2375 \\ 0.0125 & 0.2375 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.2375 & 0.0125 \\ 0.0375 & 0.7125 \end{bmatrix}$$

$I = A_4 \otimes B_4$ $\quad PTM_{OR}$ $\quad P(S)$ $\quad S_4$

FIG. 11 – Exemple de la propagation SPR dans une porte OR.

Techniques de tolérance aux fautes

Cette section présente quelques techniques utilisées pour augmenter la fiabilité des circuits numériques. Certaines techniques permettent de détecter des erreurs, d'autres de détecter et de corriger tandis qu'un troisième groupe se concentre sur la prévention des erreurs en améliorant la fiabilité du circuit.

La redondance modulaire est une famille de techniques basée sur la redondance spatiale. Proposée par Von Neumann, la Triple Modular Redundancy est la technique la plus célèbre de la famille. Elle consiste à disposer trois copies du même module fonctionnant en parallèle. Le principe du voter TMR par majorité est que si une seule erreur se produit dans l'un des modules, elle sera masquée.

Certaines applications, par contrainte de budget, ne permettent pas le recours à une solution de triplication complète du système.. Ainsi, une certaine forme de durcissement sélectif a lieu lorsque seulement quelques zones du circuit sont durcies tandis que

d'autres sont laissés intactes. Le problème est la détermination de ces zones. Ce problème a été déjà traité par plusieurs auteurs dans la litérature.

Le dimensionnement des portes est une technique particulière qui peut également être considérée comme une forme de durcissement sélectif.

Méthodes d'Analyse de Fiabilité

Cette section couvre deux méthodes qui ont été développées pour l'analyse de la fiabilité d'un circuit. Comme indiqué précédemment, on constate un manque de méthodes capables de faire face aux nombreuses difficultés imposées par l'analyse de la fiabilité. La première méthode proposée ici essaie de aborder la question de la précision en effectuant une analyse de noeuds reconvergents. Cette méthode est appelée SPR+.

La deuxième méthode proposée dans cette thèse est appelée SNaP et elle applique une approche complètement différente pour l'analyse de la fiabilité. Elle s'agit d'une méthode hybride combinant les avantages de la simulation et de solutions analytiques.

SPR+

Le charge sharing a augmenté la quantité de fautes multiples. Ainsi, des algorithmes capables de gérer de multiples fautes sont de grand intérêt. La simulation traditionnelle peut être utilisée pour la modélisation des erreurs multiples, mais elle peut facilement devenir un problème si toutes les combinaisons de sites (de fautes) doivent être prises en compte. Les méthodes analytiques sont essentielles pour ce type d'analyse.

La contribution de la méthode SPR+ est de proposer deux heuristiques simples pour estimer la fiabilité exacte du circuit. Ces heuristiques tiennent compte seulement de la convergence de premier ordre, donc elles peuvent être utilisées pour l'évaluation des grands circuits pour lesquels la simulation et les autres algorithmes analytiques ne parviennent pas à faire une estimation de la fiabilité ou le font avec un temps de simulation très élevé.

Les circuits de référence ISCAS'85 ont été analysés en utilisant la méthode SPR-MP mais limitée par une analyse de 12ème ordre (c'est à dire, les 12 noeuds plus pertinents de chaque circuit ont été pris en compte, tous en même temps). Cette analyse est appelée R_{12th}. Puisque la méthode SPR-MP permet de faire l'évaluation partielle des fanouts, elle était la méthode choisie pour donner une référence de fiabilité à chaque circuit. Les valeurs de fiabilité ainsi que les temps d'exécution sont donnés dans le tableau 1. Les temps d'exécution indiqués comprennent aussi le temps nécessaire pour trouver les 12 fanouts les plus pertinentes.

Le choix d'un 12ème ordre (au lieu d'un ordre inférieur ou supérieur) est motivé par deux raisons : un temps d'exécution acceptable et une précision suffisante. Les temps d'exécution sont donnés dans le tableau 1 avec un maximum de temps de simulation pour le circuit c1355 d'environ une heure.

Pour la précision, il est important de souligner que les fanouts peuvent contribuer de différentes manières. L'illustration dans la figure 12 montre comment la fiabilité du circuit converge vers une valeur par le passage à une analyse d'ordre plus élevée, en ajoutant un fanout à la fois. Ainsi, les premières fanouts analysées sont plus importantes que les autres. Il est également possible de voir comment les temps de calcul augmentent rapidement.

TAB. 1 – Analyse de fiabilité R_{12th}.

Circuit	Gates	Fanouts	Reliability	Execution time (s)
c17	6	3	0.9999437519	0.05
c432	160	89	0.9986423278	486.93
c499	202	59	0.9986269089	30.32
c1355	546	259	0.9977799443	3663.79
c1908	880	385	0.9967790239	130.4
c2670	1269	454	0.9933852285	1142.42
c3540	1669	579	0.9934856289	80.17
c5315	2307	806	0.9910769681	2015.51

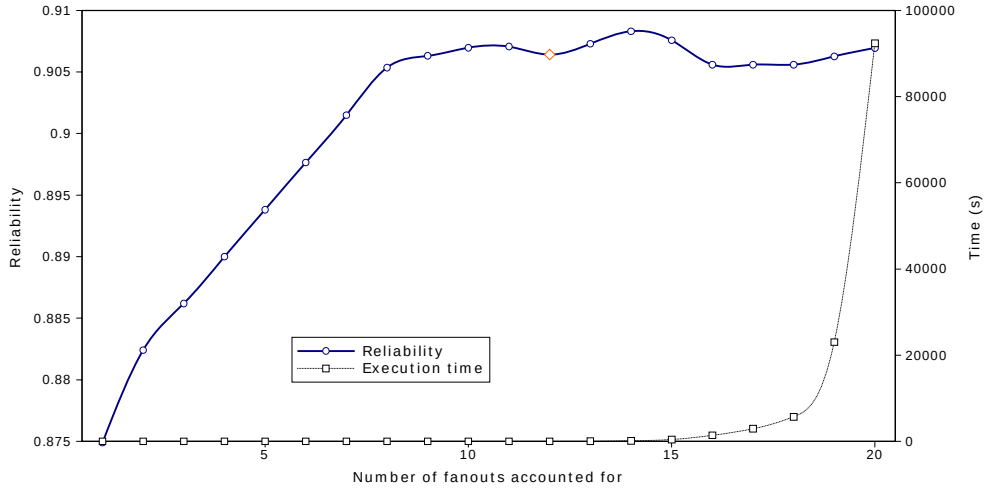


FIG. 12 – Analyse du nombre différent de fanouts, circuit c499.

Une tendance différente est vue dans la figure 13. Cette analyse a été obtenue en utilisant le circuit c3540. Néanmoins, la fiabilité mesurée pour l'ensemble du circuit converge toujours vers une valeur. De toute évidence, comme on le voit dans les deux images, la fiabilité mesurée par l'analyse de 12ème ordre (marquée par une forme de diamant orange) est plus proche de la fiabilité réelle.

$$D(f) = jR_1(f) \square R_0j \quad (1)$$

Il est clair que l'utilisation d'une estimation de 12ème ordre ne conduit pas à une valeur de fiabilité précise. Néanmoins, il est clair que les fanouts n'ont pas le même impact sur la fiabilité globale (c'est-à-dire que toutes les valeurs $D(f)$ ne sont pas forcément égales ni même de même ordre de grandeur). Le résultat montré sur la figure 14 est une tentative de classer chaque fanout par sa valeur $D(f)$. Tout d'abord, la plus grande différence a été identifiée et a été appelée D_{max}. Ensuite, l'impact de chaque fanout est classé comme suit :

– Impact majeur, si $D(f)=D_{max} > 0:8$
– Impact moyen, si $D(f)=D_{max} > 0:2$
– Faible impact, pour tous les autres ($D(f)=D_{max} <= 0:2$)

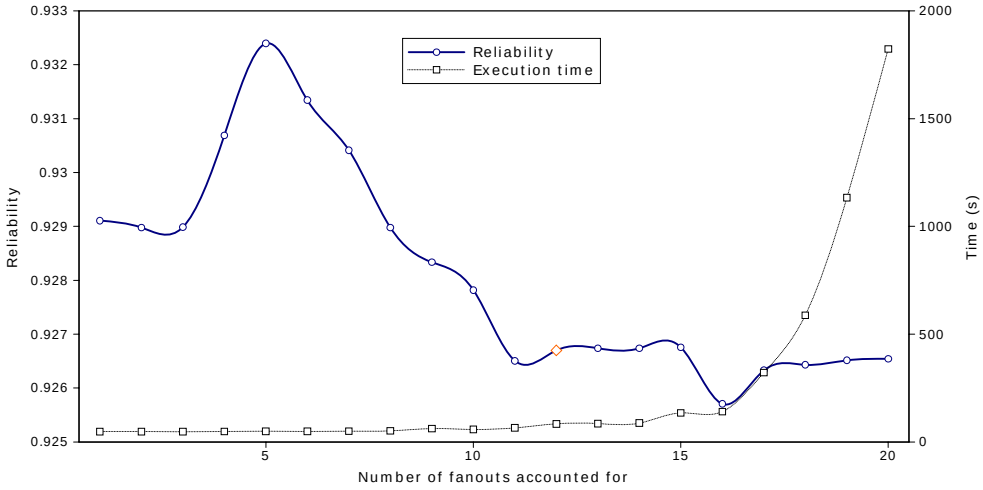


FIG. 13 – Analyse du nombre différent de fanouts, circuit c3540.

Il apparaît clairement sur la figure 14 que le nombre de fanouts d'impact majeur est très faible.Les fanouts ayant un impact majeur sont moins de 3 % du montant total de fanouts. La marge proposée pour les fanouts d'impact moyen est assez large et, même ainsi, ils représentent moins de 10 % du nombre de fanouts. Ainsi, la majorité absolue des fanouts n'est pas si importante lors de l'évaluation de la fiabilité. Ceci peut être exploité en vue d'estimer la fiabilité d'une manière précise en utilisant une courte période de temps.

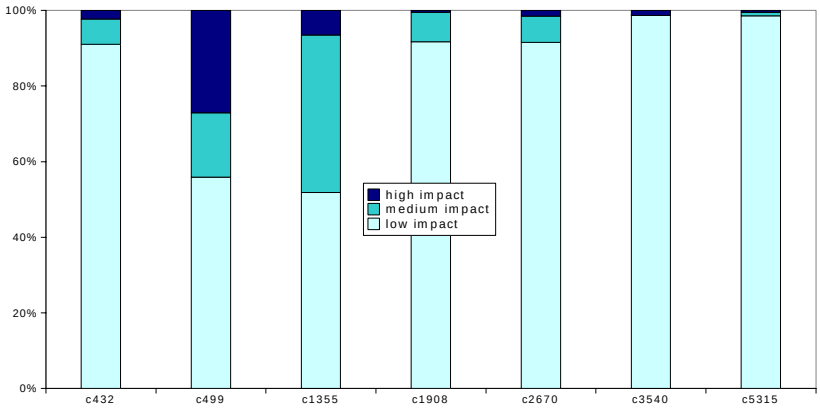


FIG. 14 – Impact des noeuds basées sur les valeurs D (f).

En prenant en compte le profil de l'impact révélé par la figure 14, deux heuristiques différentes ont été proposées. Toutes les deux ont le même objectif : se rapprocher de la valeur réelle de la fiabilité R en ne tenant compte que des estimations de premier ordre. Les résultats sont présentés dans la figure 15.

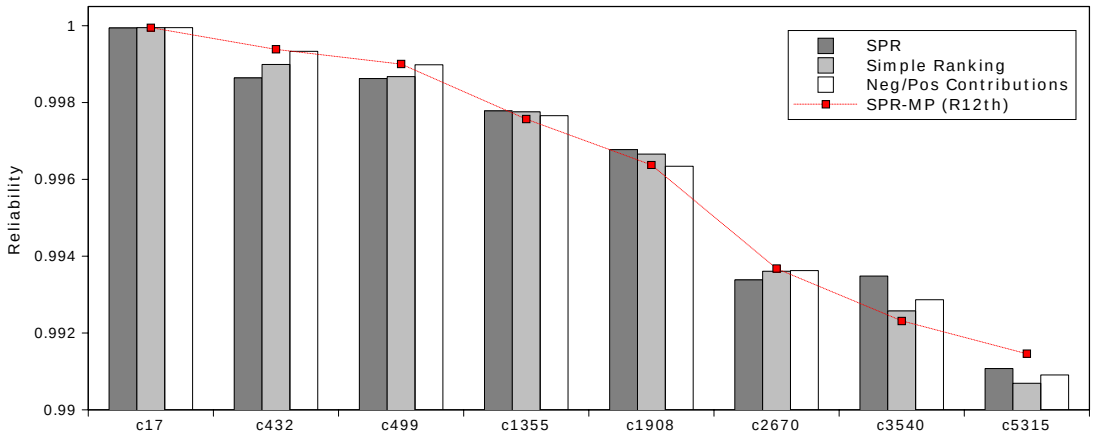


FIG. 15 – Comparaison entre les deux heuristiques SPR+ et SPR.

SNaP

La majorité des méthodes peuvent seulement traiter des circuits combinatoires et de petites tailles. Certaines méthodes sont aussi totalement incapables d'estimer la fiabilité des circuits de tailles moyennes. Au vu de ces limitations, une nouvelle méthode hybride a été développée. Cette méthode est appelée SNaP et elle est considérée comme une solution hybride car certaines parties de la méthode reposent sur de la simulation, tandis que d'autres ne le font pas.

SNaP peut également bénéficier de l'émulation, lorsqu'elle est utilisée comme une plateforme dans un FPGA. L'émulation permet une évaluation rapide de circuits complexes. Ainsi, une mise en oeuvre possible de la méthode sera montrée à l'aide d'une implémentation Verilog pleinement synthétisable.

Les concepts de base derrière la modélisation SNaP sont la création de fautes et la propagation de fautes. SNaP est basé sur ces deux concepts opposés, c'est à dire, les portes sont capables de générer des fautes et sont également capables de supprimer des fautes. C'est l'interaction qui détermine la fiabilité de l'ensemble du circuit. Le masquage logique est également considéré au cours de cette évaluation.

Modélisation de la logique combinatoire

Initialement, nous considérons un petit circuit qui contient uniquement un simple inverseur. La figure 16 (a) contient une représentation de son comportement fonctionnel. Le circuit transformé est donné dans la figure 16 (b). Il contient des signaux d'E/ S supplémentaires et un bloc d'analyse supplémentaire. Les inverseurs ne masquent pas des fautes, à savoir, une entrée défectueuse ne sera jamais filtrée par la porte. Bien qu'aucun masquage a lieu, l'inverseur est toujours une source d'erreur possible et cette 'aptitude' doit être prise en compte. Ainsi, dans SNaP, chaque porte transformée stocke une valeur gf qui exprime le taux auquel les fautes sont générées à cette porte particulière.

ifs est un paramètre qui indique le nombre de fautes pouvant atteindre ce nœud d'entrée. De même, ofs est un paramètre de noeuds de sortie et peut être défini comme suit :

$$ofs = ifs + gf_{inv} \quad (2)$$

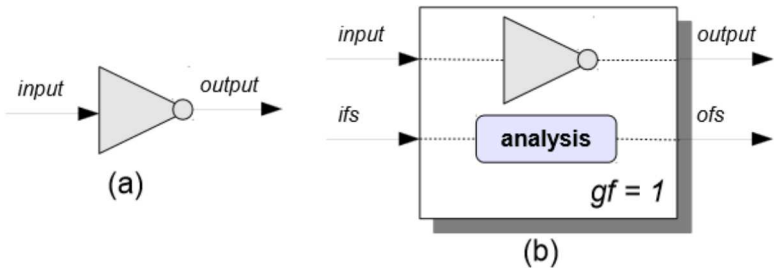


FIG. 16 – (a) comportement fonctionnel d'un inverseur ; (b) modélisation SNaP.

Il est obligatoire de prendre le masquage logique en compte dans l'évaluation de la fiabilité. Pour cette raison, nous considérons un autre circuit, illustré dans la figure 17 (a) et sa version modifiée dans la figure 17 (b).

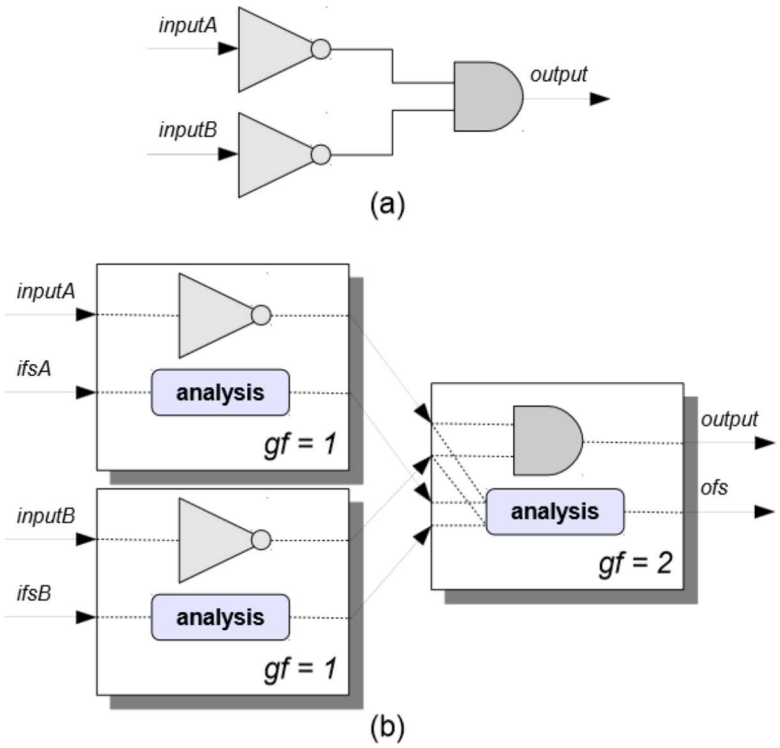


FIG. 17 – Un circuit simple et ses représentations (a) fonctionnelles et (b) modifiés par SNaP.

Le bloc d'analyse de la porte AND est mis en ouvre comme une machine d'états finis (FSM) avec 5 états : waiting, errorOnInputA, errorOnInputB, errorOnBoth et finished. Cette FSM est entièrement synthétisable et peut être généralisée pour les portes avec un nombre plus élevé d'entrées. Les états waiting et finished restent exactement les mêmes, peu importe le nombre d'entrées, tandis que les autres augmenteront. Il y aura un état pour chaque combinaison possible de fautes simples et multiples.

Pour comprendre comment la méthode fonctionne, nous procédons à une description de chaque état de la FSM. Chaque état peut être modélisé par une équation comme suit :

$$ofs = gf_{and} \tag{3}$$

$$ofs = ofs + (ifsAndA >> 1) \tag{4}$$

$$ofs = ofs + (ifsAndB >> 1) \tag{5}$$

$$ofs = ofs + ((ifsAndA + ifsAndB) >> derFactor) \tag{6}$$

Le choix du facteur de réduction est déterminé de manière empirique. Il est égal au nombre d'entrées de la porte plus un. Ainsi, pour le cas particulier de la porte AND à 2 entrées, le facteur de réduction est derFactor = 3.

Modélisation de la logique séquentielle

La modélisation de la logique séquentielle est beaucoup plus simple que celle utilisée pour la logique combinatoire. Similairement aux portes utilisées dans la logique combinatoire, une valeur gf est définie pour chaque-bascule (gf_{ff}). Aucun masquage logique prend place à l'intérieur d'une bascule. Pour calculer la valeur ofs d'une bascule, l'équation suivante est utilisée :

$$ofs = ifs + gf_{ff} \tag{7}$$

Cela étant dit, le problème devient alors de synchroniser tous les éléments du circuit correctement. Puisque toutes les cellules combinatoires de la description originale sont maintenant décrites suivant machines à états finis, elles ne peuvent pas percevoir le même signal d'horloge que les bascules dans la description originale du circuit. Ce problème est résolu avec l'utilisation des signaux de contrôle spéciaux qui créent le même effet d'un réseau d'horloge secondaire.

Résultats

Des entrées aléatoires ont été utilisées pour les circuits combinatoires utilisés dans les expériences de cette section. Néanmoins, un nombre suffisamment élevé d'entrées (plusieurs échantillons) doit être utilisé pour obtenir une fiabilité significative. Cet effort est représenté dans la figure 18 pour le circuit ISCAS'85 c432.

La figure 18 montre comment la fiabilité du circuit tend à une valeur moyenne lorsque plusieurs échantillons sont ajoutés. L'objectif est de déterminer combien d'échantillons sont nécessaires pour que l'évaluation soit une bonne approximation de la valeur réelle. Dans ce cas particulier, pour le circuit c432, il est supposé que 4000 échantillons sont suffisants (comme souligné en rouge dans l'image elle-même). L'augmentation du nombre d'échantillons pourrait être injustifiée puisque leur contribution devient négligeable.

La fiabilité par rapport à différentes entrées peut être obtenue avec la méthode proposée. La figure 19 montre l'analyse du circuit c17 en utilisant des modèles d'entrée aléatoires. L'image montre clairement que certains scénarios d'entrée (sur l'axe x) peuvent conduire à des valeurs plus élevées de ofs que d'autres (et par conséquent, des valeurs de fiabilité inférieures). Les courbes en pointillés dans la figure 19 représentent les sorties

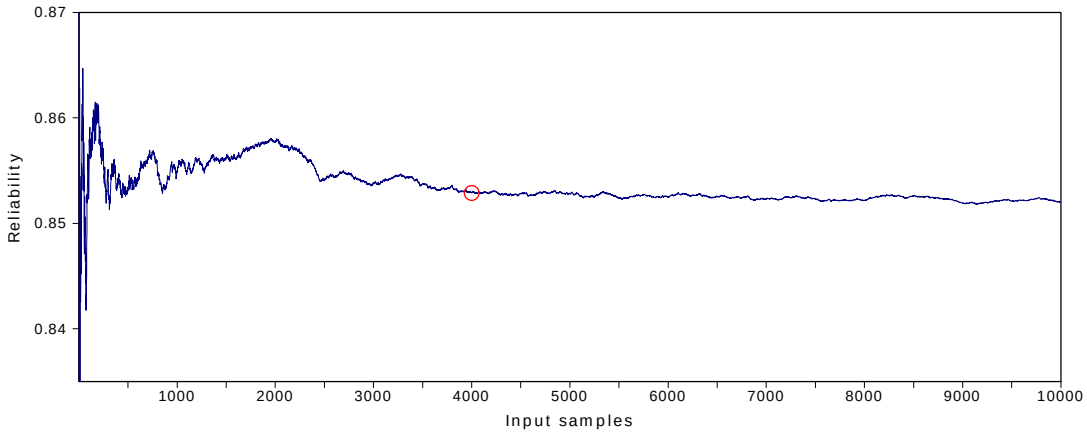


FIG. 18 – Fiabilité moyenne d'un circuit en fonction du nombre d'échantillons.

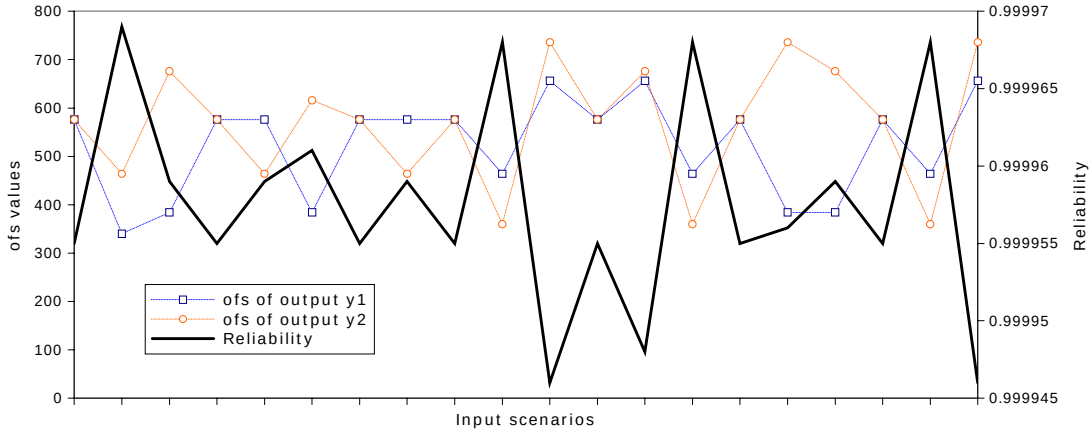


FIG. 19 – Profil de la fiabilité.

y1 et y2. Notez que la courbe de fiabilité est inversement proportionnelle au produit des deux valeurs ofs.

Afin d'évaluer la quantité de matériel supplémentaire requis par la modélisation SNaP, tous les circuits combinatoires modifiés ont été synthétisés et les résultats sont présentés dans la figure 20. La valeur gf utilisée est toujours 256. La synthèse a été réalisée à l'aide de RTL Compiler et d'une bibliothèque de cellules standard 65nm fourni par ST-Microelectronics. En fait, le Verilog modifié est destiné à une utilisation dans un FPGA. Ainsi, les valeurs présentées ici représentent juste une tendance.

Compte tenu des choix empiriques pris, il est important de vérifier si la méthode produit des chiffres de fiabilité raisonnables. Pour cet objectif, nous avons effectué une comparaison avec la méthode SPR. Toutes les cellules de la modélisation SPR ont été considérées avec $q = 0;99999$. L'équivalent a été fait pour SNaP, dans lequel chaque cellule a été créée avec $gf = 10$ (courbe rouge) ou $gf = 256$ (courbe orange). Les résultats sont présentés sur la figure 21, à partir de laquelle sont vérifiées à peu près les mêmes tendances. Les résultats présentés dans la figure 21 ont été obtenus par simulation de 4000 échantillons d'entrée pour chaque circuit. La figure 21 montre que les deux méthodes concordent bien.

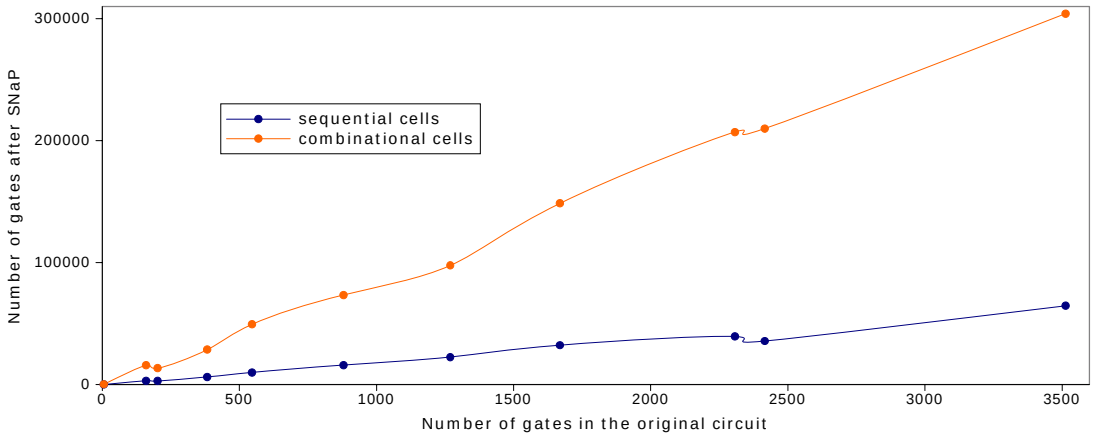


FIG. 20 – Tendances après synthèse.

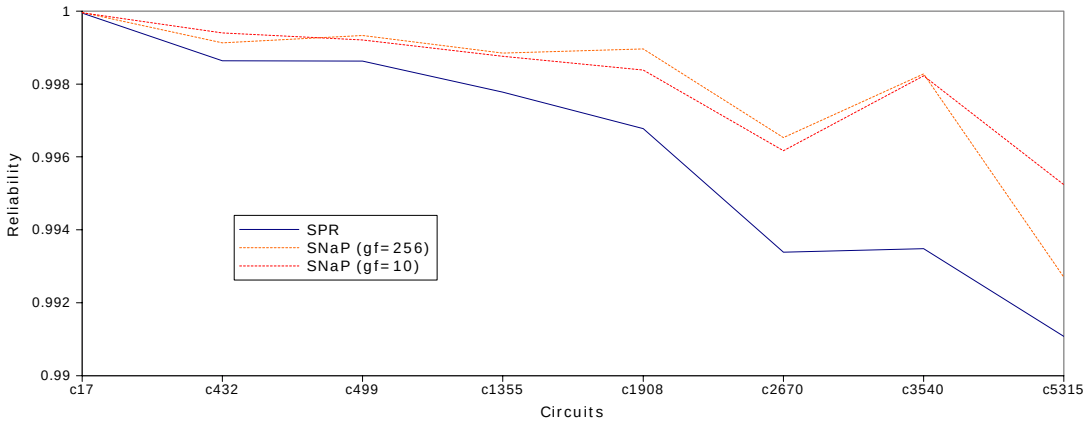


FIG. 21 – Comparaison entre SPR et SNaP.

Techniques d'Amélioration de Fiabilité

Cette section couvre une série de techniques utilisées pour améliorer la fiabilité d'un circuit donné. Ces techniques sont indissociables des méthodes d'analyse, c'est à dire, il n'y a aucune raison d'améliorer ce qui n'a pas besoin d'être amélioré. Et de nouveau, une fois que la technique a été appliquée, les méthodes sont utiles encore une fois pour estimer l'efficacité de son utilisation.

Cette section explore l'idée que les blocs d'un circuit numérique peuvent être classés en fonction de leurs importances par rapport à la fiabilité globale du circuit. Ce classement prend en compte le masquage logique. Avec la liste classée des blocs, il est possible d'appliquer un durcissement sélectif en utilisant des techniques de tolérance aux fautes.

Si nous considérons qu'un changement de la fiabilité d'un seul bloc b_i conduit à une nouvelle fiabilité q_i', alors la fiabilité du circuit devient R_i'. Etant donné que différents blocs b_i et b_j contribuent de manière différente à la fiabilité d'un circuit, des changements de blocs différents peuvent produire différentes valeurs R_i' et R_j'.

La méthodologie proposée ici suppose qu'il existe une technique capable d'améliorer la fiabilité d'un bloc donné de durcissement tels que $q_i' = 1$. Il ne s'agit pas d'une limi-

tation, c'est juste une simplification, d'autres valeurs sont également possibles. Ensuite, pour tous les blocs du circuit, une exécution de l'algorithme SPR est faite. Dans chaque exécution, un noeud b_i est sélectionné, q_i est défini comme 1, et la nouvelle valeur de la fiabilité R_i' est obtenue. Cet effort est possible uniquement car la complexité de l'algorithme SPR est linéaire.

Après l'exécution des analyses initiales, une liste de toutes les valeurs R_i' est obtenue. À cette etape, on peut trier la liste et sélectionner à durcir le bloc avec la plus grande R_i'. Néanmoins, l'intérêt ici est d'établir un compromis entre le coût de durssisement du bloc en question et ceux des autres blocs. Pour cela, un nouveau paramètre Ha_i est introduit, capable d'exprimer l'affinité de durcissement.

Le paramètre Ha_i de chaque type de cellule est défini par l'utilisateur. Il doit être limité dans l'intervalle [0,1]. Ce paramètre est générique et peut être utilisé pour exprimer tout type de compromis : surface, délai, puissance ou des combinaisons. Le Ha_i de la plus petite cellule dans une bibliothèque est considéré comme une valeur de référence et est toujours défini comme 1.

Une fois que l'affinité de chaque cellule est connue, il est nécessaire d'utiliser ces valeurs pour décider quel bloc devra être sélectionné pour le durcissement. Cette étape de la méthode présente une nouvelle valeur, le gain de fiabilité est donné par Rg_i. Il représente la différence entre la fiabilité de circuit avant (R) et après (R_i') le durcissement. Cette valeur est calculée de la manière suivante :

$$Rg_i = R_i' - R \tag{8}$$

La valeur de Rg_i obtenue est ensuite utilisée pour calculer le produit fiabilité-affinité comme suit :

$$Prh_i = Rg_i \cdot Ha_i \tag{9}$$

La méthodologie décrite a été appliquée à plusieurs circuits de référence ISCAS'85. Chaque bloc de chaque circuit a été consideré avec $q_i = 0,9999$. L'objectif d'augmentation de la fiabilité a été ajusté de sorte qu'une diminution de la non-fiabilité serait atteinte pour chaque circuit. Les résultats sont présentés dans les tableaux 2 et 3. Le premier tableau contient les résultats pour une réduction d'au moins 20% (par rapport à la non-fiabilité initiale) tandis que le second contient les résultats pour une réduction d'au moins 40%. La non-fiabilité initiale de chaque circuit est donnée dans la deuxième colonne des tableaux.

La figure 22 montre les circuits dont la méthodologie est effective. Les données sont les mêmes dans les tableaux, donc le même scénario s'applique : mêmes équations et le coût du voteur est négligé. Les valeurs de puissance indiquées sur l'axe des y sont normalisées par rapport à la puissance initiale de chaque circuit.

Une comparaison avec d'autres méthodes n'est pas simple, surtout car les objectifs sont généralement différents. Les résultats présentés dans d'autres travaux sont en alignement avec les résultats présentés dans ce travail, ce qui suggère que plusieurs fautes n'ont pas un grand impact sur la décision de quel nœud à durcir. Plusieurs fautes ont un impact considérable sur la fiabilité réelle d'un circuit. Ainsi, elles sont importantes pour déterminer les compromis entre le coût et la fiabilité.

Néanmoins, en termes qualitatifs, il est facile de remarquer que certaines cellules ont un impact plus important dans la fiabilité du circuit que d'autres. Cette observation est

TAB. 2 – Résultats pour une réduction d'au moins 20%.

Circuit	Non-fiabilité initial	Puissance (nW=MHz)	Sans affinité		Avec affinité	
			Cellules durcies	Puissance (nW=MHz)	Cellules durcies	Puissance (nW=MHz)
c17	0.000562	21498	1	21498	1	21498
74283	0.003848	189244	4	222932	8	189404
c432	0.013466	624686	9	624866	9	624866
c499	0.013611	1321460	20	1669540	41	1322280
c1355	0.021905	1907300	38	2179608	38	2179608
c1908	0.031668	2146539	58	2147699	58	2147699
c3540	0.062635	5.90e+06	54	5.90e+06	54	5.90e+06
c2670	0.064015	4.07e+06	41	4.12e+06	42	4.08e+06
c5315	0.085614	8.89e+06	59	8.96e+06	60	8.90e+06

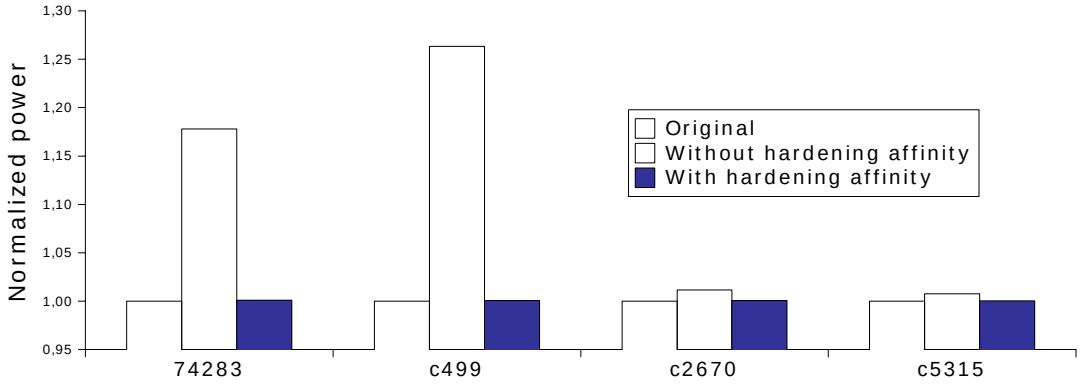


FIG. 22 – Valeurs de puissance normalisées pour le durcissement sélectif avec et sans affinité.

mise en évidence dans les résultats présentés ici. Il existe certains cas particuliers, comme celui illustré sur la figure 23, où le choix correct du noeud à durcir a un grand impact sur la fiabilité de l'ensemble du circuit. L'analyse représentée dans la figure 23 provient du circuit c1355.

En ce qui concerne la figure 23, elle contient les valeurs $R_i^{\square}$ liées au durcissement de toutes les cellules possibles. Les noeuds dans l'axe x sont ordonnés par le gain de fiabilité que le durcissement de ce nœud produirait. Le circuit a été évalué en supposant le paramètre $q_i = 0,9999$. En termes absolus, la différence entre le meilleur et le plus mauvais candidat n'est pas grande. Habituellement, plusieurs cellules sont sélectionnées pour le durcissement (comme dans le tableau 3), de sorte que ces valeurs s'accumulent. Ainsi, le choix du meilleur candidat pour le durcissement est critique.

TAB. 3 – Résultats pour une réduction d'au moins 40%.

Circuit	Non-fiabilité initial	Puissance (nW=MHz)	Sans affinité		Avec affinité	
			Cellules durcies	Puissance (nW=MHz)	Cellules durcies	Puissance (nW=MHz)
c17	0.000562	21498	2	35830	2	35830
74283	0.003848	189244	10	273464	16	189564
c432	0.013466	624686	26	625206	26	625206
c499	0.013611	1.32e+06	48	2.15e+06	80	1.42e+06
c1355	0.021905	1.90e+06	83	2.50e+06	83	2.50e+06
c1908	0.031668	2.14e+06	132	2.14e+06	132	2.14e+06
c3540	0.062635	5.90e+06	175	5.90e+06	175	5.90e+06
c2670	0.064015	4.07e+06	128	4.22e+06	128	4.08e+06
c5315	0.085614	8.89e+06	205	9.13e+06	207	8.90e+06

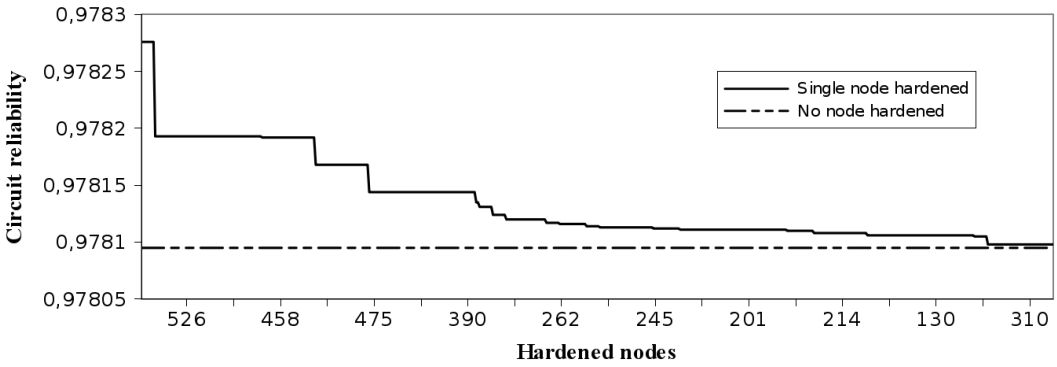


FIG. 23 – Gain de fiabilité par rapport au noeud choisi pour le durcissement.

Net Hardening

Il a été montré dans la section précédente qu'une solution basée sur les coûts peut réduire la quantité de durcissement supplémentaire requise par une technique de tolérance aux fautes. Il a également été démontré que l'occurrence de plusieurs fautes est plus fréquente et par conséquent doit être correctement gérée.

En ce qui concerne les fautes multiples, leur source détermine le profil de la localité. Les fautes multiples causées par SEEs ont toujours un biais de localité. Ce qui est présenté dans cette section est une version modifiée de la méthode de durcissement afin de tenir compte de cela.

Quand un circuit numérique est conçu en utilisant des cellules standard (standard cells), une étape de placement est exécutée. Les portes qui sont logiquement connectées ont une certaine probabilité d'être effectivement physiquement proches car les algorithmes de placement tentent de réduire la longueur des fils (wirelength). Etant donné que ces cellules sont suffisamment proches les unes des autres, elles peuvent être sensibles aux effets de charge sharing.

L'illustration de la figure 24 représente un scénario dans lequel des fautes multiples peuvent se produire. L'image 24 montre trois rangées de cellules standard et la cellule ANDX0 dans la première rangée est considérée comme le site de collision d'une particule énergétique. La zone sous le cercle rouge représente la région de voisinage du nœud frappé. Ce voisinage est sensible, c'est à dire, le cercle rouge représente le nuage de partage de charge (charge sharing) . Les cellules en jaune sont celles qui pourraient être touchées (c'est à dire qu'elles sont à l'intérieur du rayon considéré et ainsi leurs sorties pourraient être erronées). Les cellules représentées en bleu clair sont celles non affectées.

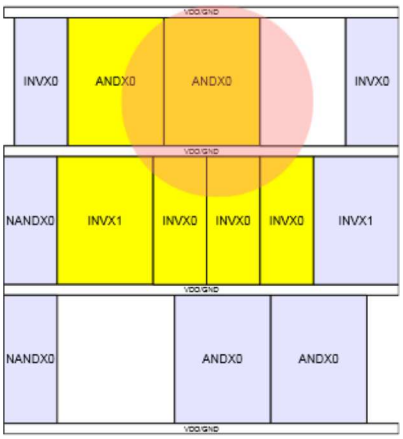


FIG. 24 – Représentation de fautes multiples selon le nuage de partage de charge.

Les fautes aléatoires multiples sont présentées dans la figure 25. Cette approche peut facilement surestimer la sensibilité du circuit réel aux SEEs. Un tel scénario peu réaliste a été utilisé dans la littérature.

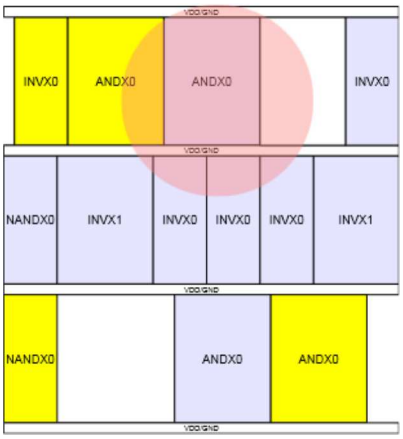


FIG. 25 – Représentation de fautes multiples aléatoires.

L'utilisation d'un biais de localité a été introduite lors de l'exécution du durcissement sélectif. Le biais est utilisé ici comme une heuristique et est introduit à travers la notion de net hardening. Au lieu de durcir une cellule unique ou un ensemble de cellules aléatoires, les nets sont considérés. Durcir un net c'est durcir toutes les cellules qui sont logiquement connectées à lui. Ceci est représenté sur la figure 26.

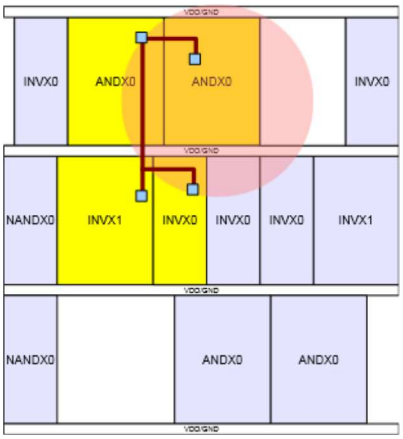


FIG. 26 – Représentation de fautes multiples selon net hardening.

Dans une première expérience, le but du durcissement a été fixé pour obtenir une augmentation relative d'au moins 10% de la fiabilité des circuits. Les résultats sont présentés dans le tableau 4, dans lequel les chiffres en gras mettent en évidence les scénarios où la méthode a été plus efficace.

TAB. 4 – Résultats pour une augmentation relative de la fiabilité de 10%.

Circuit	Surface (μm^2)	Sans affinité			Avec affinité		
		Cellules durcies	Surface (μm^2)	Aug. de surface	Cellules durcies	Surface (μm^2)	Aug. de surface
c17	33.1	3	66.3	100%	1	44.2	33.3%
74283	306.5	6	405.2	32.2%	3	339.7	10.8%
c432	1134.4	4	1209.6	6.6%	4	1209.6	6.6%
c499	2155.1	26	2579.6	19.6%	15	2407.1	11.6%
c1355	3194.7	43	3872.1	21.2%	24	3460.1	8.3%
c1908	5273.7	48	6186.7	17.3%	35	5660.8	7.3%
c3540	10855.2	61	11688.3	7.6%	30	11240.4	3.5%
c2670	8018.0	38	8602.4	7.2%	28	8419.9	5.0%
c5315	15293.6	85	16583.8	8.4%	43	15794.9	3.2%

On peut remarquer que les pourcentages d'amélioration de la fiabilité indiqués dans le tableau 4 ne sont pas importants. Néanmoins, il faut souligner qu'ils sont adéquats pour un scénario dans lequel il y a un budget de durcissement réduit.

Conclusion

Cette thèse a porté sur deux préoccupations principales relatives à la fiabilité des circuits : l'analyse et l'amélioration. Quand il s'agit de méthodes d'analyse de fiabilité, il est

clair que la littérature a été enrichie par plusieurs travaux les dernières années. La simulation s'est établie comme la méthode préférée même avec ses limitations. D'autres solutions comme PTM et SPR-MP ont leurs mérites aussi. Les méthodes présentées dans cette thèse peuvent être facilement adoptées dans un flot de conception traditionnel. SPR+ ainsi que SNaP peuvent obtenir des chiffres de fiabilité en quelques secondes, même en considérant un circuit relativement complexe.

Cette thèse a également apporté un éclaircissement sur les techniques d'amélioration de la fiabilité des circuits. L'idée d'utiliser une fonction de coût pour décider quelles portes à durcir est le cœur des techniques proposées ici. Les résultats indiquent clairement comment les économies peuvent être obtenues.

La plupart des sujets abordés dans cette thèse ont été publiés dans les forums appropriés. Une liste complète de ces publications figure dans l'annexe D.

List of Acronyms

ASICs	Application Specific Integrated Circuits
AVF	Architecture Vulnerability Factor
DICE	Dual Interlock Cell
DRAM	Dynamic Random-access Memory
ECC	Error Correcting Code
ESD	Electrostatic Discharge
FIT	Failure-in-time
FPGAs	Field Programmable Gate Arrays
FSMs	Finite State Machines
GUI	Graphical User Interface
HBD	Hardening by Design
ICs	Integrated Circuits
ITM	Ideal Transfer Matrix
ITRS	International Technology Roadmap for Semiconductors
LET	Linear Energy Transfer
MOSFET	Metal-oxide-semiconductor Field-effect Transistor
MTBF	Mean Time Between Failures
NMR	N-Modular Redundancy
NRE	Non-recurring Engineering
PBR	Probabilistic Binomial Reliability
PDD	Probabilistic Decision Diagram
PGM	Probabilistic Gate Model
PLL	Phase-locked Loop

PQ	Pulse Quenching
PTM	Probabilistic Transfer Matrices
RTL	Register Transfer Level
SEE	Single Event Effect
SEL	Single Event Latchup
SER	Soft Error Rate
SETs	Single Event Transients
SEUs	Single Event Upsets
SOI	Silicon-on-Insulator
SPICE	Simulation Program with Integrated Circuit Emphasis
SPR	Signal Probability Reliability
SPR-DWAA	SPR Dynamic Weighted Averaging Algorithm
SPR-MP	SPR Multi Pass
SRAM	Static Random-access memory
TMR	Triple Modular Redundancy
WAA	Weighted Averaging Heuristic

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last years a continuous development has been observed in the domains of electronic systems and computers. These systems are usually composed by a large number of smaller systems referred as Integrated Circuits (ICs). The technology used to produce such ICs has shifted in the last years in a process known as scaling, i.e., the actual size of a chip is approximately the same but the (number of) transistors embedded in it are quite numerous nowadays. Figure 1.1 depicts a year versus area comparison from International Technology Roadmap for Semiconductors (ITRS) [1], in which the evolution of different devices present within an IC is represented.

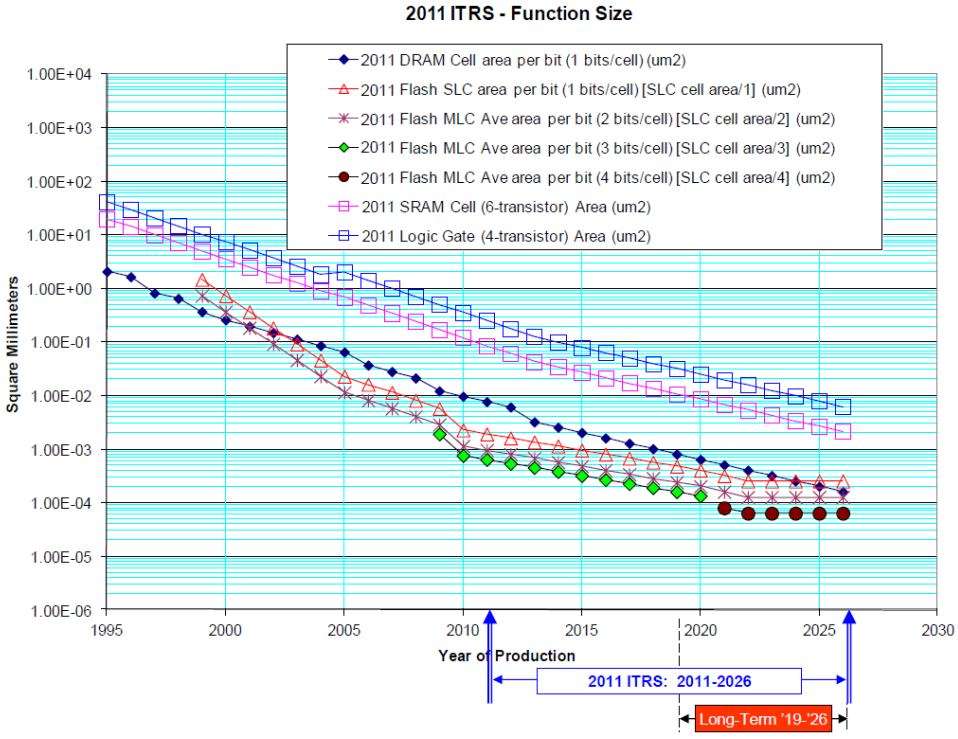


Figure 1.1: Evolution trends of different devices from ITRS [1].

The illustration in Fig. 1.1 shows the scaling trend of four different devices: Dynamic Random-access Memory (DRAM), FLASH memory, Static Random-access mem-

ory (SRAM) and logic gates. DRAMs, SRAMs and logic gates have scaling trends that are near a 70% size reduction every 2-3 years (half-pitch measured). The evolution of FLASH memories has been even faster than that.

One issue of vital importance is the reliability of these ICs and systems, especially the ones that are used in dependable computing environments. Such environments are characterized by strict requirements of a given attribute. Examples of this attribute are reliability, availability, safety, security, survivability and maintainability. These attributes are described in details in Section 1.1.

This thesis has a particular interest in the analysis of dependability attributes for a special class of ICs: digital circuits. This type of circuit is used to build most devices that are present in our daily lives, like mobile phones, computers, cameras, etc. Figure 1.2 illustrates one of the possible ways a digital circuit can be internally organized [2, 3]:

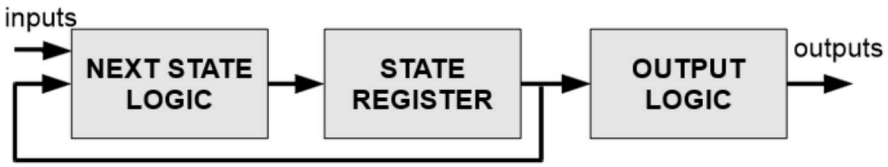


Figure 1.2: Block diagram of a digital circuit including its sequential and combinational parts.

The illustration in Fig. 1.2 shows the inputs and outputs of a circuit, as well as the internal logic for the current state and the next state. The current state logic is stored in the memory elements and is referred as sequential logic. The next state logic does not store data, it actually computes data based on inputs and the current state; this type of logic is referred as combinational or combinatorial logic. A building scheme like this one, using sequential and combinational logic, is replicated numerous times to build large circuits. This type of construction is usually referred as Finite State Machines (FSMs), which are used to control the flow of data. Moreover, the relevant information here is that, regardless of the type, the logic elements are not completely reliable, as it will be later explained in Section 1.2.

Today's hardware designers are faced with difficult decisions arising from conflicting efficiency and time-to-market pressures. In general terms, standard-cell based Application Specific Integrated Circuits (ASICs) offer the best density, performance and power but have long design times, high Non-recurring Engineering (NRE) costs, and increasingly difficult verification cycles. Field Programmable Gate Arrays (FPGAs) offer zero NRE cost but higher cost per unit and poorer density, performance and power when compared to ASICs [4]. Additionally, regardless of the choice made for a given project, i.e., to design using FPGAs or ASICs, dependability can be an important criterion during the development process.

1.1 Dependability

According to Avizienis [5], an electronic system can be characterized by four properties: functionality, performance, cost and dependability. The first three properties are very naturally tied one to each other, so a trade-off between these properties is established. This trade-off is well known among designers and companies. Yet, dependability has

also to be considered in certain scenarios, which adds an additional component to an equation that is already quite complicated.

Dependability of a computing system is its ability to deliver service that can justifiably be trusted. A full taxonomy of dependability and its related concepts is shown in Fig. 1.3. These concepts are divided into threats, attributes and means.

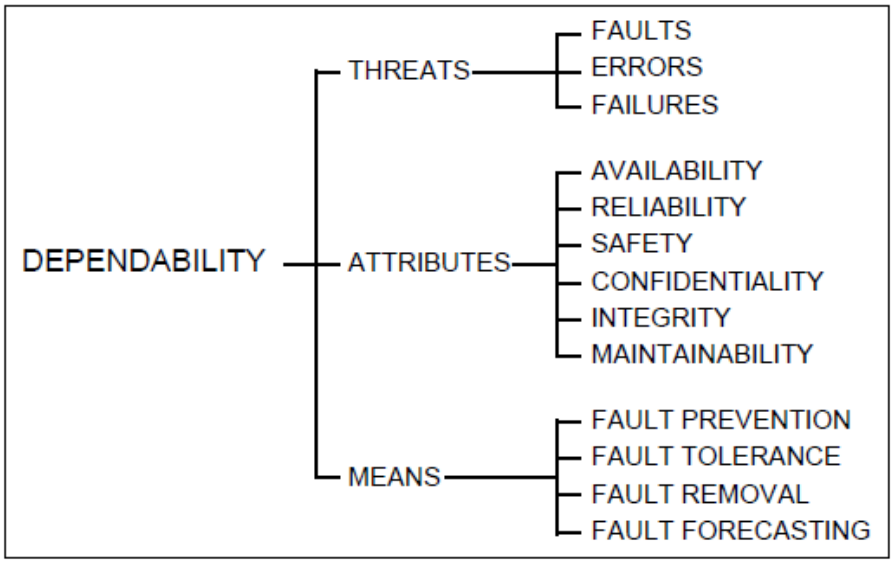


Figure 1.3: Taxonomy of dependability and its related concepts.

Threats are the sources of dependability issues. These are events that are known to affect one of the attributes of a dependable system or circuit. The creation mechanism, the type, and other characteristics of a fault are strictly dependent on the system or application being considered in a dependability analysis. Fault profile may change completely from one domain to another. Some threats and types of threats that are sources of unreliability in digital circuits are presented in Section 1.2.

Regardless of the domain being considered, there is a relationship between threats. This relationship is illustrated in Fig. 1.4. In simple words: a fault might activate an error, while an error might be propagated to cause a failure. Such failure then might represent a fault in a larger system. So, the process of activation and propagation continues until a point where it might actually achieve visibility in the system as a whole, causing an erroneous or non-satisfactory functioning.

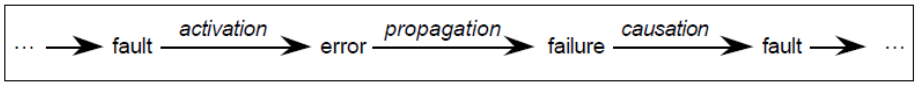


Figure 1.4: Chain of threats and threat propagation.

As previously mentioned, threats are events that affect the attributes of a system. Dependability itself is an integrative concept since it must encompass all the (meaningful) attributes of a system. A brief description of each of these attributes is given below [5]:

Availability Readiness for correct service.

Reliability Continuity of correct service.

Safety Absence of catastrophic consequences on the user(s) and the environment.

Confidentiality Absence of unauthorized disclosure of information.

Integrity Absence of improper system state alterations.

Maintainability Ability to undergo repairs and modifications.

In the context of this thesis a single attribute is considered, the reliability. The following section details such attribute from the point of view of a digital circuit. Some fault tolerance techniques are also discussed in Section 2.5.

1.2 Reliability in Digital Circuits

The advances in the semiconductor industry have deeply increased the performance of digital circuits. Most of this performance gain has been due to small dimensions and low voltage transistors, which have led to complex architectures with large parallelism combined with high frequency [6].

However, the same technology that made all this progress possible, has also reduced transistor reliability by reducing threshold voltage and tightening the noise margins [7, 8] and thus making transistors more susceptible to faults of different sources. All standard elements of a digital circuit are built using networks of transistors. Thus, a low-reliability transistor causes the whole circuit to have a low reliability as well.

Faults that affect a digital circuit are classified due to its behavior as follows:

Permanent faults which affect the characteristics of the circuit structure in a way that is not repairable.

Intermittent faults are the ones that cause an erratic behavior of the device. Such behavior is difficult to identify and repair since it seems to appear in intervals.

Transient faults are usually related to environmental conditions and tend to be harder to diagnose and repair. Common sources of this type of faults are strikes of alpha and neutron particles, crosstalk, Electrostatic Discharge (ESD), etc. This type of fault is also referred as a soft error.

For each type of fault, different strategies are applied to detect and correct (when and if possible). For permanent faults the most notorious detection scheme in the context of digital circuits is the application of input vectors during IC testing. By applying a given combination of values in the inputs of a circuit and using an appropriate fault model, it is possible to detect if a certain node is functional or not.

One of the most significant contributions in the field of IC testing and fault modelling was given by Eldred in [9] and by Galey et al. in [10]. These authors have developed a stuck-at fault model, from which it is possible to infer if a given node is stuck-at-one or stuck-at-zero. Although some nodes are extremely hard to prove to be fully functional, this technique and variations of it have been applied for a long time and are known for increasing the actual quality of produced silicon. Fault models most likely to gain significance in the near future are the delay fault models [11].

1.2.1 Defects

In the manufacture process of semiconductor-based ICs, a large amount of electronic devices is produced simultaneously in a series of very complex processing steps. The probability that all such devices and their interconnections will function accordingly depends on the level of control exercised in their manufacture. The fraction of chips that, upon completion of manufacture, can meet a set of test requirements is called the yield [12].

Manufacturing defects can be roughly classified into two big types: gross area defects (or global defects) and spot defects [13, 14]. Global defects are considered large-scale defects coming from issues on one of the steps of the manufacture process (e.g., mask misalignment, excessive etching, etc.). Spot defects have more of a random source associated with the quality of the materials used in the manufacture process. Impurities in the materials deposited on the chip can cause such spot defects. Both types of defects cause loss of yield but the former can be handled easily than the latter by properly controlling the fabrication process.

Furthermore, the occurrence of global defects is not related to die size. The same is not true concerning spot defects. Thus, the scaling process brought by the introduction of new technologies can increase the amount of spot defects. Spot defects can be classified according to location and potential harm they may cause. Typical examples are open and shorts due to missing or extra patterns. Spot defects can occur between layers or in the same layer. One example of an open defect is shown in Fig. 1.5, which shows the top-down (a) and cross-section (b) of an open in the M2 layer (metal 2).

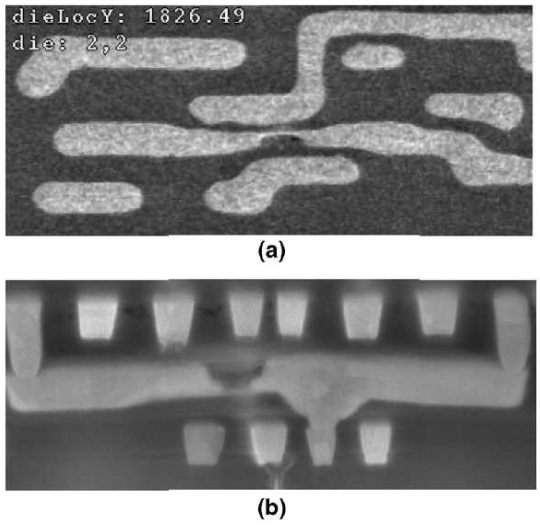


Figure 1.5: Top-down (a) and cross-section (b) view of an open defect [14].

Not all physical defects result in faults. Any imperfection in the wafer can be considered a physical defect while only the ones affecting the circuit operation are considered faults. A defect that causes a change in a continuous parameter of the circuit is referred as a parametric fault [15], i.e., the circuit is functional but do not respect its operating window or expected performance.

1.2.2 Transient Faults

When a single radiation ionizing particle strikes the silicon it interacts with it and this phenomena is known as a Single Event Effect (SEE) [1]. SEEs can be destructive and non-destructive. An example of destructive effect is Single Event Latchup (SEL) that results in a high operating current, above device specifications [16]. If no damage is done than the SEL effect must be cleared by a power reset. Silicon-on-Insulator (SOI) technologies are more and more used nowadays and one of the reasons is that this type of technology is immune to SEL [17]. SEL and other destructive effects are not part of the scope of this thesis.

One of the major concerns related to SEEs are the soft errors, which can be defined as a transient effect (or simply a fault) provoked by the interaction of energized particles with the PN junctions of a circuit. This type of upset temporally charges or discharges nodes of the circuit, generating transient pulses that can be interpreted as valid internal signals, thus provoking an erroneous result [18]. The most typical errors concerning soft errors are Single Event Upsets (SEUs), which are bit-flips in the sequential elements of a circuit. Another type of error is referred as Single Event Transients (SETs), which are transient pulses in the combinational logic. Such SETs then might be registered by the sequential portion of the circuit and, depending on a series of factors, can achieve the same (potentially severe) effects of an SEU.

Figure 1.6 shows the effect of an ionizing particle when it strikes a silicon junction. It is shown how the charge generated in the silicon substrate is collected at a reverse-biased junction. More than one charge transport mechanism might be involved, depending on the underlying technology and circuit design. The image shows two different mechanisms referred as drift and diffusion currents. The first mechanism is electric field driven and happens very quickly while the latter is not as fast.

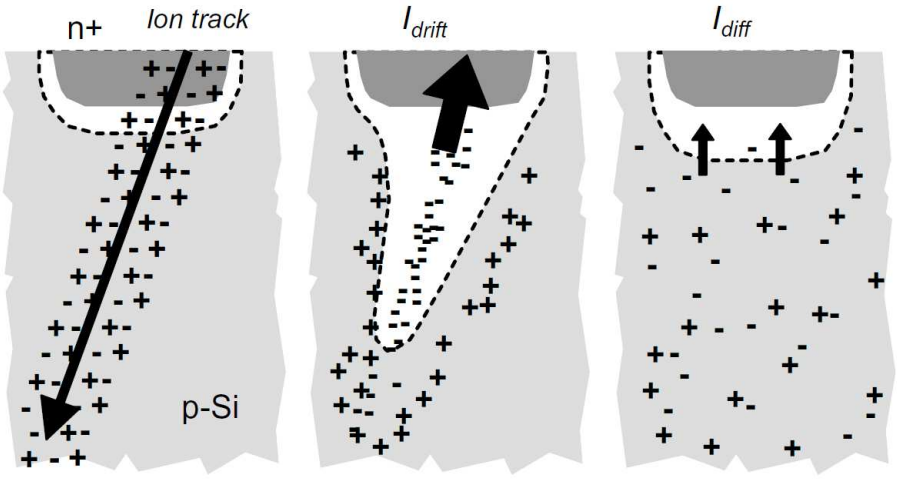


Figure 1.6: Effect of an ionizing particle in a silicon junction [19].

Initially, a cylindrical track of electron-hole pairs is formed by the interaction of the ionizing particle with the silicon. The silicon 'responds' to that interaction and creates a

[1] Some authors use different meanings for the terms SEE, SET and SEU. The convention used in this paragraph is maintained through the remainder of this text.

funnel shaped distribution of charge. Nevertheless, this shape cannot hold and eventually collapses. That is when the secondary mechanism of diffusion takes place. Although both mechanisms have different characteristics, they do have the same effect: carriers are collected at the junction. If the collected charge is bigger than the critical charge (minimum amount of charge that needs to be deposited by a particle strike to produce a transient capable of shifting a logic value), it is then perceived by the circuit as a current pulse. This is illustrated in Fig. 1.7.

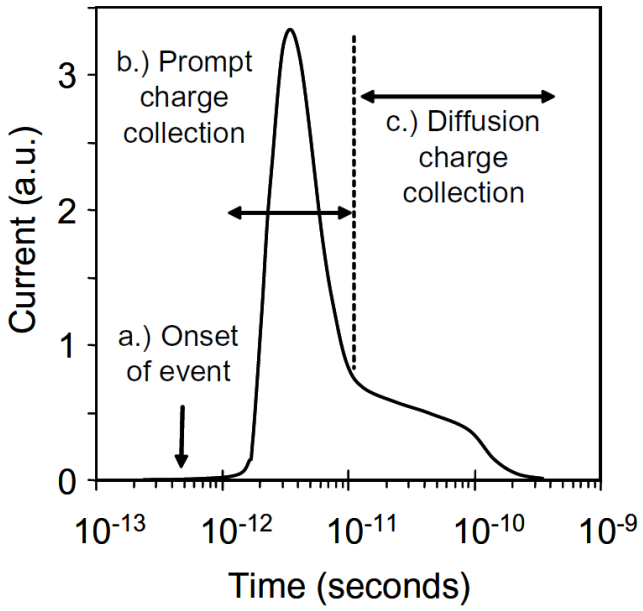


Figure 1.7: Current pulse at the junction and the involved collection mechanisms [19].

Figure 1.7 depicts the moment of the particle strike as well as both charge transport mechanisms. Since the drift current mechanism is relatively quick (in the order of picoseconds), a fast pulse of current is generated. When the diffusion mechanism starts to take place, it is not as fast (order of nanoseconds). Thus, the generated current pulse changes into a (long) tail shape.

Soft Error Rate (SER) is the rate at which a device (or system) encounters or is predicted to encounter soft errors like the ones described above. This rate is expressed by the number of failures over time. The unit used to quantify it is referred as Failure-in-time (FIT), and it is equivalent to 1 error per billion hours of operation. SER can also be measured by Mean Time Between Failures (MTBF).

Previous research shows that SER per chip is increasing substantially, mainly due to the fast growth of the number of transistors on a single chip [20–23]. Rates per flip-flop or per memory cell shift from technology to technology and some rates can be decreasing [24]. Those are usually counterparted by the increase of the overall number of transistors.

When considering advanced technology nodes, a single transient for each particle strike is not always valid, because a particle may affect multiple nodes in a circuit through the process of charge sharing [25]. That process also increases the SER per chip since one single particle hit can cause multiple SET pulses in the combinational logic or multiple SEUs in the registers and memories [26]. Charge sharing is an important issue because it can turn circuit-level hardening techniques completely ineffective. More details concern-

ing charge sharing are given in sections 4.2 and 4.4.

Multiple fault models must be taken into account to analyze the error rate in integrated circuits. Error rate and cross-section calculations may be significantly smaller than the ones observed in the actual circuit when multiple transients are not accounted for. This fact is highlighted in [25, 26].

Another recent issue with charge sharing is that it can, in very specific scenarios, be beneficial. Instead of increasing the circuit error rate, it can be used to make it smaller through a mechanism termed Pulse Quenching (PQ) [27]. This mechanism is studied and detailed in Section 4.4.

1.3 Masking

Digital circuits have particular masking properties that change the way the reliability of the circuit is perceived in a system, for example. In fact, some threats can be severely reduced by these properties. The following sections detail four different masking mechanisms. A detailed discussion on how these masking properties are expected to behave with the advent of newer technologies is given in [23] and also in [28].

1.3.1 Electrical Masking

Electrical masking is the property by which an electrical pulse generated as a result of a particle strike within a combinational logic circuit gets attenuated by subsequent gates in the logic path [29, 30]. The illustration in Fig. 1.8 exemplifies the effect of electrical masking [31].

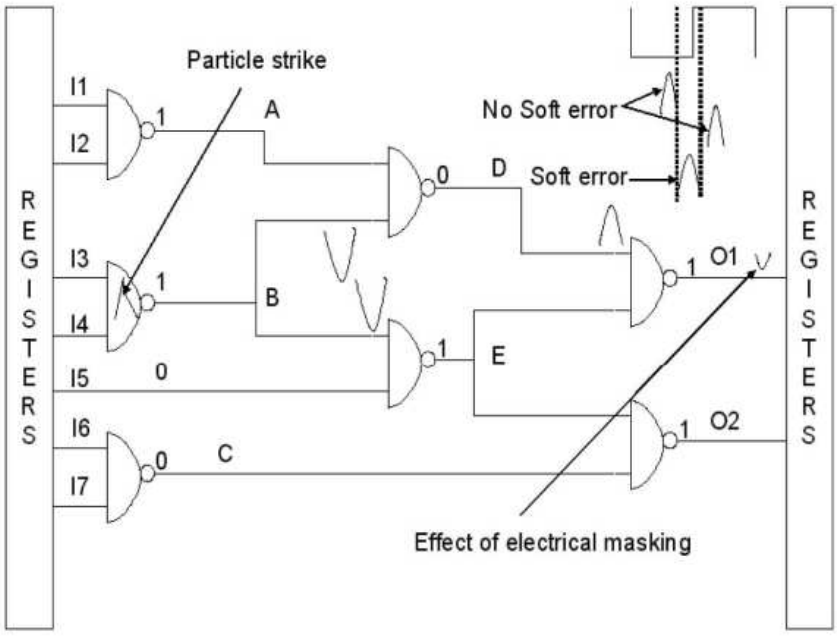


Figure 1.8: Electrical, temporal and logical masking properties in a digital circuit stroke by a particle [31].

The assumption in the image is that a particle stroke a NAND2 gate, but the idea can be generalized for other types of threats and other types of gates. Once the node is stroke, the erroneous value is propagated through the other elements of the circuit until it reaches a register element. It can be seen that the current pulse diminishes while it is propagated through the nodes.

The explanation for such effect is that electrical masking is the composition of two electrical effects that reduce the strength of a pulse as it passes through a logic gate. Circuit delays caused by the switching time of the transistors cause the rise and fall time of the pulse to increase. Also, the amplitude of a pulse with short duration may decrease since the gate may start to turn off before the output reaches its full amplitude. The combination of these two effects reduces the duration of a pulse, making it less likely to cause a soft error. The effect cascades from one gate to the next because at each gate the slope decreases and hence the amplitude also decreases [23].

Several authors have discussed this type of masking effect from different points of views. In [32] and other works, authors claim that electrical (and also temporal masking) are fairly well-understood and can easily be quantified. Performing classical electric simulation is fairly simple and can indeed calculate the masking capability of a circuit. Some other authors work with the actual masking modelling so electrical simulation can be avoided. In [33], the authors use table lookup Metal-oxide-semiconductor Field-effect Transistor (MOSFET) models to accurately capture the nonlinear properties of submicron transistors. Based on these models, they propose and validate the transient pulse generation model and propagation model for error rate analysis. It is possible to confirm that the pulse generated by the model matches well with that obtained using Simulation Program with Integrated Circuit Emphasis (SPICE).

The current trend for the newer technologies is that this masking property will be less and less effective since the transistors are continually scaled to smaller feature sizes, so the pulse attenuation effect is also decreased [23, 34].

1.3.2 Temporal Masking

The effect of temporal masking, also known as latching-window masking, is also shown in Fig. 1.8, in the top right corner. Digital circuits have a peculiar property that distinguishes them: the signals are updated by the combinational logic until they stabilize and are captured by the sequential logic. Yet, the capture mechanism only happens with a certain frequency. So it is possible that a certain threat occurs in a time frame where it will not be sampled by the capture logic of a register, for example.

Regarding such register element, let us consider that it is a type D flip-flop. Then the timing constraints associated with each edge-triggered D flip-flop are as follows:

- The data (D) input has to receive all data before the setup time (Ts) preceding the latching clock edge.
- The data input must be held steady for the duration of the hold time (Th) following the latching clock edge.

Soft errors are usually characterized by a transient glitch of a certain duration that results from a particle strike. If such a glitch is present at the data or clock inputs of a flip-flop during the whole interval [Ts,Th], it can result in an incorrect value being latched. If the glitch is present during the setup or hold time, it can prevent a correct value from

being latched. Therefore the effective interval in which a soft error is perceived by a register element is simply [Ts,Th] [35].

Also in [35], the authors developed both static and statistical analysis techniques to estimate timing masking through the error-latching window of each gate. It is then shown that the SER analysis performed indicates that 62% of gates identified as error-critical using timing masking would not be identifiable by considering only logic masking. In [34], the authors combine electrical, temporal and logical masking using SET descriptors for the elements of the circuit and SPICE simulation for determining the actual probability that a certain SET will be latched.

For the faults that are generated by SETs this masking property is important since not all current peaks will actually reach a memory element. Yet, the frequency in the modern circuits is increasing while the actual pulse width of a SET has not changed. So, in relative terms, temporal masking is no longer that much effective since there is more signal sampling in a time frame, which increases the occurrence of errors.

1.3.3 Logical Masking

Logical masking accounts for the lack of sensitized paths (from the erroneous node) to a primary output or memory element. Among the masking phenomena that render immunity to combinational logic circuits from soft errors, logical masking is the hardest to model and characterize [32].

Figure 1.8 shows the effect of logical masking on node E. Since the I5 input is zero, there is no path going through the NAND gate that can propagate the transient coming from node B.

One particular detail regarding logical masking is of fundamental importance: it is technology-independent. Thus, in the following chapters we focus on calculating the reliability of a circuit considering only the logical masking. The other masking properties can be interpreted as derating factors, i.e., by calculating only the logical masking we are providing a underestimate of the actual circuit reliability. Electrical and latching-window masking computations may actually filter out some of the errors that logical masking does not filter.

1.3.4 System-level Masking

Let us consider a processor as a system capable of performing system-level masking (sometimes also referred as architecture-level masking). If a particle strike causes a bit to flip or a piece of logic to generate a wrong result, let us refer to it as a raw soft error event. Fortunately, not all raw soft errors cause the processor to fail. In a given cycle only a fraction of the bits in a processor storage structure and some of the logic structures will affect the execution of the current instruction. A raw error event that does not affect these critical bits or logic structures has no adverse effect on the outcome of the executing program and is said to be masked. For example, a soft error in the branch prediction unit or in an idle functional unit will not cause the program to fail [20].

Previous research has shown that there is a large masking effect at the system-level. In [36] the authors demonstrate the effects (propagation and manifestation) of gate-level faults on program-level behavior through a gate-level simulation of a processor executing two representative application tasks in the presence of fault. Coverage values for several error detection mechanisms are presented, along with the associated detection latencies.

Error propagation is observed and regions where error detection probabilities are the highest are identified.

In [37] it is demonstrated that highly error-sensitive blocks are common for various workloads. At the same time soft errors in many other logic blocks rarely affect the computation integrity. The results show that a reasonable prediction of the soft error sensitivity is possible by deduction from the processor's microarchitecture. It is also demonstrated that the sensitivity-based integrity checking strategy can be an efficient way to improve fault coverage per unit redundancy.

Recently, there has been significant work motivated by the need to estimate the Architecture Vulnerability Factor (AVF) at runtime. These studies are motivated by the observation that the AVF may vary significantly across different applications or even during different phases of the same application [38, 39]. Thus, depending on the workload, the processor may be more or less vulnerable. This creates new opportunities to reduce the overhead introduced by the soft error protection while still meeting any hardness/ fault tolerance goal with a smaller penalty in energy, for example. If it is possible to estimate AVF in real-time, then it is also possible to adjust the protection scheme based on the current AVF value.

1.4 Organization of the Thesis

This thesis is organized into three main chapters, divided as follows:

Chapter 2 describes the state of the art regarding analysis and prediction of the reliability of digital circuits. Fault tolerance techniques are also detailed in this chapter. Several works from different authors are examined and discussed.

Chapter 3 covers a series of works that were done concerning methods for the analysis of a circuit's reliability. Two methods used to assess the reliability of circuits are presented. The first method is termed SPR+ and it is an extension of an already existing method for assessing the reliability of combinational logic only. Through the use of some heuristics, a new algorithm is proposed which takes first order reconvergence into account. The second proposed method is termed SNaP and it overcomes most of the limitations of other methods proposed in the literature by using an hybrid approach. In other words, it partially simulation-based while also being and analytical solution.

Chapter 4 details the works that were done concerning techniques for improving the reliability of digital circuits. In those works, different metrics for selecting which gates should be selectively hardened were applied, taking into account the gate's criticality in terms of reliability as well as the cost of hardening that given gate. Techniques adapted for analyzing multiple faults were also proposed and studied in details.

Chapter 2

State of the Art

Reliability analysis of electronic components deals with two very different aspects: reliability prediction and reliability measurement. These two aspects of reliability analysis are equally important in the design process, since reliability measurements enable validation and refinements in the high-level reliability models used for reliability prediction [40]. Thus, it is correct to say that a reasonable prediction of reliability must also come with a good measurement of it. In other words, no high-level models can be made without knowledge of the low-level issues.

The focus of the work later presented in this thesis is mostly on reliability prediction, i.e., it is assumed that there is another process used to characterize the reliability of the individual elements that compose a circuit. This is also true for the analytical techniques presented in this chapter.

Moreover, the interest of the work is targetted to the useful life period of the circuit. In this period the occurrence of faults is highly related to random nature sources. This eliminates the need to perform reliability analysis concerning infant mortality of ICs [41] and the tests related to it. Infant mortality is related to manufacturing issues like deviation from standard parameters and variability in general. Reliability analysis concerning wearout mechanisms like electromigration, hot carriers, dielectric breakdown [42] is also out of the scope of this thesis.

Given this introduction, this chapter follows with a study of state-of-the-art techniques used to estimate the reliability of a digital circuit. Some techniques only apply to combinational logic while others are of more general application. Concerning combinational logic, our focus is in the logical masking properties of it. It is well known that estimating logical masking is far more complicated than estimating electrical and temporal masking [32].

2.1 Simulation-based Fault Injection

Simulation-based fault injection, or simply fault injection, is a very simplistic and intuitive approach for estimating the reliability of a circuit. Due to its simplicity, it has received a fairly high attention from researchers [43]. The process starts by picking a node (block, cell or transistor, depending on the granularity of the analysis) and then proceeds into shifting its output value for a given time. Usually two versions of the same circuit are simulated at the same time: a fault-free version (or golden version) and a fault-prone version. The simulation then verifies if the outputs of the fault-prone circuit have deviated

from the expected value.

Whenever possible, the process described above is repeated for all nodes. Some form of metric is then applied to measure how reliable the circuit is. For instance, one could take the ratio between seen and unseen errors. That ratio is a measurement of the circuit capability of performing logical masking and therefore reflects the circuit reliability.

In [43, 44], the authors highlight some scenarios in which this type of simulation-based solution is appropriated (and when it is not). Some scenarios of particular interest are the representation of storage data corruption (such as registers, memories and disks) and communication data corruption (such as a bus or a network).

Many of the works in the current literature make use of fault injection into Register Transfer Level (RTL) modelled circuits. In other words, this means that fault injection is performed into the RTL model of a system, usually described in a language such as Verilog [45] or VHDL [46]. There are advantages and disadvantages of working at this level of description. For instance, in RTL the gates that actually compose the circuit are not known since the circuit has not undergone synthesis yet. At the same time, this can be seen as an advantage since RTL simulation is much simpler and faster than gate-level simulation, either logical or electrical.

In [47], the authors presented a tool referred as MEFISTO. This tool was used to demonstrate that simulation-based fault injection provides perfect controllability over where and when a fault is injected, in contrast to the heavy-ion technique. A similar tool is used by Massengill et al. [48] to identify the sensitive regions of a microprocessor. As a matter of fact, microprocessors are a perfect study-case for simulation-based fault injection since they are prone for injection of data corruption behaviors [6, 49, 50].

As a positive characteristic, the fault injection by simulation approach is simple and requires only a traditional simulation engine (or a slightly modified version of it). The comparison between the actual circuit response and the expected result is very straightforward. Thus, it is a low-cost technique, as opposed to the pin-level fault injection described in [51] or the heavy-ion approach applied in [52].

It must be highlighted that simulation-based fault injection is usually time consuming. The issue is that for a comprehensive analysis, it is required to simulate all possible scenarios, including all fault sites under all possible input vectors. Clearly, this combination can lead to an intractable number of scenarios. The number of scenarios increases even more if multiple faults are to be considered. Thus, multiple faults are usually not considered when performing fault injection by simulation means. For some complex circuits and systems, it is not feasible to evaluate all possible single faults either. Being so, partial evaluations are performed. Selecting which parts of the circuit should be evaluated and which ones can be left aside is also an issue. In order to deal with the time limitations of simulation-based fault injection, hardware-based techniques have been created. These techniques are explored in the next section and are referred as emulation-based.

2.2 Emulation-based Fault Injection

The guiding principles of emulation-based and simulation-based fault injection are exactly the same. Nevertheless, emulation-based solutions make use of a support platform such as an FPGA. Furthermore, most solutions make use of a low-cost commercial off-the-shelf FPGA. Such platforms offer a great deal of resources and have been successfully used to obtain faster results (with respect to simulation-based approaches). Unfortu-

nately, the use of such platforms also brings a considerable drawback: the observability is usually compromised, i.e., the user does not have direct access to all signals in the circuit being analyzed.

Several works already explored the use of FPGAs for speeding up fault simulation of permanent single stuck-at faults. For instance, the authors of [53] proposed a novel emulation technique that does not require circuit reconfiguration, neither complete nor partial. Fault injection is controlled by a scan chain scheme, which is constantly shifted to select the following fault to be analyzed.

In [54], the authors explored an FPGA platform for studying the effects of SEUs in ASICs. A modified version of the original circuit is instrumented with modified flip-flops. The proposed method then uses a host computer to control fault injection and later classifies the effect of those faults into silent (no sign of the fault can be found anywhere in the design), latent (the output of the circuit was not corrupted but a fault remains in its memory elements), failure (the output of the circuit was corrupted) or detected (a fault detection/ mitigation mechanism has captured the fault).

A mixed simulation/ emulation approach is presented in [55]. Complex digital systems are simulated together with SETs during the clock cycle where they first appear. Simulation is then able to tell how they propagate from the affected gate to the circuit's memory elements and if multiple errors could appear. Then emulation is used to observe how the error(s) resulting from the SET spreads in the circuit, possibly reaching the circuit's outputs.

The authors of [56, 57] propose a multilevel FPGA-based approach for evaluation of SETs. Such approach is considered multilevel since it integrates gate level and RTL models of the circuit under test and is able to switch to the appropriate model as needed. Fault injection itself is performed at the gate level, which provides delay accuracy, while fault propagation across clock cycles is performed at the RTL for higher performance.

In the last years the FPGAs have evolved such that partial reconfiguration is possible [58]. Some boards allow for dynamic reconfiguration, i.e., while parts of the design are operating other parts can be reconfigured. The authors of [59] replaced the use of specific external signals controlling additional logic (i.e., avoided instrumenting the original design) by relying on built-in reconfiguration capabilities of the FPGA devices. First, one fault-free run of the design is done. That first run is analyzed and the state of every flip-flop in the design is known at all cycles. Then, a second run of the design follows, in which reconfiguration is applied to modify the state of one flip-flop at a time, thus recreating the effect of an SEU.

Another solution is proposed in [60], where the authors describe a platform that is capable of evaluating multiple faults. Saboteurs are used to instrument the original circuit. A particularity of this solution is that the platform counts the number of erroneous scenarios that were masked. This value is later used as an input to a Probabilistic Binomial Reliability (PBR) model [40, 61], which in turn calculates the actual circuit reliability. More details concerning PBR are given in Section 2.4.5.

2.3 Physical Injection

Several techniques fall into the physical injection category. In general, these techniques make use of some form of accelerated fault source. Access to these sources can be costly and not trivial in some cases. A fabricated sample of the desired circuit is required for this

type of technique, which also has a cost associated with. Thus, this type of technique is prone for after fabrication screening, a process that is used to confirm that a part (usually an IC) conforms to a certain standard such as a maximum failure rate.

High dependability applications that have reliability requirements have used screening since the 1960's. For instance, the ICs that were embedded in NASA's Apollo Guidance and Navigation Computer were submitted to stress test procedures before being used [62]. At that time, variation between devices and between vendors was a big concern. Nowadays the screening process is much more complicated since the circuits are also much more complex. Some of the techniques used nowadays are explained in the next paragraphs.

Several authors have used test setups at high altitude for real-time experiments. The idea is that the flux of particles is denser at high altitudes which can make the characterization of a device faster and still reliable. Autran et al. [63] have done SER measurements of SRAM memory in the French Alps while the authors of [64] have used charge-coupled devices for measuring the event rate, also in the Alps region.

It is more or less consensus that high altitude testing can increase the event rate by tenfold. An event rate in the order of tenths per day could possibly be reached, which is still (very) low for high-fidelity estimation of circuit reliability. Artola et al. [65] have experimented with testing during balloon flight and commercial transatlantic flights as well. Once again, the number of events registered during the experiments is quite low to allow for the characterization of a complex digital circuit.

Another technique that has been used by researchers is the application of pulsed lasers to cause an effect similar to that of an SEE. This technique is very promising because pulsed laser is nondestructive and it can be focussed. In other words, whenever a fault is generated in the target circuit, its origin can be mapped and studied. This is not true for experiments that are based on ion beams, which are not focussed.

Unfortunately, as highlighted by [66], this technique has several downsides. The main issue is that the light cannot probe sensitive areas covered with metal. Thus, circuits have to have some open area on the sensitive drains through which the laser light could reach the sensitive junctions. Nevertheless, laser was successfully used in [66] to measure and compare error rates in combinational and sequential logic. The authors of [67, 68] demonstrated fine-scale laser SEE mapping of an SRAM memory and SET mapping of subregions of an amplifier. A backside polished sample of the SRAM was required, which again, highlights the drawback of the technique.

Experiments with ion beams are very common and there are several facilities that offer such services. Several types of beams can be used, depending on which ion(s) will be used. Angle of incidence is also an important parameter for such experiments. Thus, different Linear Energy Transfer (LET) values can be chosen and/ or targetted in a given experiment. When performing a heavy-ion experiment, a single source can be used [52] or a cocktail of ions can be assembled, as in [69]. Neutrons can also be used as an effective source of faults, as shown by [70]. Protons are also widely used, as shown by [69, 71]. Other less conventional sources of faults are muons [72], alpha particles [73] and pions [74].

There are several details related to all these physical injection techniques that involve low-level physics and are not the focus of this work.

An experiment using neutrons to inject faults in an FPGA is reported in Appendix B.

2.4 Analytical Approaches

Analytical approaches have been developed due to the inefficiency or inability of the traditional fault injection methods to handle large circuits. These approaches have their own limitations as well, but generally speaking they do not present the high costs of physical techniques nor the long execution times of simulation-based solutions. All of the methods described in this session can estimate the reliability of combinational logic only.

2.4.1 PTM

The first analytical method to be presented is referred as Probabilistic Transfer Matrices (PTM) [75, 76]. It is a simple method that models, through the use of matrices, the logic gates and the topology of a circuit. The main idea of the method is to define the correlation between the output patterns and the input patterns of a circuit in a 2mx2n matrix, where m is the number of inputs and n is the number of outputs of a circuit. In order to do so, each logic gate is also represented as a PTM matrix. This representation is achieved by the use of two auxiliary elements: the Ideal Transfer Matrix (ITM) and the parameter q, as illustrated in Fig. 2.1.

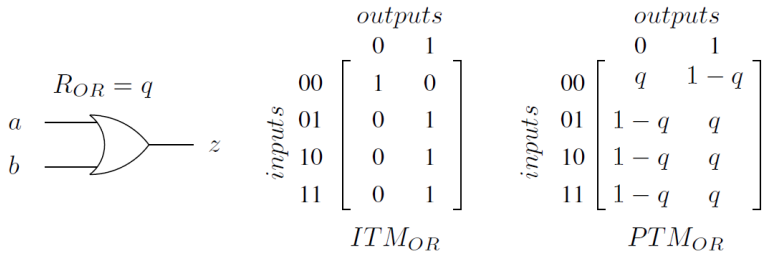


Figure 2.1: PTM's representation of an OR logic gate.

The ITM matrix represents the truth table of the logic gate. Such matrix is then modified to create the PTM: each '1' on the table is replaced by q and each '0' is replaced by (1 □ q). The q parameter represents the logic gate's ability to deliver a correct response at the output(s), i.e., it represents the singular reliability of a logic gate. Thus, (1 □ q) represents the probability of a fault. Concerning Fig. 2.1, the PTM matrix being shown uses a fault model in which it is considered that a fault causes a gate to flip its output value. Other fault models, such as the stuck-at model [9, 10, 77], can also be used.

Given each logic gate in a circuit is represented by a PTM matrix, it is then necessary to calculate the PTM matrix of the whole circuit by taking into account the topology. Such calculation is done by leveling the target circuit, as illustrated in Fig. 2.2.

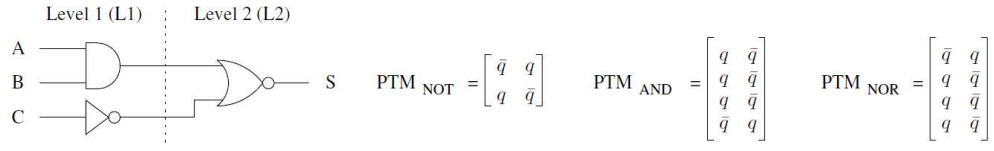


Figure 2.2: PTM's representation at circuit level [40].

The method continues by calculating the PTM of each circuit level. The PTM of the

first level (PTM_{L1}) is calculated by performing the Kronecker product of PTM_{AND} and PTM_{NOT} ($PTM_{L1} = PTM_{AND} \square PTM_{NOT}$). The PTM matrix of the second level is already known and is given by PTM_{NOR}. Finally, PTM_{L1} is multiplied by PTM_{L2} to obtain the PTM of the whole circuit.

Although the PTM method is able to estimate the reliability of a circuit accurately, it is not feasible even for medium-sized circuits. The complexity is exponential with both m and n.

2.4.2 SPR

PTM is the basis for all of the methods in the Signal Probability Reliability (SPR) family. When applying the PTM method, the size of the intermediate matrices increases at a fast pace. The SPR method [40, 78, 79] tries to avoid this issue by representing each signal in the circuit by a 2x2 matrix. Such matrix is illustrated in Fig. 2.3.

$$P_{2\times2}(signal) = \begin{bmatrix} P(signal = correct\ 0) & P(signal = incorrect\ 1) \\ P(signal = incorrect\ 0) & P(signal = correct\ 1) \end{bmatrix}$$

Figure 2.3: SPR's matrix representation.

In the SPR matrix representation, it is assumed that a signal may carry four distinct values. These values are: a correct '0', an incorrect '0', a correct '1' and an incorrect '1'. The SPR matrix then contains the probability of a signal being one of these mentioned values.

Let us consider an OR gate, as represented in Fig. 2.4. Let us also assume that its inputs are already represented as SPR matrices A4 and B4. In order to calculate the SPR matrix of the outputs, it is necessary to take into account the inputs, the logic function and also the reliability of the gate. The logic function and the reliability are already given by the PTM matrix representation. And, since a gate might have multiple inputs, the joint probability of the inputs must be considered. This is achieved by calculating the Kronecker product of the inputs, as illustrated in Fig. 2.4. The resulting matrix is multiplied by the PTM of the gate.

One last step is then performed, in which the values from the P(S) matrix are merged into a single SPR matrix. Values are merged according to the ITM matrix of the gate. This step inserts a certain loss of accuracy due to erroneous evaluation of reconvergent fanouts. This issue is explained in details in Section 2.4.4 and it is the motivation for the SPR Multi Pass (SPR-MP) method. Nevertheless, this same step allows for a improved performance (when compared with the PTM method). The complexity of the SPR algorithm is linear with the number of gates.

2.4.3 SPR-DWAA

A modified version of SPR, referred as SPR Dynamic Weighted Averaging Algorithm (SPR-DWAA), was proposed in [40]. It uses a variation of the Weighted Averaging Heuristic (WAA) [80] to correct the signal probability values. WAA's was originally intended for determining the influence of each primary input on signal probability values. This is

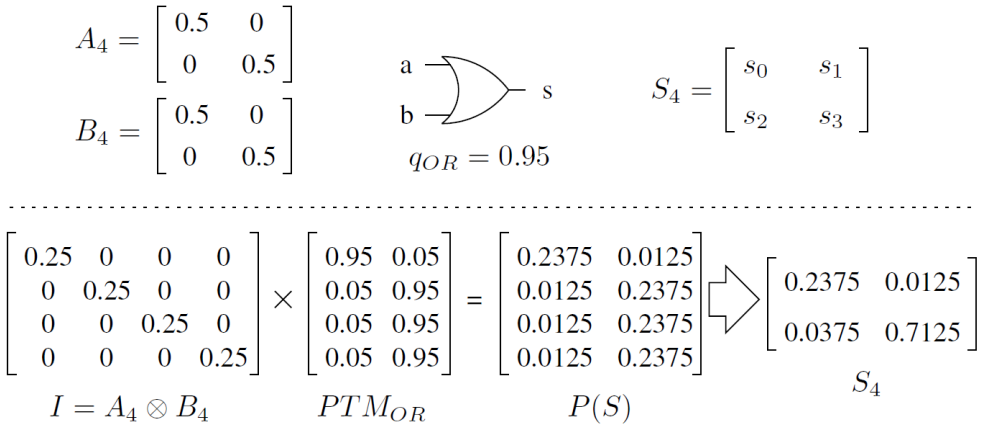


Figure 2.4: Example of signal probability propagation in an OR gate [79].

accomplished by iteratively setting each primary input probability to 0 and 1 and evaluating the results obtained. Yet, its modified version is used to calculate the influence of each fanout on the circuit reliability.

In order to apply the DWAA algorithm, the fanouts of the circuit must be ordered topologically from input to output. The algorithm processes one fanout signal at a time, successively setting its probability to 0 and 1, and dynamically updating the signal probabilities on its fanout cone. This way, the algorithm updates signal probabilities according to a dependency order.

SPR-DWAA can achieve a more precise result than SPR, but the precision depends on the order in which the fanouts are evaluated. Precision is not necessarily 100% accurate, i.e., deviations from the actual circuit reliability still occur when using SPR-DWAA. In the same way as SPR, SPR-DWAA still has a linear complexity. SPR-DWAA's complexity is bounded by O(F □ G), where F is the number of fanouts in the circuit and G is the number of gates in the circuit. Results concerning SPR-DWAA's accuracy are given in [40].

2.4.4 SPR-MP

The motivation behind the SPR-MP method comes from the issue of reconvergent fanouts. Let us assume a simple circuit C to illustrate this issue, as shown in Fig. 2.5. Without loss of generality, the input SPR matrix contains two zero values which allow for a simpler analysis (thus, only two elements are considered, a_0 and a_3). Although a reconvergent fanout is not part of the topology itself, in order to obtain the final reliability R(circuit) of the example, it is required to multiply the reliability associated with each of the circuit outputs. This operation has the same effect as an actual reconvergent fanout in the circuit topology.

First, let us start by assuming that the SPR method was applied to the given circuit, as shown in Fig. 2.5. Since SPR cannot handle reconvergent fanouts properly, it produces an approximation of the accurate result. The equation shown for the circuit reliability contains two terms that should not be accounted for. Terms like $a_3^2q^2$ are inconsistent since they depend twice on the same probability (a_3). Along the same lines, terms like $a_0a_3q^2$ are inconsistent since they depend on different states of the same signal.

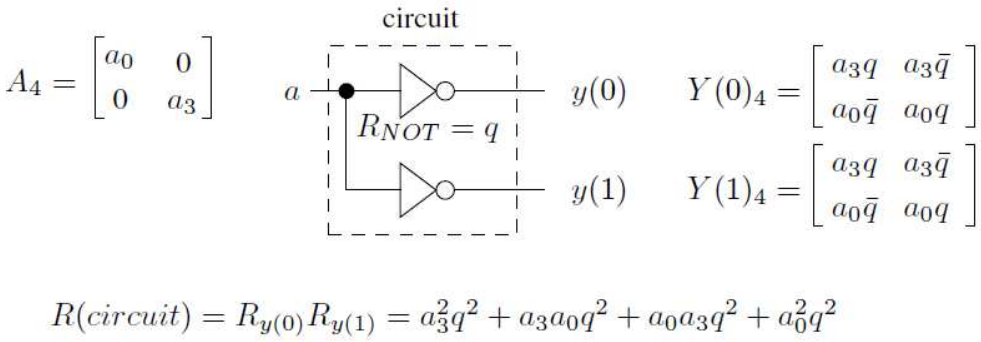


Figure 2.5: Computing the reliability of a simple circuit with a reconvergent fanout [40].

The SPR-MP method solves this issue of inconsistency by splitting the analysis in multiple passes (hence the name). In each pass a single state of a signal is considered while all the others are assumed to be zero. Figure 2.6 illustrates this concept (check matrices A_4). Since it was assumed that both $a_1 = a_2 = 0$, only two passes are being shown in Fig. 2.6. In a real scenario, all four possible states should be evaluated. Thus, there are four partial reliability values for each fanout node. The reliability of the circuit as a whole is given by the sum of all partial reliability values. For the example illustrated in Fig. 2.6 the reliability is given by:

$$R(circuit) = R(circuit; a_0) + R(circuit; a_3) = a_0 q^2 + a_3 q^2 \tag{2.1}$$

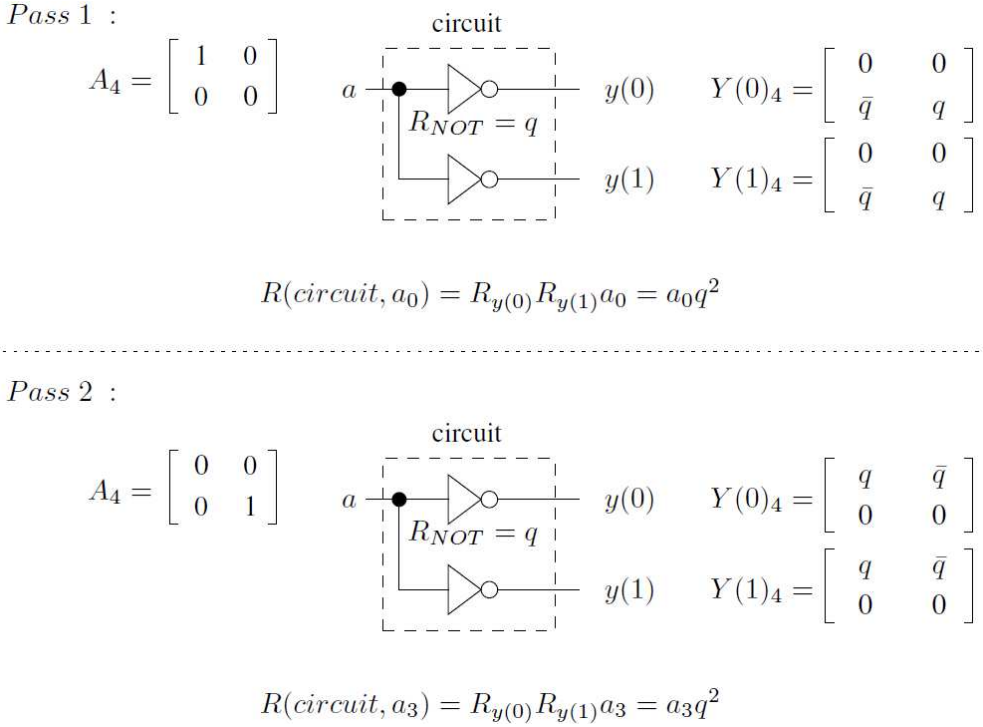


Figure 2.6: SPR-MP algorithm applied to a simple reconvergent circuit [40].

As expected, the reliability given in (2.1) no longer depends on inconsistent terms. The accuracy of SPR-MP is 100% precise if all fanouts are evaluated like previously described. The penalty comes in terms of execution time, which no longer presents a linear complexity. The complexity of the algorithm is exponential with the number of fanouts.

A straightforward trade-off between execution time and accuracy is possible when using SPR-MP. Through the concept of dominant fanouts, i.e., the idea that some fanouts are more important than others, it is possible to diminish the execution time drastically. Depending on the topology of the target circuit, even when only a small number of fanouts is considered, a large reduction in execution time is possible with an error that is smaller than 2% [78].

It must be mentioned that is up to the designer to choose which method seems to be a better fit. PTM is on the extreme edge of accuracy: it will always produce accurate results. SPR is on the other edge: it will produce a relatively inaccurate result but in an affordable linear time. On the other hand, there is SPR-MP which lies somewhere in between: accuracy and complexity can be traded-off by considering only a portion of the total number of reconvergent fanouts.

2.4.5 PBR

The PBR method itself is an analytical solution, thus it is part of this section. Nevertheless, the inputs to the method are usually obtained through simulation or emulation means. Details concerning how to obtain those inputs are described in [60] while the method itself is described in the next paragraphs.

According to the PBR model, the reliability of a digital circuit is given by (2.2), where:

- N is the number of gates that may fail.
- q represents the reliability of a gate, that is, the probability that it does not fail. All gates of the circuit are considered as having the same reliability value.
- k is the number of simultaneous faults.
- x_j is a vector containing all the inputs of the circuit.
- e is an error vector. The width of such vector is equal to the number of simultaneous faults being considered. Thus, there are C_k^N possible vector values for k simultaneous faults.
- y is the output of the circuit when an error vector e and an input vector x_j are applied.
- y_{ref} is the fault-free output of the circuit, used as reference.
- f (k) denotes the probability that k gates fail simultaneously (more complex models can also be used to evaluate this term), as shown in (2.3)
- c_k denotes a coefficient related to the masking of k simultaneous errors in a circuit. Considering that the target circuit has Z input bits, it can be calculated using (2.4)[1].

[1] The $\overline{\oplus}$ sign with the bar on top of it represents the XNOR operation

$$R = \sum_{k=0}^{N} f(k) c_k \tag{2.2}$$

$$f(k) = (1 - q)^k q^{N-k} \tag{2.3}$$

$$c_k = \sum_{j=0}^{2^I - 1} p(x_j) \left(\sum_{l=1}^{C_k^N} \frac{1}{y(x_j; e) - y_{ref}(x_j)} \right) \tag{2.4}$$

2.4.6 Other techniques

The work of Han et al. [81] uses a Probabilistic Gate Model (PGM) to obtain circuit reliability values. This modelling can be used by accurate or approximate reliability calculating algorithms in the same way that SPR and SPR-MP use the same underlying modelling. PGMs can be used to model inversion and stuck-at faults. The operations of binary functions can be mapped into a set of probability values in the real interval [0, 1], thus allowing gates of any complexity to be modelled as PGMs. The procedure in which the signal probability is propagated from inputs to outputs is much similar to SPR's approach.

Flaquer et al. [82] proposed conditional probabilities which are interesting in handling reconverging signals as they can decorrelate those signals and enable a rapid treatment using the SPR approach. Accuracy is obtained and the execution times are smaller than those obtained when using SPR-MP. The obtained speed-up depends on the circuit itself and how reconverging signals can be organized into clusters.

Other works explore different ways to represent probabilistic behavior of a circuit given the presence of faults. One of such representations is termed Probabilistic Decision Diagram (PDD) and it was proposed by Abdollahi in [83]. PDDs can be used to obtain exact reliability figures. The PDD approach consists in transforming the circuit into a graph in which each gate is a PDD node. Each node implements the original gate function as if a probabilistic inverter was connected to its output. Such inverter is used to model the fault probability of that gate.

2.5 Fault Tolerance Techniques

This section discusses a few techniques that are used to increase the reliability of digital circuits. Some techniques are able to detect errors, some others are able to detect and correct while a third group focuses on avoiding errors by enhancing the reliability of the underlying circuit.

A whole different set of techniques is used for memories, but these are not covered by this thesis. Generally speaking, these techniques are some form of Error Correcting Code (ECC).

2.5.1 Modular Redundancy

Modular redundancy is a family of techniques that is based on spacial redundancy. Proposed by Von Neumann [84], Triple Modular Redundancy (TMR) is the most notorious

technique of the family. It consists in placing three copies of the same module operating in parallel. The output of the modules is voted such that if a (single) fault occurs in any of the modules, it will be masked.

Several other works have made use of TMR in different ways. One of the main research efforts related to TMR is how to optimize the placement of the voter. The simplest approach is to place a single voter per output, a solution that is also referred as global TMR. Nevertheless, other solutions are also possible by placing multiple voters in different areas of the circuit. Different granularities lead to different reliabilities with also different implementation costs.

One of the reasons for which TMR is widely used is because majority is easily calculated in the case of 1-out-of-3 faulty modules. Two modules are good enough for detecting an error, but some form of recomputation is needed since it is impossible to tell which module is erroneous. Higher order schemes, termed N-Modular Redundancy (NMR), present issues to determine the majority and usually require larger and/ or more complicated voting schemes.

Examples of TMR use can be seen in [85–90]. The technique proposed in Section 4.2.3 also uses TMR by combining local and global TMR schemes for improving the circuit reliability.

2.5.2 Selective Hardening

Some applications do not have enough of a hardening budget that allows for a full TMR-like replication of the system. Thus, some form of selective hardening takes place when just a few areas of the circuit are hardened while others are eventually left untouched. The concern then is how to choose the location and the amount of such areas, for which several authors have proposed different solutions.

Many recent works concerning selective hardening (also referred as partial hardening) are simulation/ emulation based. For example, in [91], the authors identify the critical parts of a microprocessor with respect to SETs.

Circuit hardening by applying TMR was studied in [85], similarly to what is later proposed in Section 4.1 of this thesis. The difference is that in [85] the trade-off between area and reliability is not directly evaluated. Multiple architectures are generated and only the ones under a certain area limit L are evaluated.

In [92], the authors have studied the problem of selective hardening by considering only single faults. Reliability figures were obtained by applying the same simulation/ emulation model that is described in Section 2.4.5. The main drawback of the PBR methodology is the need for fault simulation or fault emulation, tasks that are inherently time consuming. Two metrics for ranking gates are proposed in [92], termed sensitivity and eligibility. Both metrics try to measure how each gate (or each enhancement of a gate) contribute to the overall circuit reliability.

In [93] the authors have described a cost-limit approach for selective hardening. They have chosen an algorithm with linear complexity and accept the fact that inaccurate values will be used for estimating the relative reliability gains of hardening a given cell in a circuit. Yet, they are only concerned with single faults. Also, they propose a cost target such that the number of hardened cells does not exceed a cost limit L.

Gate sizing is a particular technique that can also be seen as a form of selective hardening. For instance, in [94] the authors use simulations to find which gates are more likely to propagate zeros or ones. With that information it is possible to increase the size of the

transistors in specific gates. By doing so, the critical charge is increased and transients can be mitigated.

2.5.3 Other Techniques

Works such as [95] have made use of modified synthesis algorithms to achieve a more resilient circuit. The main idea is that common target properties such as area, timing and power can be traded-off with reliability. Therefore, the synthesis process will pick gates (from a library) that are more reliable but do not necessarily present the same efficiency (with respect to other properties).

Some techniques combine spatial and temporal redundancy, such as DWC-CED [96]. Two copies of the main module are used together with a comparator to detect faults. Once a fault is detected, a temporal scheme is applied to detect which module is the faulty one. The main advantage of the method is that it has a lower cost than standard TMR.

Several techniques promote circuit changes at the layout level and are referred as Hardening by Design (HBD) techniques. Such techniques are classified according to which circuit element they modify, i.e., these could be hardened memory cells, hardened flip-flops, hardened clock gating cells, etc. For instance, two similar approaches have been proposed in [97, 98], termed Heavy Ion Tolerant and Dual Interlock Cell (DICE). Both approaches duplicate the state-holding nodes in order to avoid upsets in memorizing elements. A considerable amount of different solutions concerning flip-flops is available, allowing the designer to decide on which trade-offs to pick. Several of those solutions are updates of the original DICE.

Recent works have also concerned with the reliability of the clock network. Since this is a very important part of the design, any upset hitting it could be severe. The work of Ghahroodi et al. [99] proposes duplicated clock-gating latches connected in series with an AND3 gate (the inputs being both the latched signals plus the actual clock). Other works propose the use of standard TMR in the clock network, which can have a high impact on the circuit power budget.

Chapter 3

Reliability Analysis Methods

This chapter covers two methods that were developed for the analysis of a circuit's reliability. As previously stated, there is a lack of methods capable of coping with the many difficulties imposed by reliability analysis. Thus, the first method here proposed tries to tackle the accuracy issue by performing an analysis of fanout reconvergent nodes. This method is termed SPR+.

The second method proposed in this thesis is termed SNaP and it applies a completely different approach to reliability analysis. First and foremost, it is a hybrid method, combining the benefits of simulation and analytical solutions. The obtained accuracy of the method is discussed in depth and compared against SPR and SPR-MP.

3.1 SPR+: Heuristics for Reliability Assessment of Combinational Logic Using First-Order-Only Reconvergence Analysis

Considering the recent trend of increase in the number of defects and soft errors, a great deal of concern is given to properly estimating circuit reliability. Without accurate or near accurate estimation, circuit hardening techniques can be applied in an uneffective way. Some specific applications, like medical and avionics, require high reliability. Nevertheless, algorithms capable of assessing circuit reliability have severe limitations. Those limitations are discussed in Chapter 2.

On top of those limitations, a recent mechanism has made circuit reliability assessment even more complicated. Such mechanism, known as charge sharing, has increased the amount of multiple faults due to strikes of particles. Thus, algorithms capable of handling multiple faults are of great interest. Although traditional simulation can be used for multiple fault modelling, it can easily become an intractable problem if all combinations of fault sites have to be taken into account. Analytical methods are critical for such type of analysis.

This section proposes an adapted version of the SPR-MP method [79], in which reconvergent fanouts are partially taken into account. This adapted version is simply termed SPR+ and it was developed using two heuristics for handling the reconvergent fanouts. The next section highlights the contribution of the method and its heuristics.

3.1.1 Contribution of the SPR+ Method

The contribution of this adapted method is to propose two simple heuristics for estimating exact circuit reliability. Such heuristics take into account first-order-only fanout reconvergence, thus they can be used for the evaluation of large circuits where simulation and other analytical algorithms cannot or would take a long time to do it.

Fanout reconvergence is a problem that has been addressed by different authors in the literature. The authors of [100] have proposed exponent suppression while the authors of [80, 101] have proposed heuristics for correcting signal correlations. None of these solutions can calculate accurate circuit reliability.

Such problem is resolved by the use of the SPR-MP algorithm, as shown in Section 2.4.4 and also in Fig. 2.5. Each fanout reconvergent node is handled separately, in multiple passes, thus the name SPR-MP. The problem of such solution is that the algorithm complexity becomes bounded by $O(4^f)$, where f is the number of fanouts in the circuit. Nevertheless, the algorithm is accurate. The solutions presented in this section do not have the same complexity, i.e., they present a trade-off between accuracy and execution time.

3.1.2 Metric for Comparison

A full SPR-MP analysis that takes into account all of the circuit fanout nodes is not possible, even for small-sized circuits. The execution times are very long and/ or completely unfeasible. Nevertheless, in order to assess how much accurate the proposed heuristics are, a feasible reference target for comparison is required.

Thus, circuits from the ISCAS'85 set of benchmarks were analyzed using the SPR-MP method but limited by a 12th order analysis (i.e., the 12 most relevant fanout reconvergent nodes from each circuit were taken into account, all at the same time). This analysis is termed R_{12th}. Since the SPR-MP method can be used to do partial fanout evaluation, it was the chosen method for giving a reliability reference for each circuit. The reliability values as well as the execution times are given in Tab. 3.1. Execution times shown include the time required to find out the 12 most relevant fanouts.

Table 3.1: Reliability analysis using R_{12th}.

Circuit	Gates	Fanouts	Reliability	Execution time (s)
c17	6	3	0.9999437519	0.05
c432	160	89	0.9986423278	486.93
c499	202	59	0.9986269089	30.32
c1355	546	259	0.9977799443	3663.79
c1908	880	385	0.9967790239	130.4
c2670	1269	454	0.9933852285	1142.42
c3540	1669	579	0.9934856289	80.17
c5315	2307	806	0.9910769681	2015.51

The choice of a 12th order (instead of a lower or higher order) is motivated by two reasons: feasible execution times and relative accuracy. The execution times are given in Tab. 3.1, and the longest execution time (concerning the circuit c1355) is of approximately one hour.

When it comes to accuracy, it is important to highlight that not all fanouts contribute equally to a higher accuracy. The illustration in Fig. 3.1 shows how the circuit reliability converges to a certain value by successively moving to a higher order analysis, adding one fanout at a time. Thus, the first analyzed fanouts are much more important than the others. It is also possible to see how the execution times rapidly increase.

The reliability estimations plotted in Fig. 3.1 were obtained from the analysis of the c499 circuit. SPR-MP modelling was used and each gate was set with a reliability value of $q = 0{:}999$.

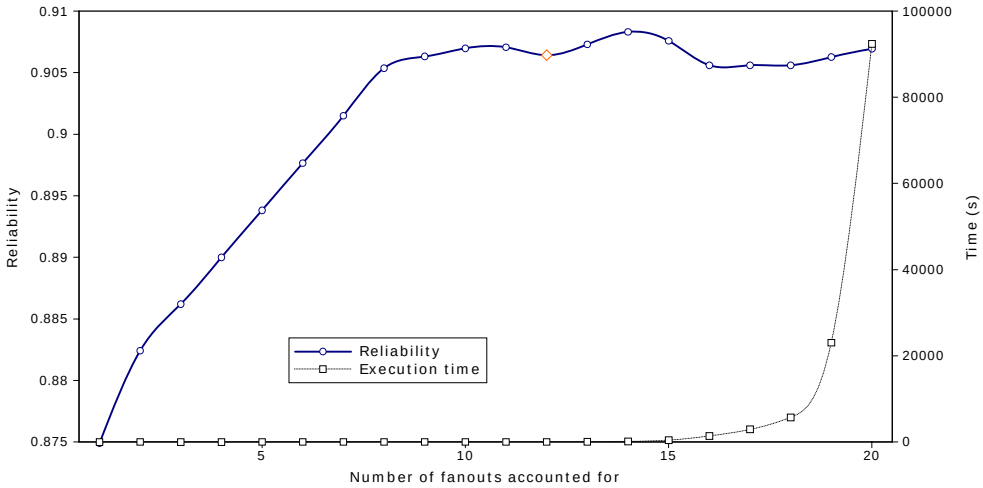


Figure 3.1: Analysis of different number of fanouts, circuit c499.

A different pattern is seen in Fig. 3.2. This analysis was obtained using the c3540 circuit and a reliability value $q = 0{:}9999$. Choosing a higher q value does not change the general profile shown in the images, since it is a property of the circuit and its fanout nodes. A different q only changes the order of magnitude of the reliability values. Nevertheless, the measured reliability of the whole circuit still converges to a value. Clearly, as seen in both images, the measured reliability for the 12th order analysis (marked with an orange diamond shape) is closer to the actual circuit reliability.

In order to chose which are the 12 most significant fanouts, the following approach was used: let C be a target circuit with F reconvergent fanouts. Also let R_0 be the circuit reliability when zero fanouts are considered (i.e., the same reliability obtained when using SPR). Let $R_1(f)$ be the circuit reliability when only one fanout f is considered. By varying f, a total of F circuit reliability values is obtained, from which the following comparison can be made:

$$D(f) = |R_1(f) - R_0| \tag{3.1}$$

The 12 values with the biggest difference $D(f)$ are considered the most significant ones. In other words, the fanouts that are the most neglected by the SPR analysis are considered more critical.

It is clear that using a 12th order estimation does not lead to an accurate reliability value. Nevertheless, it is clear that not all reconvergent fanouts have the same impact on the overall reliability (i.e., not all $D(f)$ values are equal or even from the same order of magnitude). The result shown in Fig. 3.3 is an attempt to classify each f fanout by its

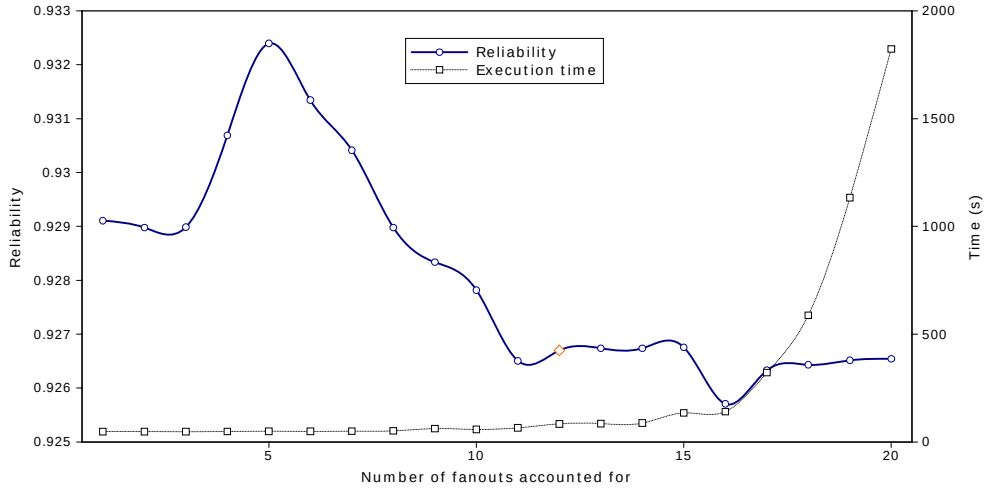


Figure 3.2: Analysis of different number of fanouts, circuit c3540.

$D(f)$ value. First, the largest difference was identified and it was termed D_{max}. Then, the impact of each fanout is classified as follows:

- High impact, if $D(f)/D_{max} > 0.8$
- Medium impact, if $D(f)/D_{max} > 0.2$
- Low impact, for all the others ($D(f)/D_{max} <= 0.2$)

It is clear from Fig. 3.3 that the number of high impact fanouts is quite low. High impact fanouts are less than 3% of the total amount of fanouts. The margin given for medium impact fanouts is quite broad and, nevertheless, they still account for less than 10% of the average number of fanouts. Thus, the absolute majority of fanouts is not that important when assessing circuit reliability. This can be exploited in order to find circuit reliability figures that are more precise and in a short amount of time. Circuit c17 is not represented in Fig. 3.3 because it has only 3 fanouts.

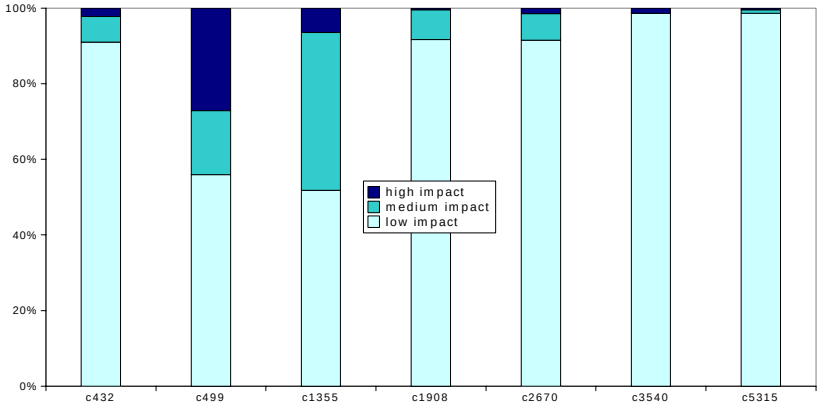


Figure 3.3: Impact profile of the fanout nodes based on $D(f)$ values.

3.1.3 Proposed Heuristics

Taking into account the fanout impact profile revealed by Fig. 3.3, two different heuristics were proposed. Both are detailed in the next subsections. Nevertheless, both have the same goal: to approximate the actual reliability value R by taking into account only first order elements. In other words, only $R_1(f)$ values will be used in an attempt to predict R.

3.1.3.1 Simple Ranking

This heuristic identifies the fanout f that is associated with D_{max}. This fanout is termed f_{max} and it is the only fanout to be taken into account. Circuit reliability is then estimated as:

$$R = R_1(f_{max}) \tag{3.2}$$

The goal of this heuristic is to answer the following question: is taking only one fanout into account better than taking no fanout into account at all? If so, then the goal becomes to evaluate how far from R this solution still is.

3.1.3.2 Negative/Positive Contributions

This heuristic identifies the average behavior of the fanouts in the circuit. As seen in Fig. 3.1, the majority of fanouts cause the circuit reliability to increase while others cause the opposite effect (a situation that appears when 12, 15 and 16 fanouts are considered). These were termed as positive fanouts and negative fanouts. Then, the circuit reliability is calculated as a ratio of both:

$$R = R_0 + wp \sum_{i=0}^{Fp} D(i) - wn \sum_{i=0}^{Fn} D(i) \tag{3.3}$$

Fp is the number of positive fanouts, Fn is the number of negative fanouts, while wp and wn are weights for each. Initially, both weights were set as 1, which does not lead to good estimations. Our assumption then is that a circuit with more positive fanouts will have a higher reliability value and vice-versa, but both types of fanouts do not contribute equally. Thus, it is necessary to identify the ratios between these fanouts, which were then used as weights:

$$wp = Fp/(Fp + Fn) \tag{3.4}$$

$$wn = Fn/(Fp + Fn) \tag{3.5}$$

The results shown in the next section were obtained using this ratio concept.

3.1.4 Results

Both heuristics described in the last section, as well as the simple SPR algorithm, were used to calculate the reliability of the case studied circuits. Each cell of each circuit was set with $q = 0{:}99999$ for all experiments. The results are shown in Fig. 3.4, in which it

is possible to identify that the proposed heuristic entitled 'Negative/ Positive Contributions' (white bar) is able to estimate circuit reliability with a good degree of accuracy for nearly all circuits.

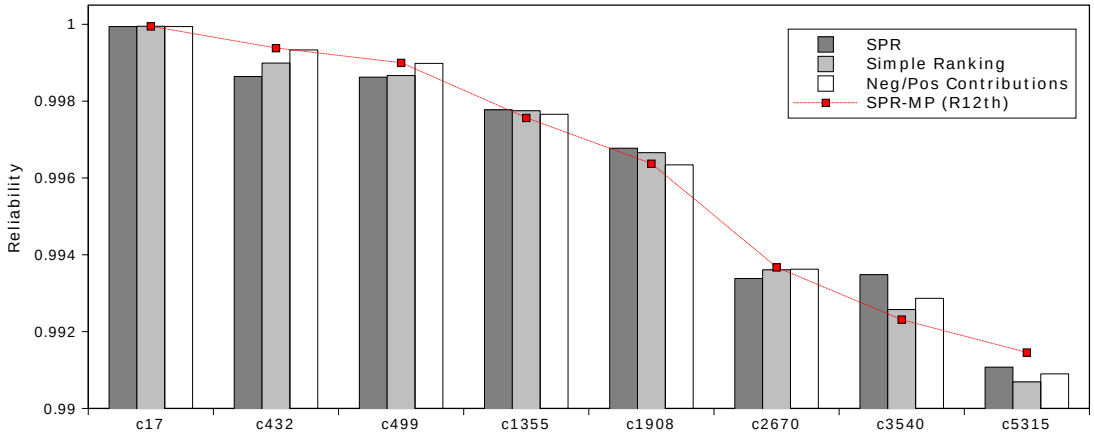


Figure 3.4: Comparison between both SPR+ heuristics and SPR.

Another interesting result obtained from Fig. 3.4 is that none of the SPR+ heuristics (or SPR itself) can be used as a method for worst case estimation. For instance, for the c3540 circuit, all three results are higher than the one given by R_{12th}. The same is also true for other circuits but not for the three methods at the same time.

In order to compare the results between SPR+ heuristics and between SPR itself, the chosen approach was to use a figure of merit for the accuracy error. Since each cell in each circuit was set with $q = 0{:}99999$, the value $e_{min} = 0{:}00001$ was taken as the smallest error value considered. We have then calculated the differences in reliability from each approach with respect to the 'nearly accurate' reliability obtained using R_{12th}. The values are shown in Tab. 3.2 as multiples of e_{min}.

Table 3.2: Figures of merit for the estimation errors.

Circuit	SPR	Simple Ranking	Averaged Neg/Pos
c17	0.68	0.01	0.34
c432	74.22	38.9	5.11
c499	37.44	32.9	1.46
c1355	21.09	18.6	9.32
c1908	40.27	28.6	3.11
c2670	29.24	6.84	5.21
c3540	117.00	26.14	55.21
c5315	38.44	76.88	55.59
Sum	358.42	229.12	135.37
Average	44.80	28.64	16.92

Average values in Tab. 3.2 clearly indicate that the 'Negative/ Positive Contributions'

heuristic introduces less error in the estimation of R. The same is also true for the summed values. The 'Simple ranking' heuristic introduces, in average, 37% less error than the simple SPR estimation. Similarly, the 'Negative/ Positive Contributions' heuristic introduces 63% less error than the simple SPR estimation.

As highlighted in Fig. 3.1, a high order estimation of the reliability can easily take more than a day to be complete. None of the heuristics presented require more than one minute to achieve a final result, even for the biggest of the circuits.

Both presented SPR+ heuristics can estimate circuit reliability with less error than the SPR algorithm. The 'Negative/ Positive Contributions' heuristic introduces less average error with very affordable execution times and thus can be used for estimating circuit reliability. As future work, this study can be extended to different circuits and search for alternative weighting schemes for (3.3). It can also be verified if second-order-only analysis can be used in the same way for estimating circuit reliability. If so, if there is any gain in accuracy given the increase in execution time.

3.2 SNaP: a Hybrid Method for Reliability Assessment

As shown in Section 2.4, in order to overcome long simulation times, several authors have proposed statistical methods to calculate reliability [40, 61, 75, 76, 78–80]. Nevertheless, most methods can only deal with combinational circuits and usually small-sized ones. Also, some methods do not scale well, being totally unable to estimate the reliability of average-sized circuits.

In the light of those limitations, a novel hybrid method was developed. Such method is termed SNaP and it is considered a hybrid solution since portions of the method rely on simulation, while others do not. Nevertheless, no fault injection takes place. In traditional simulation-based fault injection, one gate after another is selected as faulty and a simulation run is then performed for each gate. The output of that chosen gate is inverted during that run and the primarity outputs of the circuit are observed. SNaP avoids such laborious effort by assuming that all gates have a certain failure rate and by observing the failure rate obtained at the primary outputs. Thus, all gates can be evaluated at a single run. No internal signals are inverted while using SNaP's approach.

SNaP can also benefit from emulation, when used as a platform in an FPGA device. Emulation allows for faster evaluation of complex circuits. Thus, one possible implementation of the method will be shown, which is a fully synthesizable verilog implementation. Such implementation can be simulated (which allows for easy debugging) or emulated in an FPGA (which allows for rapid evaluation).

The main contribution of the method is to present itself as a new reliability estimation technique, from which the following characteristics should be highlighted:

- It is able to handle both combinational and sequential logic
- It is capable of dealing with single and multiple faults
- It is prone for handling complex circuits
- Its calculation time for the combinational logic is of linear complexity with the number of gates in the longest path
- Its calculation time for the sequential logic is constant

- Its suggested implementation scales linearly with the total amount of gates in the circuit
- Its suggested implementation can be emulated

The method described in this section was patented [102] and later was used as the basis for publications in conferences and journals [103, 104].

3.2.1 Basics on SNaP: a Hybrid Method for Reliability Assessment

The core concepts behind SNaP's modelling are fault sourcing and fault propagation. SNaP is based on these two opposing concepts, i.e., gates are able to generate faults and are also able to suppress faults. It is the interplay of both that determines how reliable the entire circuit is. Considering a whole circuit with multiple gates, the goal is to assess how many of those gates are actually capable of generating a fault that propagates (successfully) to any of the primary outputs. Logical masking is considered during that assessment. Certainly, logical masking must be considered since it is an important masking phenomena in combinational logic.

SNaP obtains circuit reliability for a gate level description of a circuit, i.e., the input of SNaP's analysis is a synthesized verilog netlist. Such netlist can be generic (library-free using verilog primitives) or already mapped to a library. This initial circuit description is modified and instrumented with additional signals and logic that will handle the modelling of faults, i.e., a series of transformations is performed to allow for the propagation and accountability of faults. Combinational and sequential logic are handled differently, as to be shown in the text that follows. The modelling of combinational logic is shown in details in Section 3.2.1.1. The same applies for sequential logic in Section 3.2.1.2.

As previously mentioned, gates are able to generate faults. SNaP models each gate as a fault source and different gates may generate faults at different rates. That being said, we define a parameter gf that determines the fault generation rate of gates (gf stands for gate faults). Inverters are considered as the reference rate and are used for determining the gf parameter of other gates. That is to say that it is assumed that $gf_{inv} = 1$ and other gates take gf values that are multiples of gf_{inv}.

3.2.1.1 Modelling Combinational Logic

Initially, let us consider a small circuit that only contains a simple inverter. Figure 3.5 (a) contains a representation of its functional behavior. The transformed circuit is given in Fig. 3.5 (b), which contains an additional analysis block and additional I/O signals. Inverters have no logical masking ability, i.e., a faulty input will never be filtered by the gate. Although no masking takes place, the inverter still is a possible fault source and that 'ability' must be taken into account. Thus, in SNaP, each transformed gate stores a value gf that expresses how likely faults are generated at that particular gate.

Still referring to Fig. 3.5 (b), the modified description has two additional I/Os. The first is a multi-bit input, termed ifs, which contains information on how many previous faults sites can reach that circuit node. The second I/O is a multi-bit output, termed ofs, which contains the same information for the node after the inverter. The block termed 'analysis' (drawn in blue) calculates ofs by taking into account ifs and gf. Each type of gate can have a different analysis block. Since the inverter does not mask faults, there is

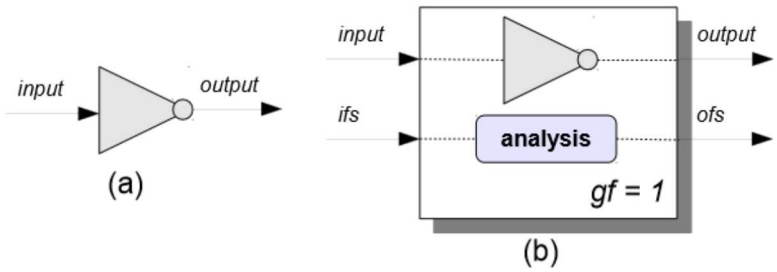


Figure 3.5: (a) Functional representation of an inverter; (b) SNaP representation of an inverter.

no need to take functional inputs into account. Therefore, ofs can be easily determined as:

$$ofs = ifs + gf_{inv} \tag{3.6}$$

Nevertheless, it is mandatory to take logical masking into account when assessing circuit reliability. For that reason, let us consider another circuit, illustrated in Fig. 3.6 (a) and its modified version in Fig. 3.6 (b). The goal is to calculate the value of the ofs output of the AND gate. For simplicity's sake, let us assume that all four primary inputs of the circuit (inputA, inputB, ifsA and ifsB) are zero. Thus, there are no external fault sites reaching the inverters.

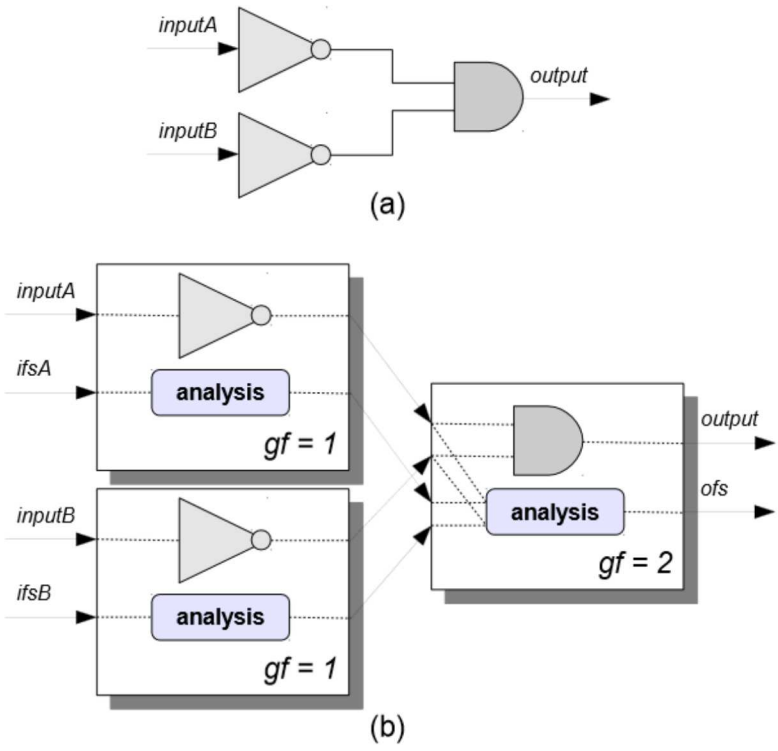


Figure 3.6: A simple circuit and its functional and SNaP's modified representations.

Some of the signals in Fig. 3.6 were removed to keep the image clearer, but there

are control signals related to the I/ Os of each gate in the circuit. These are called ready signals. A given gate only begins its ofs computation when all of its input ready signals are asserted. Likewise, it only asserts its own ready signal when it has finished calculating.

Also, parallelism comes naturally with the developed method. In the example circuit depicted in Fig. 3.6 (b), both inverters would receive a ready signal at the same time and would begin the analysis at the same time. Thus, SNaP's calculation time for the combinational logic depends linearly of the number of gates in the longest path. The number of clock cycles needed to reach a final result is not dependent on the total number of gates.

In the example given in Fig. 3.6 (b), the inverters will behave exactly as previously described. And once they are finished calculating, they will assert their own ready signals. Once the AND gate detects that the ready signals related to both of its inputs are asserted, it begins calculating. In the text that follows an implementation for the analysis block of the AND gate is suggested. Certainly it is not the only one possible, but it is an attractive solution since it is guaranteedly synthesizable.

That being said, the analysis block of the AND gate is implemented as a Finite State Machine (FSM) with 5 states: waiting, errorOnInputA, errorOnInputB, errorOnBoth and finished. Such FSM is fully synthesizable and can be generalized for gates with a higher number of inputs. The waiting and finished states remain exactly the same no matter the number of inputs, while the others increase in number. There will be a state for each possible combination of single and multiple faults. Thus, for a gate with three inputs, 7 possibilities will be considered: errorOnInputA, errorOnInputB, errorOnInputC, errorOnInputsAandB, errorOnInputsAandC, errorOnInputsBandC and finally errorOnABC.

In order to comprehend how the method works, let us proceed with a description of each state of the FSM. The waiting state is very simple: at each clock cycle, all the ready input signals are checked and once all are asserted, the state changes to errorOnInputA. As previously mentioned, the gate itself contributes to the ofs value. Thus, at the waiting state, ofs is set as shown in (3.7). The states that follow are the ones actually in charge of handling logical masking.

$$ofs = gf_{and} \tag{3.7}$$

The next state is errorOnInputA, which verifies if a toggle in inputA could be able to affect the output of the gate. If it does not, nothing is done and the analysis continues. On the other hand, if the output could toggle, then ofs must be updated. This is the moment where logical masking is considered in SNaP's analysis.

Naturally, ofs must be updated using some form of balancing or ratio that involves the amount of fault sites related to the inputA signal, referred here as ifsAndA. This balance was empirically determined as shown in (3.8), where the >> symbol represents the shift right operation. Such operation was chosen due to its easiness of implementation in a digital design.

$$ofs = ofs + (ifsAndA >> 1) \tag{3.8}$$

Notice that the ofs is updated with a new value in an accumulator-like fashion. The gf_{and} value that was previously stored in the register is not lost.

The following state is errorOnInputB, which is analogous to the previous one. The difference is that ofs is updated using ifsAndB, as shown in (3.9). At this point all the single faults have been considered, either the ones coming from inputA, or the ones coming from inputB or the ones generated by the AND gate itself.

$$ofs = ofs + (ifsAndB >> 1) \tag{3.9}$$

The analysis done for multiple faults is similar to what is done for single ones but using a derating factor. Equation (3.10) is applied, where derFactor is a derating factor since multiple faults are (still) much less common than single faults.

$$ofs = ofs + ((ifsAndA + ifsAndB) >> derFactor) \tag{3.10}$$

The choice of the derating factor is empirically determined the same as the number of gate inputs plus one. So, for the particular case of the 2-input AND gate, the derating factor is given by derFactor = 3.

Finally, the finished state is reached. The register that stores ofs is already set with the proper value. It is then only a matter of asserting the ready output, so a possible next gate knows it may start its computation.

Going back to the example depicted in Fig. 3.6 (b), once the inverters are done calculating, both internal ofs signals will take a value of 1. Thus, the AND gate takes the following inputs: inputAndA = 1, inputAndB = 1, ifsAndA = 1 and ifsAndB = 1. A fault-free AND gate, when both inputs are 1, will output 1. But, if one or both inputs toggle, the AND gate will output 0. Knowing that, let us proceed with an evaluation, state by state, in order to calculate the final value of the ofs output:

- waiting: according to (3.7), ofs = 2 (the AND gate has a gf value of 2).
- errorOnInputA: according to (3.8), ofs = ofs + (1 >> 1) = 2.5.
- errorOnInputB: according to (3.9), ofs = ofs + (1 >> 1) = 3.
- errorOnBoth: according to (3.10), ofs = ofs + ((1 + 1) >> 3) = 3.25.
- finished: no change to the value of ofs.

Let us reason about the obtained result, $ofs = 3{:}25$. Although the derating factor used is empirically determined, its choice is far from being arbitrary. Let us look back at the circuit depicted in Fig. 3.6 (b) and let us assume a different scenario in which no logical masking takes place and all 3 gates are connected in series (i.e., cascaded). For this particular worst-case scenario, of $s = gf_{inv} + gf_{inv} + gf_{and} = 4$. The issue with such analysis is that multiple faults are considered as likely as single faults, thus deviating from reality. Since our obtained result is below this worst-case threshold, this is a good sign that our result is 'closer to reality'.

In an analogous way, one can think of another scenario in which multiple faults are neglected. In that scenario, the AND gate would be affected from faults coming from either inputAndA or inputAndB, but never both at the same time. In this case, the 0:25 value added during the errorOnBoth state would not be taken into account, thus leading to a ofs value of 3. Once again, this value deviates from reality. Finally, reasonable values for ofs lie within the range of [3,4]. Picking a value closer to 4 means to give a higher probability of occurrence to multiple faults, while choosing a value closer to 3 achieves exactly the opposite effect.

In order to obtain the final reliability figure of the circuit in Fig. 3.6 (b), we resort to an approach similar to the one used in SPR analysis. It is assumed that the circuit under analysis is equivalent to a chain of inverters. The depth of the chain is equal to the value

given by ofs (thus, 3.25 inverters). If it is assumed that a gate with a gf value of 1 is equivalent to a gate with a singular reliability q of 99.999% in SPR's analysis, then the circuit reliability R can be calculated as follows:

$$R = q^{ofs} = 0{:}99999^{3{:}25} = 0{:}9999675 \tag{3.11}$$

Properly choosing q and gf is imperative for a good estimation of the circuit's reliability. Such values can be calibrated in different manners, depending if the goal is to calculate circuit reliability with respect to physical defects or transient faults. For the former case, foundry's data can be used while for the latter laser/ radiation tests can be used. A simple approach is to assume that gf is directly related to area of each gate (i.e., a larger gate has the potential of generating more faults).

The width of the ofs determines the accuracy of the method. A trade-off between accuracy and size of the transformed circuit is naturally established by it. In the experiments that follow, we have used the minimal ofs width value for each circuit, such that no overflow occurs. Another issue is the representation of the reference value gf_{inv}, for which the value 8 was chosen (1000 in binary). Such value is sufficiently high, such that 3 shift right operations will not generate a truncated value. Also, it is a good fit since most of the gates in our experiments have 2 inputs (thus a derating factor of 3 that leads to 3 shift operations). Higher values are also possible (16, 32, etc.) but, then again, they contribute to a larger modified circuit.

An example of an instrumented circuit is given in Appendix C.

3.2.1.2 Modelling Sequential Logic

The modelling behind the sequential logic is much simpler than the one used for combinational logic. The same way a gf value has been defined for each gate, it will be defined for each flip-flop (gf_{ff}). Yet, no logical masking takes place inside a flip-flop. In order to calculate the ofs of a flip-flop, the following equation is used:

$$ofs = ifs + gf_{ff} \tag{3.12}$$

After synthesis, it is quite common to find that flip-flops have a mux-like structure controlling the data input. This mux selects between new data or old data. If the old data is selected, the output of the flip-flop is connect to its input in a loop fashion. One could then rewrite (3.12) as $ofs = ofs + gf_{ff}$, which clearly allows for the accumulation of errors in the flip-flops.

That being said, the problem then becomes to synchronize all the circuit elements properly. Since all the combinational cells from the original description are now described as FSMs, they cannot perceive the same clock signal as the flip-flops in the original circuit description. That issue is solved with the use of special controlling signals that create the same effect of a secondary clock network. The clock signal of the combinational logic toggles every cycle while the one used for the sequential logic only toggles when all ready signals are asserted (i.e., when the combinational logic is done calculating).

The modelling of the sequential portion is by no means what limits the performance of the method. Each flip-flop is roughly transformed into a register array (to store the ofs value) and an adder. The limitations are more likely to come from the combinational portion. The number of clock cycles required to reach a final result (for the sequential logic) is always constant and equal to 1.

3.2.2 Experimental Results

The results shown in this section have been separated by circuit type. Results for purely combinational circuits are given in Section 3.2.2.1 while the results for sequential circuits are given in Section 3.2.2.3

3.2.2.1 Combinational Circuits

Before showing the reliability figures obtained when using SNaP, it is important to clearly state that the (set of) inputs used affects the estimation of the circuit reliability directly. Thus, when an application's input pattern is known it should be used to obtain reliability figures that represent the circuit operation and/ or its neighbourhood environment.

For the combinational circuits used in this section's experiments, such patterns are not available. Therefore random inputs have to be used. Nevertheless, a sufficiently high number of inputs (samples) must be used to achieve a meaningful average reliability. This effort is shown in Fig. 3.7 for the c432 circuit of the ISCAS'85 set of benchmarks.

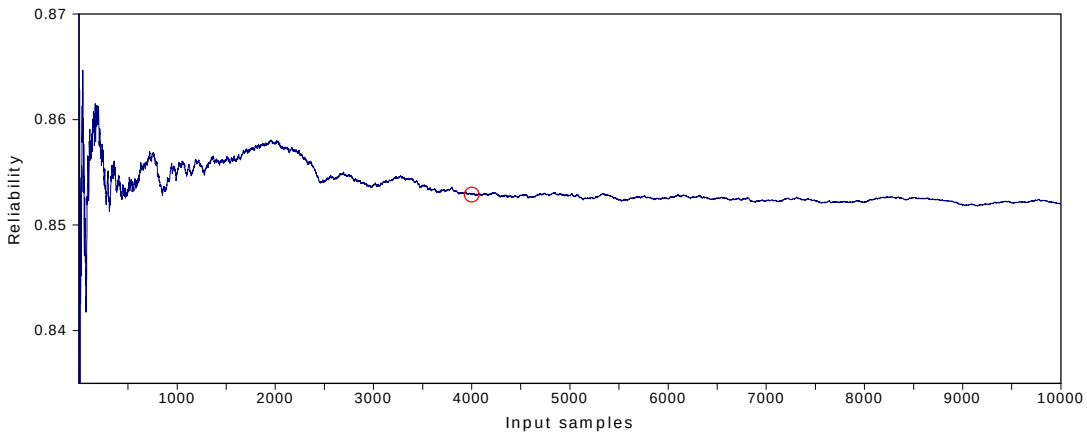


Figure 3.7: Average circuit reliability versus number of samples.

Concerning Fig. 3.7, it shows how the circuit reliability tends to an average value as more and more samples are added. The goal is to determine how many samples are necessary for an evaluation that is a good approximation of the actual average value. In this particular case, for the circuit c432, it is assumed that 4000 samples are enough (as highlighted in red in the image itself). The cost of adding more samples could be unjustified since their contribution is not that important.

When the number of samples is still below 10, some values are outside the range shown in the y axis of Fig 3.7. Values outside the given range rapidly fade away since the average starts to converge to a value close to 0.853. Also, the analysis depicted in Fig 3.7 was obtained by assuming that a gate with gf = 512 is the equivalent of a gate with a singular reliability q of 99.9%. Such value is considered to be low and because of that allows for more variability in the analyzed average.

Still concerning Fig. 3.7, it was generated using simulation and the whole analysis for the 10000 samples takes less than 10 seconds[1]. In order to guarantee that no overflow

[1]The computer used for this analysis has a Intel(R) Xeon(R) CPU E5620 with 8 cores running at 2.40GHz, with 12Gb of memory.

occurs and that the precision is sufficiently high, the width of the ofs registers was set as 32 bits and all gates were set with $gf = gf_{inv} = 512$. Those values are quite high and are not necessarily required to be that high or are not a good match. As previously stated, those two parameters are the key parameters for determining how accurate the method is and, because of that, the manner they are related was also studied.

Once again, the c432 circuit was taken as study case to determine how SNaP parameters are related. Tables 3.3 and 3.4 show different scenarios by varying the width of the ofs parameter and the gf value. The goal of such experiment is to determine which scenarios lead to overflows in any of the ofs registers. The cells mentioning 'Yes' are scenarios that cause overflows and should be avoided.

If an overflow occurs, a large number of faults is being neglected in the analysis, potentially leading to a reliability figure that is much higher than the actual circuit reliability. Since random inputs are being used, the randomization itself is a concern. All the experiments reported here use the same seed, thus guaranteeing that each instrumented version of the c432 circuit was submitted to the same scenarios.

Table 3.3: Occurrence of overflow in the ofs registers for different widths and gf values.

ofs width	gf=4	gf=6	gf=8	gf=10	gf=12	gf=14
8 bits	Yes	Yes	Yes	Yes	Yes	Yes
9 bits	No	Yes	Yes	Yes	Yes	Yes
10 bits	No	No	No	Yes	Yes	Yes
11 bits	No	No	No	No	No	No
12 bits	No	No	No	No	No	No
13 bits	No	No	No	No	No	No
14 bits	No	No	No	No	No	No
15 bits	No	No	No	No	No	No
16 bits	No	No	No	No	No	No
17 bits	No	No	No	No	No	No

The data shown in tables 3.3 and 3.4 shows how several scenarios are not feasible. It also shows a trend for the size of the ofs registers: if gf is encoded using B bits, they should be sized using B + 9 bits. This trend is very clear in Tab. 3.4. The size of the ofs registers is a function of the circuit size, its logical masking, and the value of B. The critical path from a timing point of view is not necessarily the same from the reliability point of view.

The circuit reliability obtained from the same set of configurations is shown in Tables 3.5 and 3.6. The figures in bold are the ones that are not affected by overflows and thus could, potentially, estimate circuit reliability properly.

A trend that is very clear from the gathered data is that increasing the ofs width does not necessarily lead to a more precise reliability value. In fact, there is a point where increasing the ofs registers becomes useless. This value changes according to the gf value used. For the c432 circuit, it seems that 8 more bits for the ofs width is enough. More than that is not necessary and might compromise the size and performance of the method when considering emulation. Other circuits not necessarily have the same '8 extra bits'

Table 3.4: Occurrence of overflow in the ofs registers for different widths and gf values.

ofs width	gf=16	gf=32	gf=64	gf=128	gf=256	gf=512
8 bits	Yes	Yes	Yes	Yes	Yes	Yes
9 bits	Yes	Yes	Yes	Yes	Yes	Yes
10 bits	Yes	Yes	Yes	Yes	Yes	Yes
11 bits	Yes	Yes	Yes	Yes	Yes	Yes
12 bits	No	Yes	Yes	Yes	Yes	Yes
13 bits	No	No	Yes	Yes	Yes	Yes
14 bits	No	No	No	Yes	Yes	Yes
15 bits	No	No	No	No	Yes	Yes
16 bits	No	No	No	No	No	Yes
17 bits	No	No	No	No	No	No

Table 3.5: Reliability figures for different ofs widths and gf values.

ofs width	gf=4	gf=6	gf=8	gf=10	gf=12	gf=14
8 bits	0.92711	0.92276	0.93784	0.93995	0.94675	0.95037
9 bits	0.92705	0.91043	0.90689	0.91718	0.93298	0.92980
10 bits	0.92705	0.91029	0.89954	0.89266	0.88904	0.89272
11 bits	0.92705	0.91029	0.89954	0.89186	0.88529	0.88127
12 bits	0.92705	0.91029	0.89954	0.89186	0.88529	0.88127
13 bits	0.92705	0.91029	0.89954	0.89186	0.88529	0.88127
14 bits	0.92705	0.91029	0.89954	0.89186	0.88529	0.88127
15 bits	0.92705	0.91029	0.89954	0.89186	0.88529	0.88127
16 bits	0.92705	0.91029	0.89954	0.89186	0.88529	0.88127
17 bits	0.92705	0.91029	0.89954	0.89186	0.88529	0.88127

characteristic.

A few interesting trends can be seen in tables 3.5 and 3.6. For instance, by taking a fixed width for the ofs registers and varying the gf value, it is possible to notice that the non-bold reliability figures tend to increase. This is so remarkable that a few scenarios in Tab. 3.6 calculate the circuit reliability as being 1 (100%). In other words, the amount of faults neglected is so high that it looks like the circuit has no faults at all.

Another trend with a completely different behavior can be seen for the bold values. Once again, by taking a fixed width for the ofs registers and varying the gf value, it is possible to notice that the circuit reliability decreases. For instance, the last row of both tables shows how circuit reliability shifts from 0.92 to 0.85 for the same ofs width. The issue then becomes to find, for a given ofs width, which is the gf value that is sufficiently high to estimate circuit reliability. Such analysis is depicted in Fig. 3.8.

Notice that the curves shown in Fig. 3.8 have different values but very similar trends. The 'Reliability given q=0.999' curve is related to the y axis on the left while the 'Reliabil-

Table 3.6: Reliability figures for different ofs widths and gf values.

ofs width	16	32	64	128	256	512
8 bits	0.95251	0.97406	0.98483	0.99429	1	1
9 bits	0.93244	0.95360	0.97437	0.98437	0.99447	1
10 bits	0.89765	0.93532	0.95373	0.97374	0.98475	0.99424
11 bits	0.87733	0.91143	0.93551	0.95171	0.97388	0.98485
12 bits	0.87705	0.86881	0.91480	0.93302	0.95147	0.97414
13 bits	0.87705	0.86577	0.86430	0.91399	0.93223	0.95191
14 bits	0.87705	0.86577	0.86001	0.86158	0.91345	0.93322
15 bits	0.87705	0.86577	0.86001	0.85698	0.86056	0.91690
16 bits	0.87705	0.86577	0.86001	0.85698	0.8556	0.85987
17 bits	0.87705	0.86577	0.86001	0.85698	0.8556	0.8549

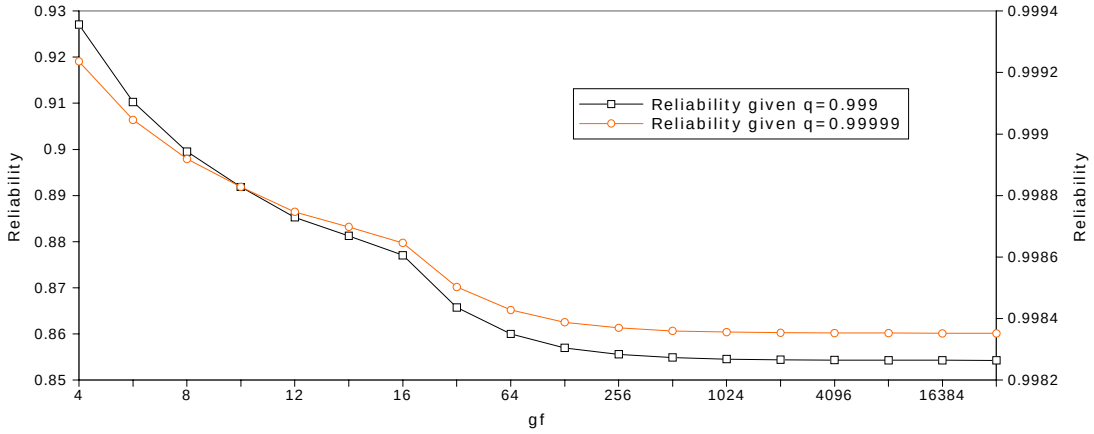


Figure 3.8: Circuit reliability versus gf values (for q=0.999 and q=0.99999).

ity given q=0.99999' curve is related to the y axis on the right. It is also possible to notice that both curves reach sort of a saturation point near gf = 256. Thus, in the experiments that follow, that was the value used for encoding gf .

The results in Tab. 3.7 show the minimum width that must be used to store the ofs values given gf = 256. These values were determined by an exhaustive effort as shown in Fig. 3.7. It is clear that they are not directly related to circuit size, specially in the case of c6288. Furthermore, we have concluded that the minimum width is related to the circuit's logic depth since the number of faults tend to accumulate along the logic paths. Logical masking also influences the width of the registers used to store the ofs values since it inhibits a rapid increase in these values.

Table 3.7 also presents the number of clock cycles that are required for the full analysis of a single combination of inputs. The third column, entitled 'Clock cycles (first result)' shows the number of clock cycles that are required for the method to output its first result. Since the method transforms the combinational circuit in a pipeline-like structure, the results for the following input scenarios can be obtained faster than that. Those are

Table 3.7: Circuit size versus ofs width and number of clock cycles.

Circuit	Number of gates	Width	Clock cycles (first result)	Throughput
c17	6	10	16	6
c432	160	16	1861	514
c499	202	14	106	34
c880	383	14	174	18
c1355	546	15	169	34
c1908	880	16	451	258
c2670	1269	15	223	34
c3540	1669	16	400	258
c5315	2307	14	633	514
c6288	2416	11	617	5
c7552	3513	15	220	34

the values shown in the fourth column, entitled 'Throughput'.

These values are not related to any particular input pattern since the previously described FSMs have a fixed number of states. Regardless of the inputs used, each FSM will always take the same number of clock cycles to reach the finished state.

Circuit c17 is depicted in Fig. 3.9. The path that is highlighted goes through gates g1-g2-g5 and it is (one of) the longest path(s) in the circuit. For each NAND2 gate, the SNaP representation will use an FSM with 5 states. Such FSMs are analogous to the one described in Section 3.2.1.1. Since the longest path has 3 gates, a total of 15 clock cycles is required to reach the first result, as shown in the image. An additional clock cycle is used to activate an internal control signal that tells the user that the computation is finished. Thus, the total of 16 clock cycles shown in Tab. 3.7 is reached.

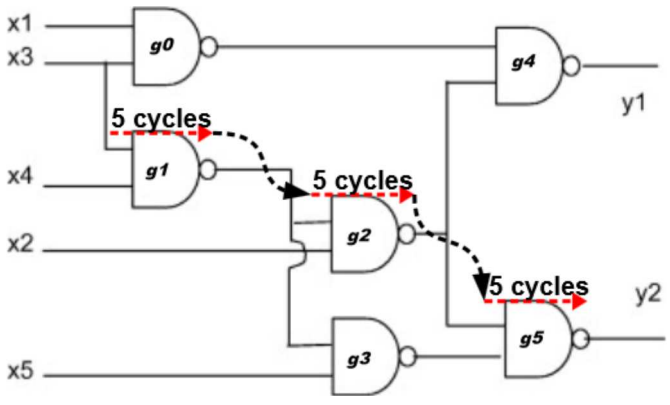


Figure 3.9: Longest path of the c17 circuit and its modelling by FSMs.

The required number of clock cycles to reach the first result can be formally defined as follows: let P_i be one of the circuit paths with length L_i. Such path is composed of L_i gates $g_0:::g_{L_i}$. Let size(g) be a function that returns the number of inputs of a gate g.

Thus, the required number of clock cycles for any given path P_i is given by:

$$Cycles(P_i) = \sum_{k=0}^{k_i} (2^{size(g_k)} + 1) + 1 \quad (3.13)$$

Equation (3.13) can be used to find the third column of Tab. 3.7, given all paths are evaluated and the largest value is found. The values in the fourth column are much easier to find since they depend on the gate with the highest number of inputs. For instance, circuit c499 has gates with 5 inputs, which means 32 cycles where faults are accounted for plus one cycle for the finished state. One additional clock cycle is used for signaling that the computation is ready, totalling the 34 cycles shown in Tab. 3.7.

Other circuits require a longer analysis time. For instance, the circuit c432 has 9-input AND gates in it. If all combinations of up to 9 multiple faults are taken into account, FSMs for this type of gate end up having more than 500 states.

Circuit reliability with respect to different inputs can also be obtained with the proposed method. Figure 3.10 shows the analysis of the c17 circuit using random input patterns and a gf value of 256 for all gates. The image clearly shows that some input scenarios (x axis) can lead to higher ofs values than others (and consequently, lower reliability values). The dashed curves shown in Fig. 3.10 are related to the outputs y1 and y2 shown in Fig. 3.9. Notice that the 'Reliability' curve is inversely proportional to the product of both ofs outputs. Also, in order to obtain the number of fault sites in the same order of magnitude as previously shown in the equations of Section 3.2.1.1, the of s values shown have to be divided by 256.

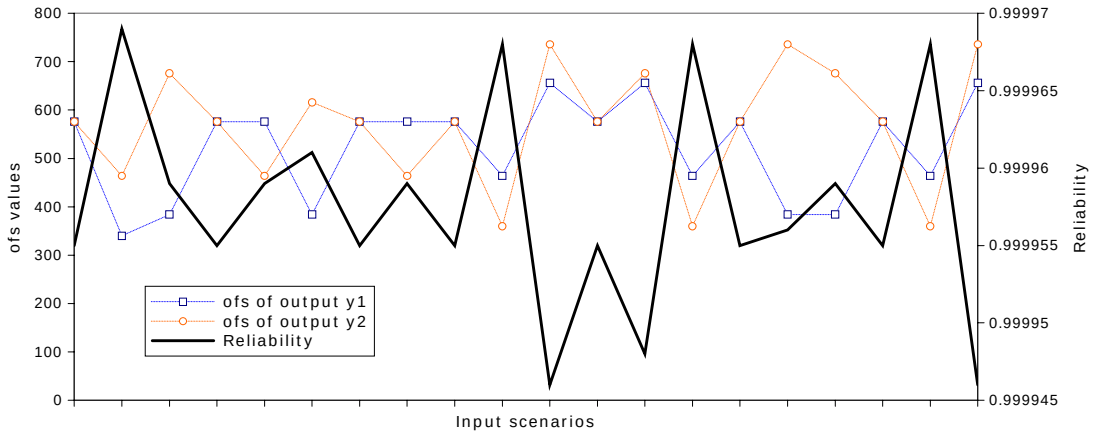


Figure 3.10: Reliability profile versus SNaP's ofs for the circuit c17.

The profiling given by the analysis in Fig. 3.10 can be extended, as shown in Fig. 3.11. Such figure shows the distribution of ofs values (including average and maximum values) for all of the outputs of the circuit c432. This type of profiling can be used to establish hardening metrics. For instance, the image shows that outputs 5 and 6 have the highest value of all outputs (both outputs are tied with 40796, which is the equivalent of nearly 160 fault sites). Thus, a possible hardening strategy would be to improve the reliability of gates connected to these outputs. Another possible approach could opt for hardening gates related to output 2, since that output has the highest average value. Other hardening strategies are also possible but are out of the scope of this section.

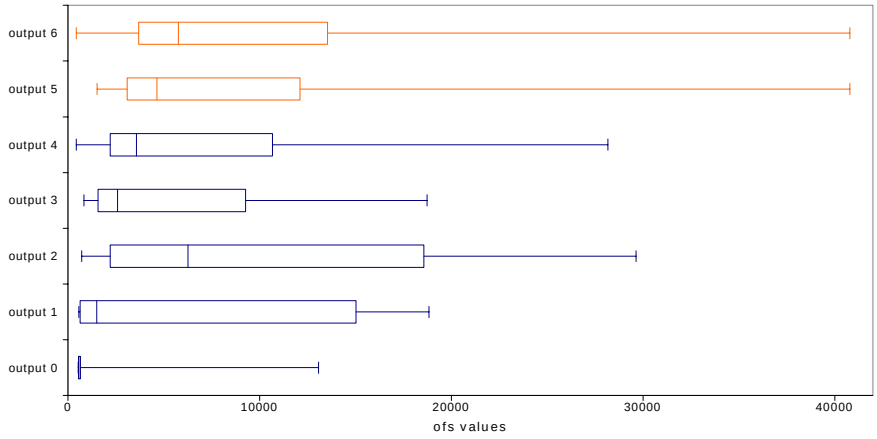


Figure 3.11: Profiling of the ofs outputs for the c432 circuit.

In order to evaluate how much additional hardware is required by SNaP's modelling, all modified combinational circuits have been synthesized and the results are presented in Fig. 3.12. The gf value used was always 256 while the width of the ofs registers changes from circuit to circuit according to Tab. 3.7. Synthesis was conducted using Cadence's RTL Compiler and a 65nm standard cell library provided by STMicroelectronics. As a matter of fact, the modified verilog is targeted for use in an FPGA. Thus, the actual values presented in here are not as relevant as the (scaling) trends.

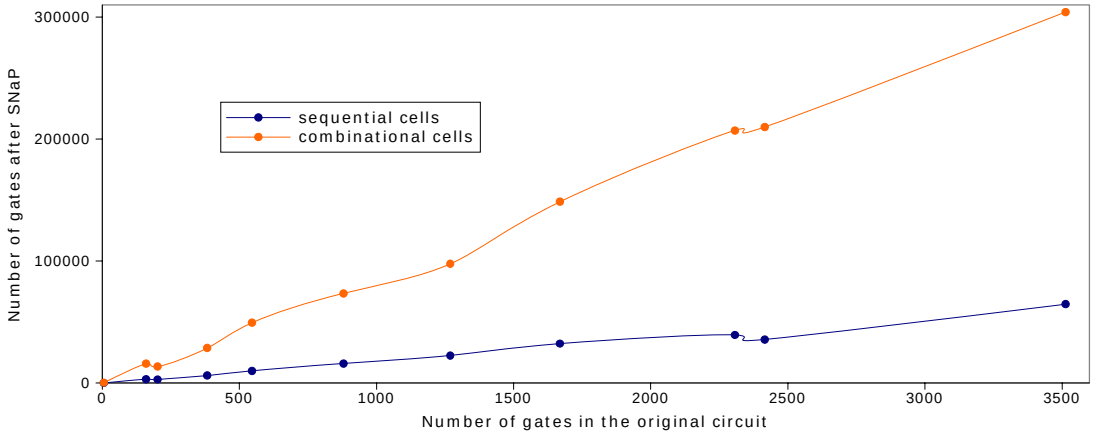


Figure 3.12: Growth trends for sequential and combinational cells.

It is possible to notice that the number of cells increases linearly with respect to the original circuit size. The linear trend is a property of the method. Yet, the slope depends on the implementation's efficiency. Optimizations for reducing the slope of the curves are possible, therefore allowing for the analysis of even larger circuits.

Figure 3.12 shows two curves. The fast growing curve, drawn in orange, shows how much additional combinational logic is present in the modified circuit version. Likewise, the curve drawn in blue shows the amount of sequential logic added. It is clear from the image that the combinational logic grows much quicker and should be targeted for optimizations in the suggested implementation.

Table 3.8 shows the critical path of the synthesized circuits. Once again, the width of the ofs registers was set according to Tab. 3.7. As expected, critical paths are not related to the circuit size. For instance, c17 and c6288 have the shortest critical paths and circuit sizes that are very different. Yet, it seems that the width of the register arrays used to store the ofs results is linked to the critical path. For instance, circuits c432, c1908 and c3540 require 16 bits for storing ofs and they are also the ones with the longest critical paths.

Table 3.8: Synthesis' critical path results.

Circuit	Critical path (ps)
c17	1330
c432	4794
c499	3296
c880	3252
c1355	3390
c1908	4738
c2670	3346
c3540	4722
c5315	3936
c6288	2236
c7552	3380

Data presented in Tab. 3.9 strongly suggests that the number of cells (and therefore circuit area) is deeply related to the width of the register arrays that store the ofs values. For instance, the original version of the c6288 circuit has more gates than c5315. Nevertheless, since they require different widths for the register arrays (11 and 14, respectively), SNaP's representation of the circuit c5315 is bigger than the one for the circuit c6288.

3.2.2.2 Comparison with the SPR Analysis

Given the empirical choices made, it is important to check if the method produces reasonable reliability figures. For that goal, we have performed a comparison with the SPR method. All the cells in the SPR modelling were set with $q = 0{:}99999$. The equivalent was made for SNaP, in which every cell was set with $gf = 10$ (red curve) or $gf = 256$ (orange curve). The results are shown in Fig. 3.13, from which similar trends are verified. The results shown in Fig. 3.13 were obtained through simulation of 4000 input samples for each circuit. Fig. 3.13 demonstrates that both methods are in a good agreement. The accuracy of SNaP is further explored in Section 3.2.3.

Concerning execution times, Tab. 3.10 shows data drom SPR and SNaP. Some of the SPR runs are very short and hard to measure. Instead, the values shown in Tab. 3.10 are obtained by doing 100 SPR runs of each circuit. SNaP was also configured to do 100 runs with 4000 inputs samples in each run. The size of the ofs registers was configured according to Tab. 3.7.

The execution times shown in Tab. 3.10 clearly show that SNaP's execution times are

Table 3.9: Additional synthesis results.

Circuit	Area (μm^2)	Sequential cells	Combinational cells
c17	1468	58	296
c432	113051	3107	15847
c499	89338	2940	13473
c880	196947	6218	28758
c1355	360557	9958	49498
c1908	544449	15979	73452
c2670	711282	22525	97723
c3540	1093300	32320	148691
c5315	1370466	39442	207024
c6288	1365205	35719	209986
c7552	2257612	64719	304149

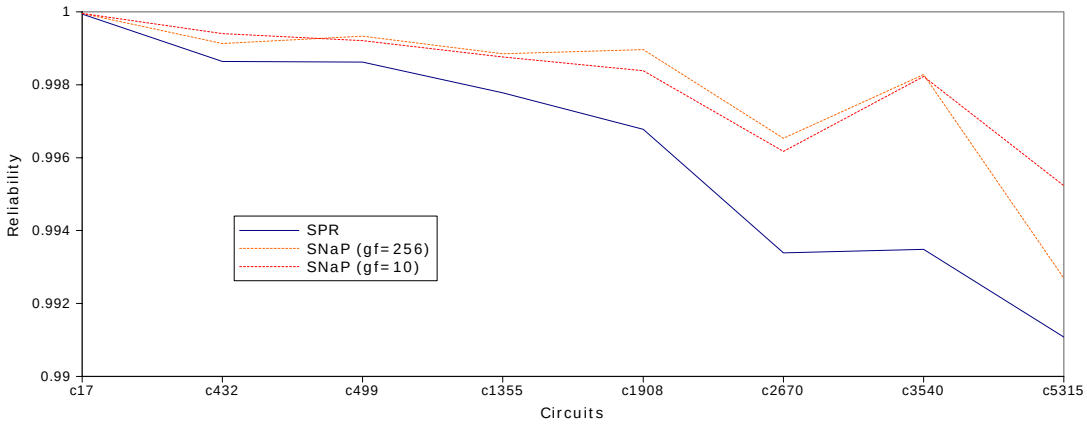


Figure 3.13: Comparison of reliability figures obtained with SPR and SNaP.

larger and that is expected given SNaP's simulation-like approach. Nevertheless, even for the c5315 circuit which has 2307 gates, one execution run takes only 16.8 seconds to complete, which is still practical. Methods like SPR-MP and PTM are completely unable to estimate the reliability of that given circuit. SNaP, on the other hand, is able to, even if accuracy is compromised. The execution times shown in Tab. 3.7 can be even smaller if emulation is to be considered or when using the approach presented in Section 3.2.3.

Furthermore, while other methods usually rely on using probabilities of ones and zeros as inputs, SNaP can use realistic input patterns from an actual application. SNaP differentiates itself from the other methods because of its support of sequencial circuits. Using realistic input patterns is fundamental to capture the behavior of a sequential circuit and, therefore, to be able to assess its reliability. SNaP's modelling of sequential circuits is discussed in the next section.

Table 3.10: Execution times for 100 runs using SPR and SNaP.

	Execution time (s)	
Circuit	SPR	SNaP
c17	0.36	1.65
c432	4.81	104.09
c499	0.34	33.10
c1355	0.75	66.18
c1908	3.18	467.94
c2670	1.88	122.88
c3540	7.77	1054.57
c5315	8.12	1680.85

3.2.2.3 Sequential Circuits

Other than using the ISCAS'85 circuits, SNaP was also validated using a sequential circuit. In order to make a simple yet conclusive analysis, we have chosen to analyze a small microprocessor, referred as mini□p. Such microprocessor was coded in verilog using RTL description. The full verilog code of mini□p is given in Appendix A. The code was later submitted to synthesis using Synopsys's Synplify tool [105]. The output netlist was then used as input in SNaP.

Figure 3.14 shows a simplified block diagram of the studied circuit, which has only fours instructions:

- LDA: register A gets the content of the data input.
- LDB: register B gets the content of the data input.
- ADD: the output result gets A + B.
- SUB: the output result gets A - B.

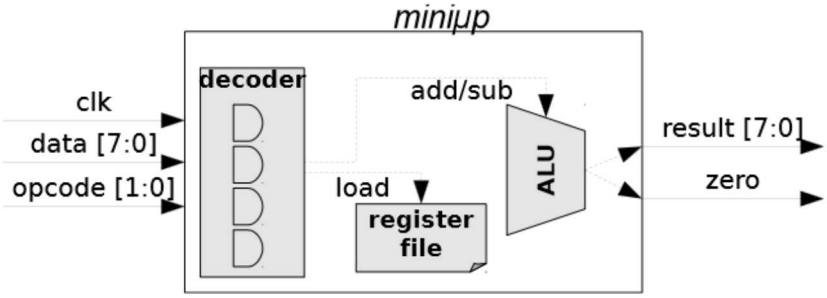


Figure 3.14: Block diagram of the case-studied circuit.

The studied circuit has two outputs: result and zero. All instructions update the 8-bit output result. The output zero is a flag and it is only driven by the ALU, thus it is updated during the execution of ADD and SUB instructions.

Figure 3.15 shows the reliability profile found for the case-studied circuit. Overall circuit reliability R is calculated as the product of the reliability of each output bit, as shown in (3.14). Each term in the product is evaluated as previously shown in (3.11).

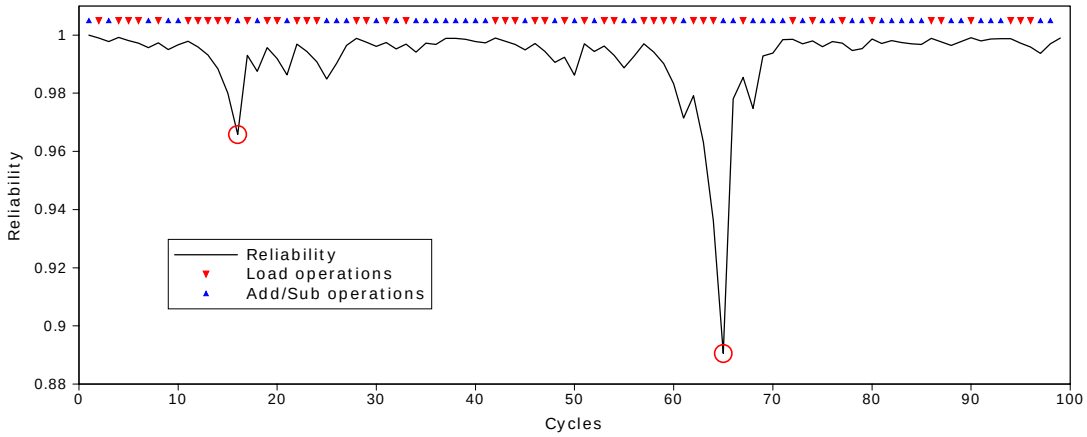


Figure 3.15: Reliability profile in time of the case-studied circuit.

$$R = R_{zero} \cdot R_{result[7]} \cdot R_{result[6]} \cdot \ldots \cdot R_{result[0]} \quad (3.14)$$

Still concerning Fig. 3.15, inputs were left completely random during the execution of 100 clock cycles. This is equivalent to the execution of 100 instructions, which in turn represents nearly 17000 clock cycles for the transformed circuit. The image also highlights two local minima in the profile of the reliability. The instructions that were executed are shown in the top area of the image and it is very easy to notice that those lower points in reliability are associated with sequences of load operations (represented as red upside-down triangles in the image).

Results such as the one shown in Fig. 3.15 allow designers to make high-level changes in the circuit that can lead to a higher reliability. A deeper inspection of the results revealed that long sequences of load operations can cause faults to rapidly accumulate at the zero output (since it is only refreshed by ALU instructions). The other output bits of the circuit maintain a nearly steady reliability figure.

For instance, one solution that would increase the circuit reliability is to allow the load instructions to update the zero output. This solution was implemented and the reliability of the new design is shown in Fig. 3.16 (black dotted line). The new design is referred as mini-p_v2. The lowest point in reliability is a bit under 0.997, near cycle 18. If we go back and analyze Fig. 3.15, we will see that the lowest point in reliability is below 0.9.

Figure 3.16 also shows the ofs values of three different outputs (the other ones were removed and have similar trends to result[0] and result[7]). It is clear that the output zero is still responsible for bringing the reliability down (notice that the y axis on the left has a log scale). The ofs values reported for that output are at least one order of magnitude higher than the ones reported for any other output. This is explained by the fact that the zero output depends on all other output bits (the zero flag is connected to a comparator which is connected to all other output bits). Once again this result can be used to harden the circuit. For instance, hardened gates could be used to build the aforementioned comparator logic, which would certainly bring the reliability to a higher level.

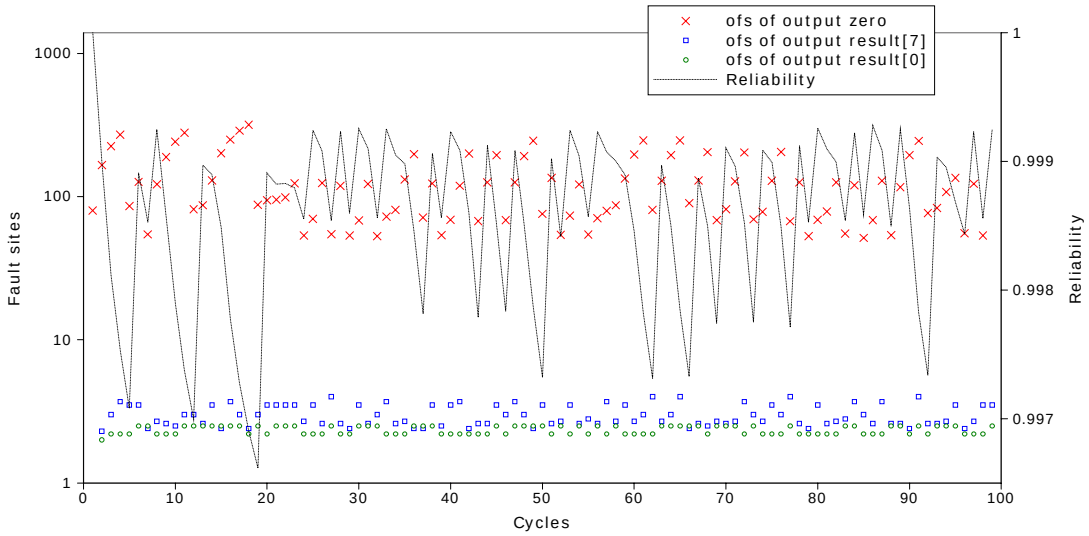


Figure 3.16: Profile of the number of fault sites that can reach three different outputs of the case-studied circuit.

Results concerning the synthesis of sequential circuits are shown in Tab. 3.11 and were obtained using Synopsys' Synplify tool. The target FPGA chosen is the model A3PE3000 from Microsemi's ProASIC3 family [106].

As one can see, the changes promoted by mini□p_v2 have little to no effect to the circuit synthesis results. On the other hand, the synthesis of SNaP's instrumented version increases the gate count and reduced the frequency of operation. The increase in the number of gates is expected but it can be reduced by optimizing the implementation of the method, i.e., these numbers are linked to the implementation more than to the method's principle. The decrease of the frequency is due to the presence of large adders.

Table 3.11: Synthesis results of both mini□p versions and SNaP's instrumented version.

Circuit	Number of gates	Frequency (MHz)
mini□p	142	88.4
mini□p_v2	149	88.8
SNaP's mini□p_v2	38451	35.9

3.2.3 Pessimistic Analysis Using SNaP

A series of parameters was introduced when the SNaP method was presented. Theoretically, if all the parameters are properly calibrated and properly chosen, the reliability estimated by SNaP can be very close to accurate (given a sufficiently large amount of input samples). SNaP does not suffer from the well known fanout reconvergence issue while other methods do. In SNaP's representation a signal is always constant, it does not have a probabilistic profile, which contributes to its accuracy.

This section proposes another approach which has a completely different goal. It tries to eliminate some of the complexity of the method and, by doing so, obtain a pessimistic

figure for a given circuit's reliability. First, all the balancing between single and multiple faults is removed (derating factors for states errorOnInputA and errorOnInputB are set as zero and the accounting of faults done at the errorOnBoth state is neglected). That being said, the method operates in a sort of fault accumulation fashion.

Whenever two or more faults appear on the same signal, there is a chance they will be derated when considering the original SNaP method. That behavior is similar to what happens in the actual circuit, where multiple faults can actually cancel each other. In other words, derating models this effect. Without the derating, faults never cancel each other, they only accumulate. This is why this approach is pessimistic. This is not to be confused with the natural derating introduced by logical masking, which still takes place.

Since derating factors are not used, there is no longer a need to encode the gf_{inv} value with multiple bits. All the experiments reported in this section have used $gf_{inv} = 1$. These changes do not affect the previously shown results for sequential logic since its modelling remains the same. On the other hand, the combinational logic modelling changes and will be discussed in the next paragraphs.

First, the result shown in Fig. 3.7 was also obtained for SNaP's pessimistic approach, i.e., it is still possible to obtain an average circuit reliability value with a relatively low amount of samples. The c7552 circuit was used in the analysis depicted in Fig. 3.17.

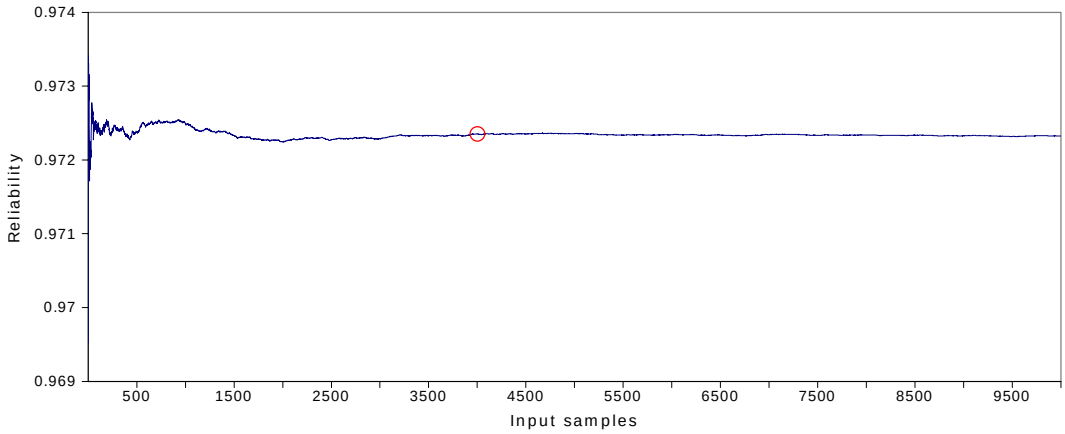


Figure 3.17: Average circuit reliability versus number of samples.

Let us then proceed with a comparison between the reliability figures obtained by both SNaP approaches. Such comparison is depicted in Fig. 3.18. It is clear from the image that the pessimistic approach always gives reliability values that are lower than the ones given by the original version. All the simulations performed for both approaches used enough bits so that no overflow was observed. The values used for encoding gf_{inv} were $gf_{inv} = 256$ for the original approach and $gf_{inv} = 1$ for the pessimistic one.

It is clear from Fig. 3.18 that this approach is pessimistic with respect to the original one. Nevertheless, these results must be compared against the ones generated by an accurate method. SPR-MP was chosen for that purpose. A comparison using the smallest of all the ISCAS'85 circuits, c17, is shown in Fig. 3.19. In order to see more relevant differences between the methods, low reliability values were chosen for modelling the gates. The values chosen are in the [0.99, 0.9999] range and are shown in the x axis of the image.

Two facts should be highlighted concerning the result shown in the image. First, for

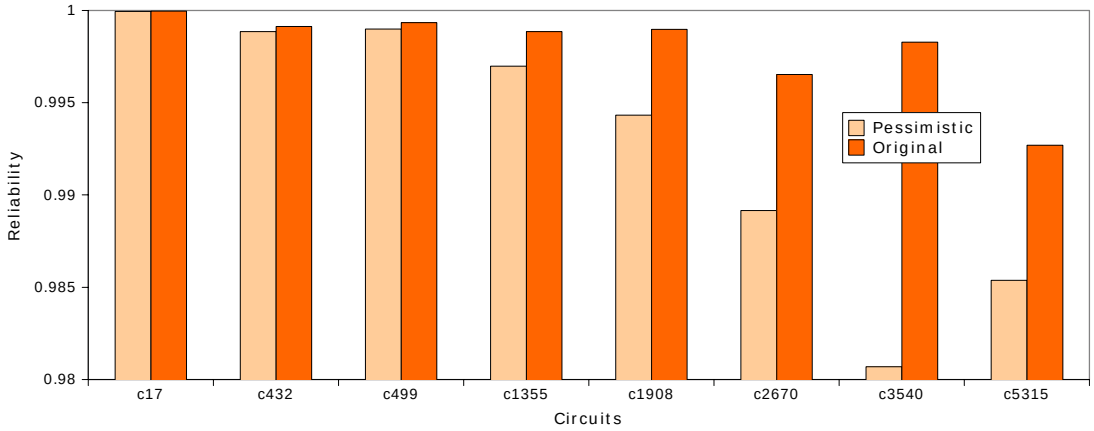


Figure 3.18: Comparison of both SNaP approaches: original versus pessimistic.

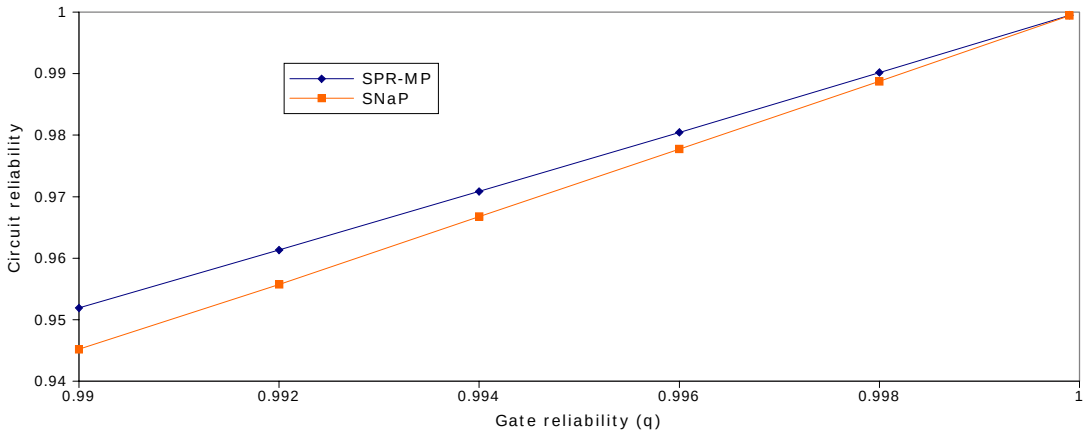


Figure 3.19: Reliability assessment of the c17 circuit using pessimistic SNaP and SPR-MP.

this given circuit, regardless of the gate reliability value, SNaP always gives a reliability value lower than the accurate value. In other words, the pessimistic approach is also pessimistic with respect to SPR-MP. Second, it is possible to notice that the distance between both lines seems to be proportional to the value chosen for the gate reliability (q). The lower the value the higher the difference, which suggests that a fitting function can be used to estimate a more accurate reliability value from SNaP's results. The chosen function is as follows:

$$R = R_{SNaP} \cdot q^{K} \tag{3.15}$$

where R_{SNaP} is the reliability obtained using pessimistic SNaP, q is the gate reliability, and K is a constant. For the c17 circuit that constant is 0:704. The image in Fig. 3.20 shows the fitted results for the c17 circuit.

Another circuit, 74283, was chosen and the same fitting was applied. Such circuit is still small enough that SPR-MP can be used to evaluate all the reconvergent fanouts such that an exact reliability figure is obtainable. The results are shown in Fig. 3.21, in which a similar behavior can be found: two lines that converge with the increase of q's value.

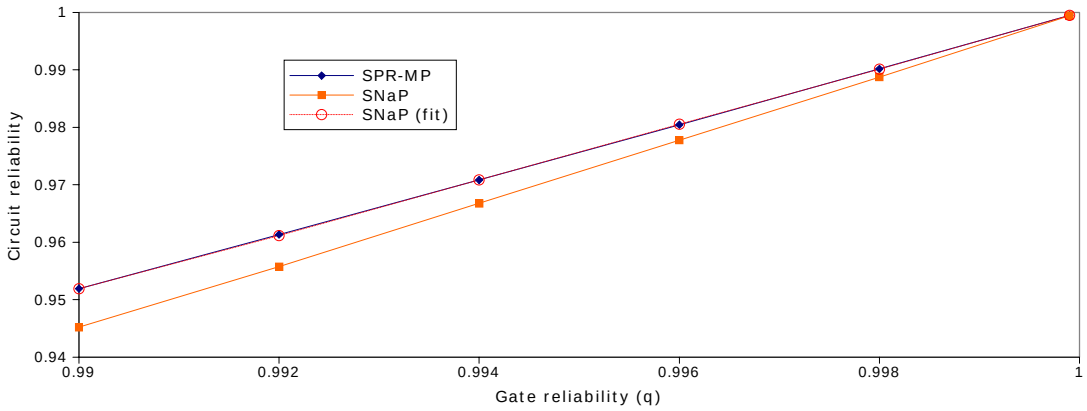


Figure 3.20: Reliability assessment of the c17 circuit using pessimistic SNaP, fitted SNaP, and SPR-MP.

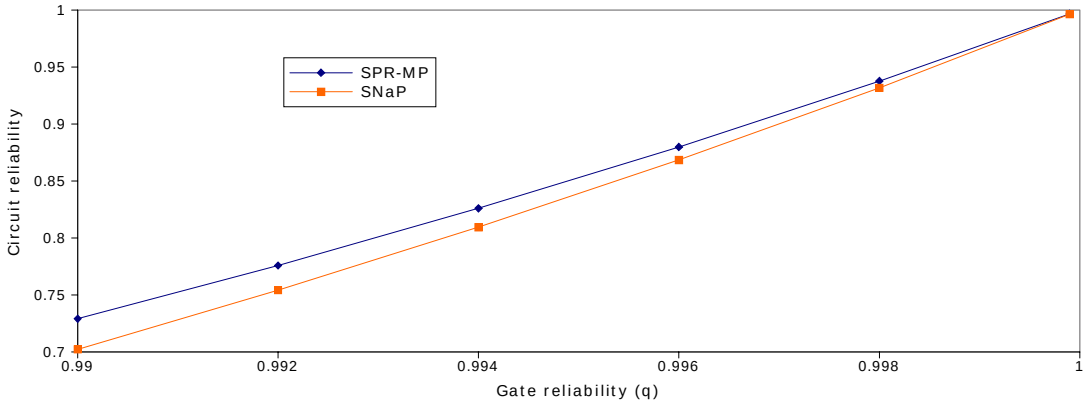


Figure 3.21: Reliability assessment of the 74283 circuit using pessimistic SNaP and SPR-MP.

Figure 3.22 shows the fitted values for the 74283 circuit. Two values were used for the K constant, $K = 0{:}704$ (same as c17) and $K = 3{:}6$.

Once again, it was possible to find a constant that is able to establish a good fit. Nevertheless, it would be interesting if that constant could be defined based on the circuit profile. If that is possible, then the method can be applied to circuits to which SPR-MP cannot be applied. The results previously obtained for the 12th order analysis using SPR-MP were used in the results that follow. Figure 3.23 shows how circuits from different sizes can be fitted using the function given in (3.15).

Each of the circuits shown in Fig. 3.23 uses a different value for the K constant. Those values are listed in Tab. 3.12. Generally speaking, the larger the circuit the larger the value of K. Nevertheless, the relationship between these two parameters also depends on the size of the paths in the circuit. Once again, the higher the number of gates in a path, the higher the value of K (to compensate for the deratings that SNaP does not take into account anymore).

The plots in Fig. 3.24 show how the values for K are related to a circuit metric. Such metric is defined as the number of gates squared divided by the number of primary

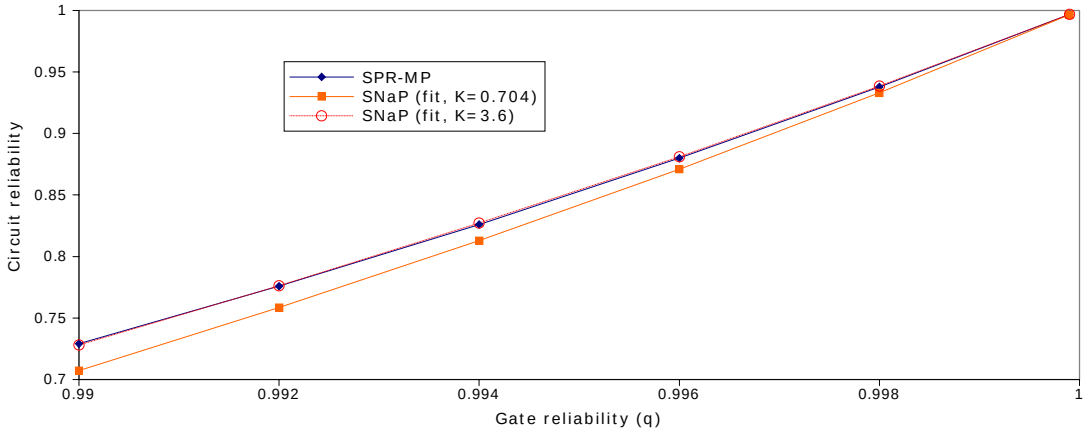


Figure 3.22: Reliability assessment of the 74283 circuit using pessimistic SNaP (different fittings) and SPR-MP.

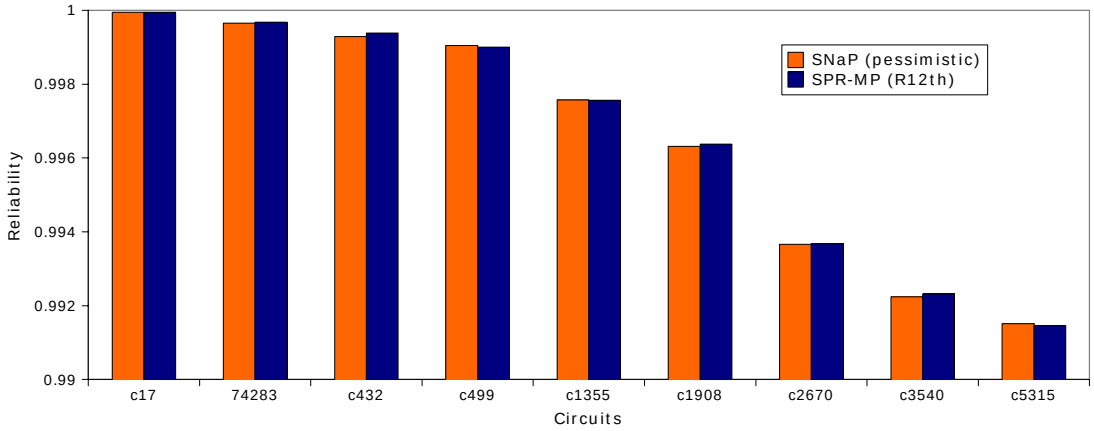


Figure 3.23: Reliability assessment of the 74283 circuit using pessimistic SNaP (different fittings) and SPR-MP.

outputs. It is a simple and fast way of determining an average value for paths' length. This approach can be used for circuits to which SPR-MP is not applicable.

The circuits used in the previous experiments were also synthesized. All synthesis options were the same as in the previously shown results, except the circuits are smaller due to the reduction in the number of bits used. This reduction is shown in the column entitled 'ofs width'. For instance, circuit c499 went from requiring 14 bits to requiring only 7. The critical path is also shorter for several circuits. For instance, circuit c3540 had a critical path of 4722 ps, which was reduced by nearly half, to the value of 2339 ps. One of the reasons for which the critical path is shorter is because there is less logic in the circuit. All the logic that calculated the derating is no longer part of the circuit.

Concerning the circuit size, it was already mentioned that it is proportional to the width of the ofs registers. Thus, a smaller width leads to a smaller circuit size due to less sequential logic. Since the combinational logic is also smaller, circuit sizes are fairly reduced. For instance, the c499 circuit under SNaP's original approach had 2940 flip-flops

Table 3.12: Values used for the constant K.

Circuit	Constant K
c17	0.7
74283	3.6
c499	6
c432	45
c1355	60
c1908	200
c2670	455
c3540	1170

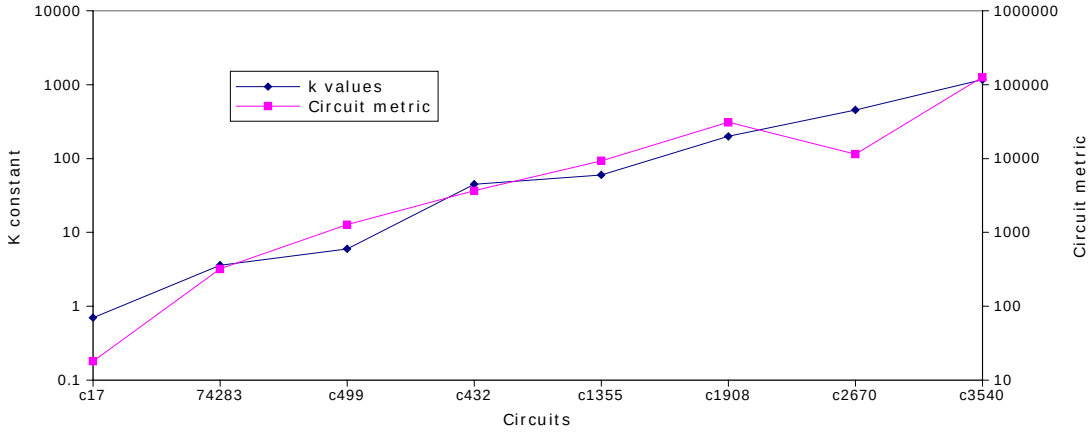


Figure 3.24: Trends for the K values and a circuit metric.

and 13473 combinational cells, for a total of 16413 cells. The synthesis of the pessimistic approach generates 1998 flip-flops and 7530 combinational instances, for a total of 9528 cells. That is a reduction of approximately 40%. Results concerning other circuits are shown in Tab. 3.13.

3.2.4 SNaP Graphical User Interface

A Graphical User Interface (GUI) was developed to aid with the circuit instrumentation process and a screenshot of the developed GUI is shown in Fig. 3.25. Its use is quite simple, the user is only required to browse for the original circuit description and then the tool is able to generate all of the following:

- SNaP's instrumented version of the circuit
- A testbench for simulation purposes
- Scripts for launching the simulation
- Scripts for synthesis

Table 3.13: Synthesis results for the pessimistic approach.

Circuit	ofs width	Cells	Critical path (ps)
c17	3	177	1090
74283	4	1575	1278
c432	8	9345	1882
c499	7	9528	1970
c880	7	18071	2061
c1355	11	35997	2102
c1908	10	55728	2123
c2670	12	88752	3342
c3540	12	119401	2339
c5315	9	149787	3022

The parameters to which the user has access are shown in the interface divided into three parts, one concerning the circuit, one for simulation and one for the modelling options. The meaning of each parameter is as follows:

Reset signal Defines the name of the reset signal in the modified circuit description

Clock signal Defines the name of the clock signal in the modified circuit description

Calculate R each sample Tells the tool that simulation should output reliability figures for each input sample

Calculate average R Tells the tool that it should output the average reliability value at the end of the simulation

Random inputs Simulation inputs are random if checked. Otherwise the user must have a testbench

Print reliability for outputs Tells the tool that simulation should output reliability figures for each output separately

Samples Defines how many input samples should be simulated (cycles of the original description, not SNaP's cycles)

ofs width Defines the width of each ofs register

gf_{inv} Defines the gf_{inv} value. All other fields are analogous but for other gates. Not all gates are shown in the interface, some are set externally

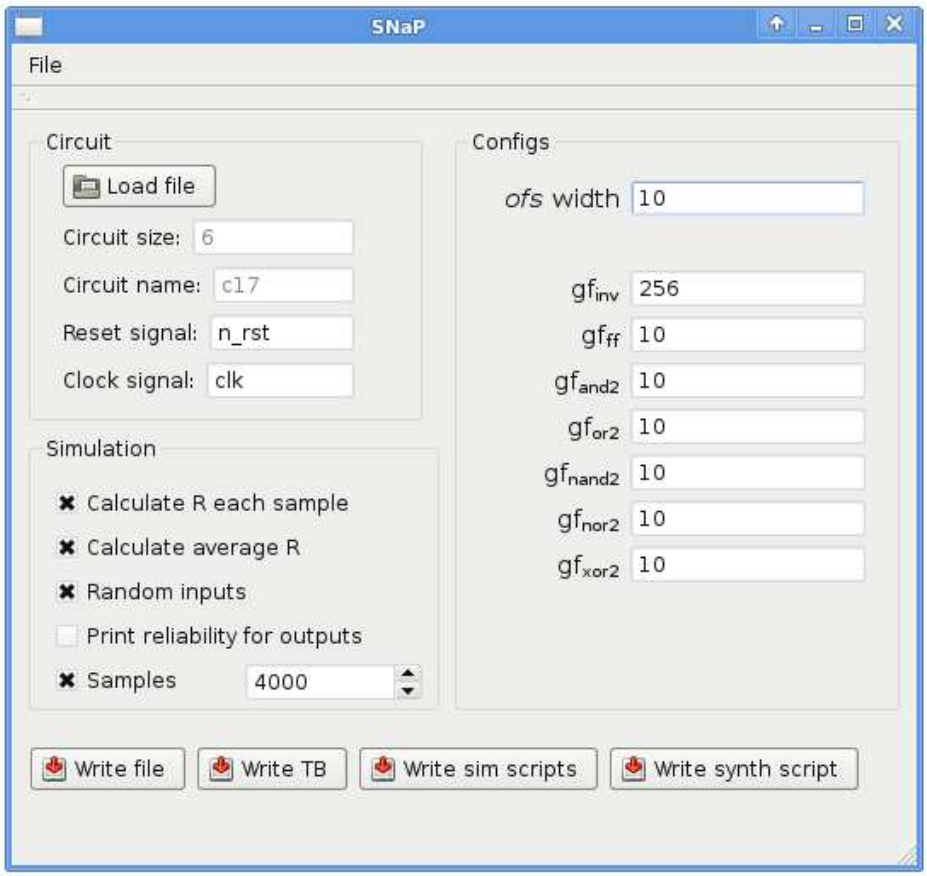


Figure 3.25: SNaP graphical user interface.

Chapter 4

Reliability Improvement Techniques

This chapter covers a series of techniques used to improve the reliability of a given circuit. These techniques go hand in hand with the analysis methods, i.e., there is no reason to improve that that does not need to be improved. And again, once a technique has been applied, the methods are useful once again to estimate how effective its use was.

The techniques showed in this chapter are mainly based in selective hardening. Different aspects of it are studied as to define which are the gates selected to be hardened. At first, a generic cost function is proposed and used to define those gates. Later the same function is modified to take into account the effect of multiple faults and how they manifest in the actual circuit (locality bias). The cost function itself is also the target of the studies here reported and some ways to approximate the cost function profile are also proposed. Last but not least, a technique that uses PQ at the circuit level is proposed.

4.1 A Selective Hardening Methodology for Combinational Logic

It is well known that hardening techniques can be applied in order to mitigate transient errors. The issue with traditional hardening is that it consumes excessive area and/ or energy to be cost-effective (specially concerning commercial applications). Selective hardening, which is applied only to a design's most error-sensitive parts, offers an attractive alternative [92, 107], as previously explained in Section 2.5.2.

This section explores the idea that blocks of a digital circuit (i.e., standard cells) can be classified or ranked with respect to their relative significance to the overall circuit reliability. That ranking takes logical masking into account. With the ranked list of blocks, then it is possible to apply selective hardening either by using HBD techniques or by more generic fault tolerance techniques like TMR.

The work of Naviner et al. [92] and several of the works studied by Polian et al. in [107] assume that different blocks of a circuit can yield different improvements in the circuit reliability. Thus, those methods are able to identify which are the critical gates in a design from a reliability point of view. The work presented in this Section does the same while also performing a cost reasoning of the hardening of each of those gates. For instance, it is quite less costly to triplicate an inverter than to triplicate an XOR gate with 2 or 3 inputs.

In [108], a strategy based on gate-level information was proposed. The method does not take into account any electrical or timing information as a means to select the critical gates of a design while still on its early phases. Although the selection procedure

does not take into account other masking phenomena, simulations of the hardened circuit considering these phenomena were performed and the results suggest that these masking mechanisms have little influence when selecting critical nodes. Later, in [107], the authors have evaluated the validity of choosing critical nodes of a circuit based only on its logical masking ability and have come to the same conclusion. Thus, both papers present enough evidence that logical masking can be used to selective hardening means. This approach was also followed in the methodology reported in this section.

In [86], the use of cost-based selective hardening methodology was proposed. By using an additional hardening affinity parameter, a trade-off between the cost and the reliability gain is then clearly established. Such parameter can be seen as a hardening cost, which allows the designer to drive the methodology using accurate cost values for the hardening of each block. Furthermore, the proposed methodology takes into account the effects of multiple faults since those are more prone to happen nowadays. The methodology itself is scalable, since it relies in an algorithm with linear complexity (SPR). The details of this work are shown in this section.

4.1.1 Preliminaries

The reliability of a given circuit is the degree of confidence observed in the outputs of this circuit, given a certain scenario in which faults are expected to occur with a given probability. In this section, all the obtained reliability figures of a circuit come from using the SPR algorithm, shown in Section 2.4.2. Let us then define signal reliability.

4.1.1.1 Signal Reliability

Signal reliability of a given signal is defined as the probability that this signal carries a correct value. So, it is assumed that a binary signal x can also carry incorrect information. This results in the fact that x can take four different values: correct zero (0_c), correct one (1_c), incorrect zero (0_i) and incorrect one (1_i). Then, the probabilities for occurrence of each one of these four values are represented in matrices, as shown bellow [40, 79]:

$$\begin{bmatrix} P(x = 0_c) & P(x = 1_i) \\ P(x = 0_i) & P(x = 1_c) \end{bmatrix} = \begin{bmatrix} x_0 & x_1 \\ x_2 & x_3 \end{bmatrix} \tag{4.1}$$

The signal reliability for x, noted R_x, comes directly from expression (4.2), where P(:) stands for the probability function:

$$R_x = P(x = 0_c) + P(x = 1_c) = x_0 + x_3 \tag{4.2}$$

4.1.1.2 Reliability of a Block

Digital circuits are composed of many connected blocks, typically standard cells. Let us consider one of these blocks which performs a function on a signal x in order to produce a signal y, i.e., y is the output of the block. The probability that this block fails is given by p, such that ($0 \leq p \leq 1$). Thus, $q = (1 - p)$ is the probability it works properly. Reliability of the signal y can be obtained as:

$$R_y = (x_0 + x_3).q + (x_1 + x_2).p \tag{4.3}$$

Equation (4.3) shows that, when the input signal is reliable, the output signal reliability is given by q. This implies that for fault-free inputs, the reliability of the output signal is given by the inherent reliability of the block that produces this signal. More complex scenarios are evaluated by also taking into account the truth table of the logic blocks (refer to Section 2.4.2).

4.1.2 Selective Hardening Methodology

The reliability of a circuit consisting of several blocks depends on the reliabilities of these individual blocks. This is shown in equation (4.4) for a circuit consisting of K blocks, where R is the circuit's reliability and q_i, q_j stand for the reliabilities of the blocks b_i, b_j respectively ($1 \leq i, j \leq K$).

$$R = f(q_1, q_2, \ldots q_i, \ldots q_j, \ldots q_K) \tag{4.4}$$

Assume that the blocks are independent in the sense that changing the reliability of a given block b_i has no impact on the reliability of another block b_j with $i \neq j$.

If we consider that a reliability change of a single block b_i brings in its new reliability q_i', the circuit's reliability becomes R_i'. Because different blocks b_i and b_j make different contributions to the reliability of a circuit, changes of different blocks may produce different values R_i' and R_j' [92].

The methodology here proposed assumes that there is a hardening technique able to improve the reliability of a given block b_i, such that $q_i' = 1$. This is not a restriction, it is just a simplification, other values are also possible. Then, for all blocks of the circuit, an evaluation run of the SPR algorithm is executed. In each evaluation run, a node b_i is selected, q_i' is allowed to be 1, and the new reliability value R_i' is obtained. This effort is only possible since the complexity of the SPR algorithm is linear.

After all initial evaluation runs are performed, a list of all R_i' values is obtained. At this point, one could sort the list and select the block with the highest R_i' to be hardened. This is the approach followed in [92]. Nevertheless, the interest here is to establish a trade-off between the cost of hardening this block against the cost of hardening any other block. In order to do so, a new parameter Ha_i is introduced, a parameter capable of expressing the hardening affinity of such block.

The parameter Ha_i of each type of cell is defined by the user. It must be constrained in the interval [0,1]. This parameter is generic and can be used to express any type of hardening trade-off: area, delay, power or combinations of the previous. The Ha_i of whichever is the smallest, fastest or less power consuming cell in a library is taken as a reference value and is always 1. Any other cell will have a $Ha_i < 1$.

Such parameter can also be used to model other situation of particular interest: assume that a certain cell is available in two versions in the same standard cell library, let us say XOR_t and XOR_h. The former is a traditional design while the latter is a hardened by design version. In this case, the cell itself is also the hardening technique and it has no additional cost to be implemented. Nevertheless, the cost can be different from XOR_t to XOR_h. When XOR_t is used, its cost is proportional to having 3 XOR_t cells, while the XOR_h cost is proportional to one single instance of it. Nevertheless, this is sort of a special scenario and will not be considered. Our experiments consider that all cells in a library exist in standard versions without any HBD applied.

Table 4.1 shows some of the Ha_i values for some cells that were used in the experiments. These values are extracted from an actual 90nm standard cell library provided by Synopsys [109]. In our analysis we considered that only the dynamic power of the blocks would be considered to calculate the hardening affinity. So, for each cell in the library, we have divided the dynamic power of the smallest inverter in the library by the given cell actual dynamic power. It is possible to notice that negated cells (like NOR and NAND) benefit from the CMOS natural inversion and have a higher hardening affinity. It is also possible to notice that inverters have the smallest dynamic power of all cells. All the other Ha_i values are normalized.

Table 4.1: Hardware affinity (Ha_i) parameters for some cells.

Block	Power (nW=MHz)	Hardening affinity
INVX0	10	1
NAND2X0	3583	0.002790957
NOR2X0	4211	0.002374733
AND2X1	6545	0.001527884
OR2X1	6859	0.001457938
OR4X1	7698	0.001299039
MUX21X1	8639	0.001157541
XOR2X1	8702	0.001149161
AOI21X1	13912	0.000718804

After each cell's affinity is known, it is necessary to use these values to decide which block should be selected for hardening. This step of the methodology introduces a new value, the reliability gain or reliability difference, given by Rg_i. This is the difference from the circuit reliability before (R) and after (R_i') a single block was hardened. For each evaluation run, this value is calculated as follows:

$$Rg_i = R_i' - R \tag{4.5}$$

The Rg_i value obtained from (4.5) is then used to calculate the reliability-affinity product as follows:

$$Prh_i = Rg_i^{w1} \cdot Ha_i^{w2} \tag{4.6}$$

Weights w1 and w2 are applied in (4.6). In other words, the user may choose if reliability should be more important than power or vice-versa, and by which amount. In the experiments that are presented in Section 4.1.3, these values were set as w1 = 2 and w2 = 1.

Once the value of (4.6) has been calculated for all cells, these are sorted and the highest value is picked. This block is then assumed to be hardened and the new circuit reliability (R_i') is obtained. This reliability is then compared against a user-given reliability target. If it is lower than the target, the methodology algorithm starts again and all cells still not hardened are considered as candidates. Otherwise, if the target is met, the algorithm ends and outputs the ordered lists of cells to be hardened.

4.1.2.1 Comparison with an Accurate Reliability Analysis Algorithm

Reliability values used in (4.5) and (4.8) come from SPR analysis. An accurate analysis is possible using the multi-pass algorithm referred as SPR-MP. It is well known that both algorithms produce different values for the reliability of a circuit. Yet the interest is to compare how well SPR estimates the critical node in comparison with the actual critical node that would be obtained with SPR-MP.

Let us first consider a simple circuit, c17, which has only 6 nodes. By applying the methodology using both algorithms just once and neglecting a reliability target given by the user, two lists of b_i nodes are created. These lists are sorted according to the $R_i^{\square}$ of each node and are referred as L_b. These lists are showed in Tab. 4.2.

Table 4.2: Comparison of the ranking of critical nodes obtained with either SPR or SPR-MP algorithms.

Position	SPR-MP's L_b	SPR's L_b	Position difference	Normalized difference
1st	2	4	1	0.2
2nd	4	5	1	0.2
3rd	5	2	2	0.4
4th	1	1	0	0
5th	0	0	0	0
6th	3	3	0	0
			Average error:	0.133

The meaning of each column of Tab. 4.2 is as follows:

- Position is the ranking of the nodes according to $R_i^{\square}$.
- SPR-MP's L_b is the list of nodes generated by the SPR-MP algorithm, i.e., the accurate list.
- SPR's L_b is the list of nodes generated by the SPR algorithm.
- Position difference is the difference in the ranking from column 3 with respect to column 2. For instance, block 4 is ranked first in the SPR's L_b list while it is ranked second in the SPR-MP's L_b list, thus the difference is of one position.
- Normalized difference is the position difference given in column 4 divided by the maximum position error possible. In this example it is 5 since the circuit has 6 blocks.

According to the analysis of the circuit c17 presented in Tab. 4.2, the average error introduced by the SPR algorithm is 13.3%. This same analysis was performed for other circuits with different profiles, all containing multiple reconvergent fanouts branches. The selected circuits are very limited in size since the execution times of the SPR-MP algorithm are very high even for medium-sized circuits. The details of the chosen circuits are as follows:

- 74283, a 4 bit adder.
- AOI, which contains an and-or-inverter logic and 2 multiple fanouts.
- AOIX2, which contains a larger and-or-inverter logic followed by a buffer network with many multiple fanouts.
- decoder, which contains a large or-like logic to decode 8 inputs. Each input feeds many gates so reconvergent fanouts appear already in the inputs.
- chain, which contains a chain of inverters and OR gates.

Table 4.3 summarizes the comparison of all the presented circuits plus the already presented c17 one.

Table 4.3: Comparison of the methodology using the SPR and SPR-MP algorithms.

Circuit	Minimum error	Average error	Maximum error
c17	0	0.133	0.4
74283	0	0.07	0.28
AOI	0	0	0
AOIX2	0	0.35	0.85
decoder	0	0	0
chain	0	0	0

The results in Tab. 4.3 clearly show that for some circuits both algorithms produce the same list of blocks to be hardened. Yet, for the circuits c17, 74283 and AOIX2 the maximum error is quite high. A deeper analysis is performed for such circuits, where the error distribution is analyzed.

The error distribution for the 74283 circuit is shown in Fig. 4.1, where the nodes in the x axis are sorted according to R_i (the actual labels are the ids of the blocks in the circuit). It is possible to notice that some blocks have a higher error probability. That is not the case for the blocks that are closer to the y axis (which are exactly the best candidates for hardening). This same profile, where the error is not that high in the elected block, is also seen in the error distribution of the other circuits. The same profile is also observed after some cells have already been elected for hardening. Thus, our results are presented using the SPR algorithm only, which allows the analysis of larger circuits from the ISCAS'85 set [110].

In the particular case of the AOIX2 circuit, the higher error values are due to the relatively high large number of fanouts and it is shown in Fig. 4.2. Regarding the circuit c17, it only has 6 cells. So any small difference is very meaningful, even in relative terms.

4.1.3 Experimental Results

The methodology described in Section 4.1.2 was applied to several ISCAS benchmark circuits [110]. Each block from each circuit was set using $q_i = 0{:}9999$. The reliability target was adjusted so a decrease of the unreliability would be reached for each circuit. The results are presented in tables 4.4 and 4.5. The former table contains the results for

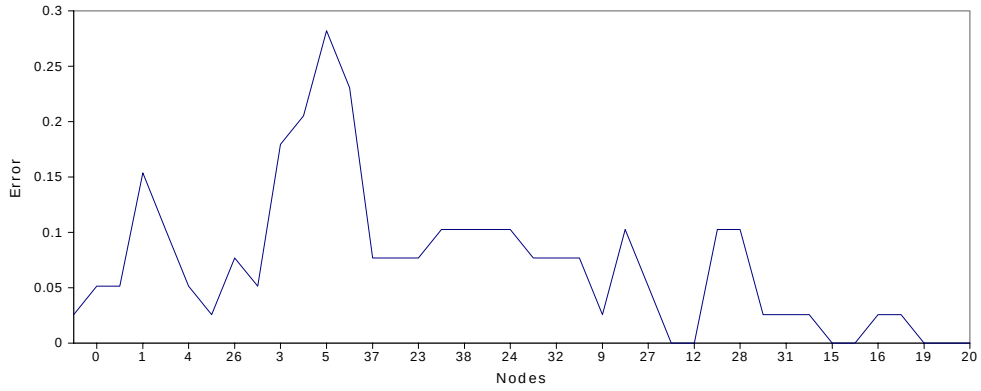


Figure 4.1: Error distribution for the circuit 74283.

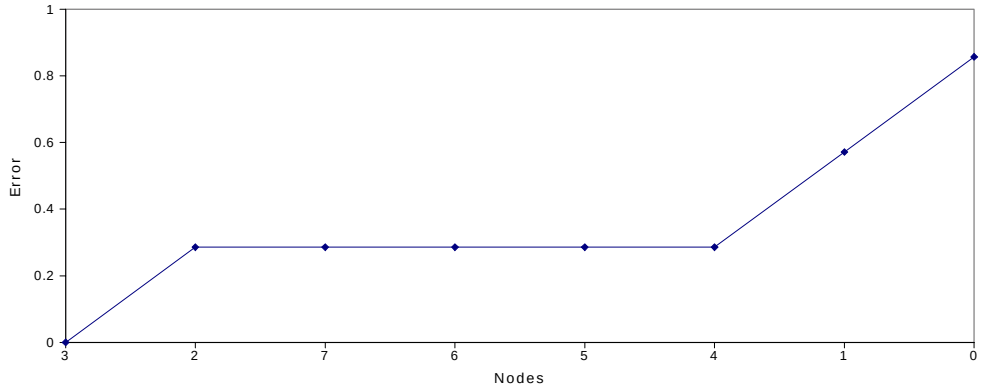


Figure 4.2: Error distribution for the circuit AOIX2.

a reduction of at least 20% (with respect to the original unreliability) while the latter contains the results for a reduction of at least 40%. The original unreliability of each circuit is given in the second column of the aforementioned tables.

The column entitled "Original Power" contains the sum of the dynamic power of all cells of each circuit. The columns entitled "Hardened Cells" contain the amount of cells that are selected for hardening. By using the hardening affinity parameter, this number tends to increase. Then, the columns entitled "Power" contain the sum of the dynamic power of all cells of the new version of the circuit. A fairly simple assumption was made: on hardening a given node we should add three times the value of the power of that node to the overall circuit power.

Thus the additional power that would be consumed by a voter is not considered. Once again, this is a simplification. A voter can be considered for a group of cells and not for a single cell, otherwise the costs can become unfeasible quite fast. Assuming one voter for each hardened cell would create a large cost both in terms of area and power. Therefore the power figures given in the tables are a minimum value estimate. Voter placement (i.e., TMR granularity) is not the scope of this work.

In tables 4.4 and 4.5, some power figures are highlighted in bold. It is clear that applying the methodology considering the hardening affinity is an effective trade-off between power and reliability in those cases. This does not mean that the methodology is not ap-

Table 4.4: Results for decreasing the unreliability by at least 20%.

Circuit	Original Unreliability	Original Power (nW=MHz)	No hardening affinity		With hardening affinity	
			Hardened cells	Power (nW=MHz)	Hardened cells	Power (nW=MHz)
c17	0.000562	21498	1	21498	1	21498
74283	0.003848	189244	4	222932	8	189404
c432	0.013466	624686	9	624866	9	624866
c499	0.013611	1321460	20	1669540	41	1322280
c1355	0.021905	1907300	38	2179608	38	2179608
c1908	0.031668	2146539	58	2147699	58	2147699
c3540	0.062635	5.90e+06	54	5.90e+06	54	5.90e+06
c2670	0.064015	4.07e+06	41	4.12e+06	42	4.08e+06
c5315	0.085614	8.89e+06	59	8.96e+06	60	8.90e+06

propriate for the other circuits. In fact, it means that the current choice of parameters w1 and w2 might not be a good one for the circuits that are not highlighted.

Figure 4.3 shows only the circuits for which the methodology (and its chosen weights) is effective. The data are the same from the tables, thus the same scenario applies: same values of parameters w1 and w2 and voter cost is neglected. The power values shown on the y axis are normalized with respect to each circuit's original power.

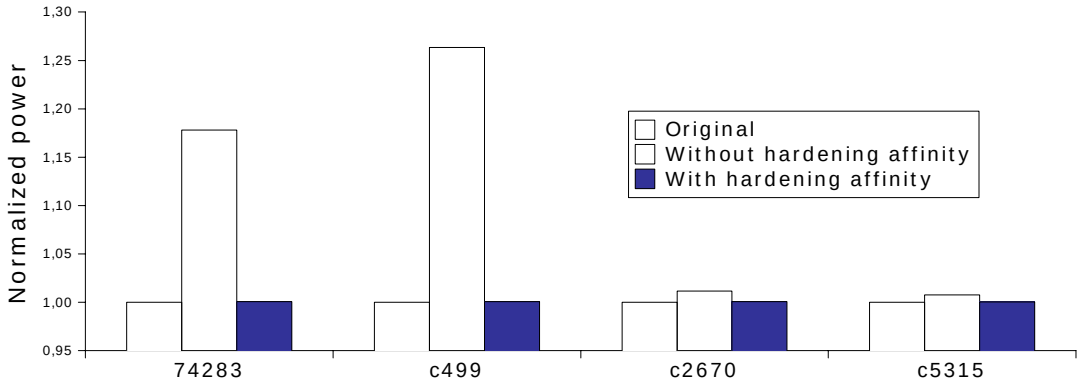


Figure 4.3: Normalized power values for selective hardening with and without hardening affinity.

4.1.3.1 Comparison

A straightforward comparison with other methodologies is not simple since the goals are different. The results presented in [92] are in alignment with the results presented in this work, which is a strong suggestion that multiple faults do not have a large impact on the decision of which node to harden. Multiple faults have a considerable impact on the actual reliability of a circuit. Thus, they are important when determining the trade-offs

Table 4.5: Results for decreasing the unreliability by at least 40%.

Circuit	Original Unreliability	Original Power (nW=MHz)	No hardening affinity		With hardening affinity	
			Hardened cells	Power (nW=MHz)	Hardened cells	Power (nW=MHz)
c17	0.000562	21498	2	35830	2	35830
74283	0.003848	189244	10	273464	16	189564
c432	0.013466	624686	26	625206	26	625206
c499	0.013611	1.32e+06	48	2.15e+06	80	1.42e+06
c1355	0.021905	1.90e+06	83	2.50e+06	83	2.50e+06
c1908	0.031668	2.14e+06	132	2.14e+06	132	2.14e+06
c3540	0.062635	5.90e+06	175	5.90e+06	175	5.90e+06
c2670	0.064015	4.07e+06	128	4.22e+06	128	4.08e+06
c5315	0.085614	8.89e+06	205	9.13e+06	207	8.90e+06

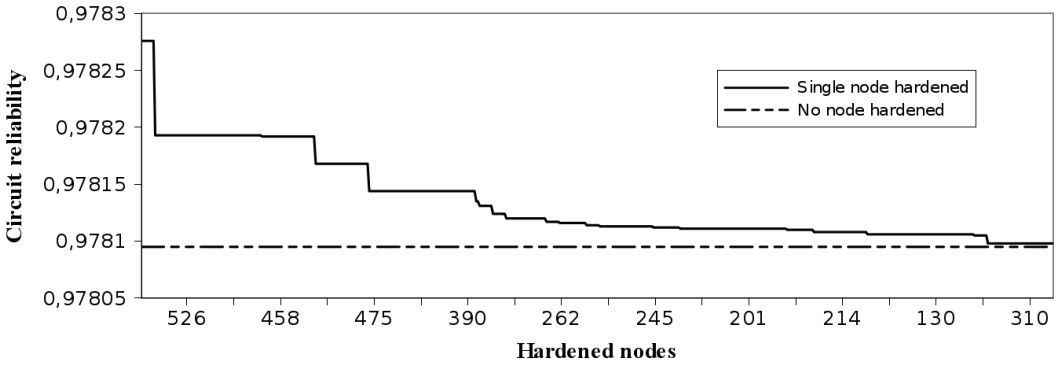


Figure 4.4: Reliability gain versus chosen node to be hardened for the c1355 circuit.

between cost and reliability.

A radiation hardening technique for combinational logic is proposed in [94]. The hardening is achieved by increasing the gate size of some critical nodes in the circuit but no hardening against defects is mentioned. Thus the technique presented here is more of a general solution since it is technology-independent. The overheads mentioned in [94] are not directly comparable.

Nevertheless, in qualitative terms, it is easily observed that certain cells have a larger impact in the reliability of the circuit than others. This observation is highlighted in [92–94]. The same was also observed in the experiments shown in Section 4.1.3. There are some particular cases, like the one illustrated in Fig. 4.4, where choosing the correct node to harden has a large impact in the overall circuit reliability. The analysis depicted in Fig. 4.4 is from the circuit c1355.

Regarding Fig. 4.4, it contains the $R_i^{\square}$ values related to the hardening of all possible cells. The nodes in the x axis are ordered by the reliability gain that hardening that node would produce. The circuit was evaluated given the parameter $q_i = 0{:}9999$ for each cell. In absolute terms the difference from the best to the worst candidate is not large. Yet,

usually several cells are selected for hardening (as in Tab. 4.5), so these values accumulate. Thus choosing the best candidate for hardening is critical.

It must also be mentioned that the methodology can be integrated in commercial design flows in a very straightforward manner. Other cost-effective schemes are also possible since the methodology has a generic parameter to define the hardening affinity of a node. The study of the relationship between parameters w1 and w2 is a matter for a future work.

Following the work shown in this section, we have proceeded with the investigation of multiple faults and their effects in the reliability of a circuit. Section 4.2 shows a modified methodology that limits the effect of multiple faults locally in the circuit.

4.2 Net Hardening: A Heuristic-Based Locality Bias for Selective Hardening Against Multiple Faults

It was shown in Section 4.1 that a cost-based solution can reduce the amount of additional hardening required by a fault tolerance technique. It was also shown that multiple faults are more common and therefore should be properly handled.

Concerning those multiple faults, their source determines the locality profile of the faults. Physical defects can be randomly distributed in the circuit but they can also show locality patterns [13, 14]. On the other hand, multiple faults caused by SEEs always have a locality profile. What is presented in this section is a modified version of the methodology presented in Section 4.1. The works described in [87] and [90] describe the methodology and are summarized in this section.

4.2.1 Introducing a Heuristic-Based Locality Bias

When a digital circuit is designed using standard cells, a placement step is executed during the design flow. Those gates that are logically connected have a certain probability of being actually physically close to each other. Since placement algorithms [111] try to reduce wirelength, it makes sense that those cells are close to each other. And, since these cells are close enough to each other, they might be susceptible to charge sharing effects. Several parameters related to the technology being used have to be considered to determine if the effect itself occurs, but given its occurrence, having a short distance between nodes is mandatory.

The illustration in Fig. 4.5 depicts a scenario in which multiple faults could occur. The image shows three rows of standard cells and the ANDX0 cell in the first row is considered the strike site of an energetic particle. The area under the red circle represents the neighbourhood region of the struck node. Such neighbourhood is susceptible to charge sharing, i.e., the red circle represents the charge sharing cloud. The cells drawn in yellow are the ones that could possibly be affected (i.e., they are inside the considered radius thus they might have their outputs toggled). The cells represented in light blue are the ones not affected.

In [26], it is stated that, when considering simulation as a means to measure the effects of single-event-induced charge sharing, it is not realistic to inject multiple random faults. Such approach can easily overestimate the actual circuit sensitivity to single event effects. The affected nodes must be placed together in a certain minimum distance for this phenomenon to occur. Such unrealistic scenario is depicted in Fig. 4.6. It is possible

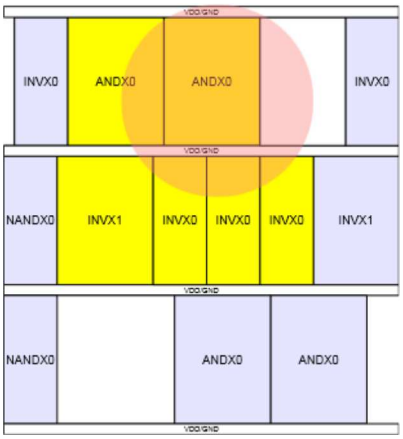


Figure 4.5: Representation of multiple faults according to charge cloud.

to notice that the analysis could be completely inaccurate, since the ANDX0 cell that is exactly under the charge cloud is not taken into account while cells that are 2 rows away from the strike site are.

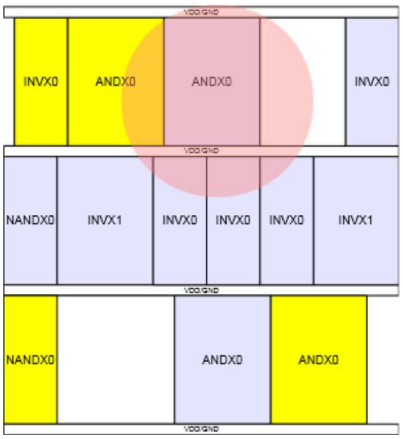


Figure 4.6: Representation of multiple faults according to random modelling.

In accordance with [26], the use of a locality bias was introduced when performing selective hardening. The bias is used here as a heuristic and is introduced through the concept of net hardening. Instead of hardening a single cell or a set of random cells, only nets are considered. Hardening a net means hardening all the cells that are logically connected to it. This is represented in Fig. 4.7.

Many studies have been performed with the sole goal of characterizing the charge sharing profile [112–114]. Factors such as particle's energy, angle of incidence and the type of the device that was struck (i.e., NMOS or PMOS) should be considered for obtaining a realistic scenario. In our analysis, we assume a particle always has enough charge and intensity, thus all cells in a given net are affected by the strike. This approach remains as an approximation of the real behavior since some cells might not be considered, such as both inverters in the middle row of Fig. 4.7.

It is also important to mention that this type of analysis does not require a placed

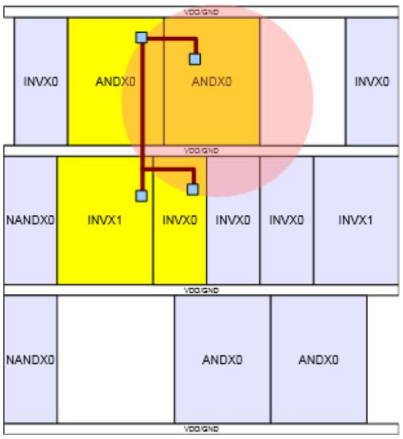


Figure 4.7: Representation of multiple faults according to the net-based analysis.

circuit. Any circuit that has undergone logical synthesis is suitable since the information on how cells and nets are connected is already known. Nevertheless, such information is not precise since placement and optimizations are still to take place. Thus, our strategy is similar to a heuristic: accuracy is compromised for time, i.e., the analysis might be executed earlier in the design flow. A fruitful discussion regarding the validity of an approach similar to this is given in [108], in which the authors verify that the improvements predicted at gate-level analysis are indeed obtained at the final circuit.

4.2.1.1 Node Selection

First, each cell in the circuit is already characterized by a q_i value, the singular reliability. Such value expresses how reliable a cell is. Similarly to what was shown in Section 4.1, it is considered that a reliability change of a single cell b_i brings its new reliability to q_i' and the circuit's reliability R then becomes R_i'.

In this modified version of the methodology shown in Section 4.1, it is still assumed that there is a hardening technique that is able to improve the reliability of a given block b_i, such that $q_i' = 1$. The difference here is that it is assumed that, when a cell is hardened, its area becomes three times bigger than before. The choice of this value was inspired by the TMR technique (although the cost of adding voters is neglected). The analysis does not rely on TMR, any hardening technique can be applied, as long as the cost given by the increase in area is properly modelled. These considerations are not restrictions, they are just simplifications.

That being said, what follows is that for all nets of the circuit, an evaluation run of the SPR algorithm is executed. In each evaluation run, all cells that are connected to a given net are selected, q_i' of each cell is allowed to be 1, and the new reliability value R_i' is obtained (check Fig. 4.8). In general, the number of nets in a circuit is from the same order of magnitude of the number of gates. Once again, the effort of analyzing every net is only possible since the complexity of the SPR algorithm is linear.

After all initial evaluation runs are performed, we obtain a list of all R_i' values (R_i' is defined exactly as in (4.5), except nets are used instead of gates). The size of the list is proportional to the number of nets in the circuit. At this point, one could sort the list and select the net with the highest R_i' to be hardened. A graphical representation

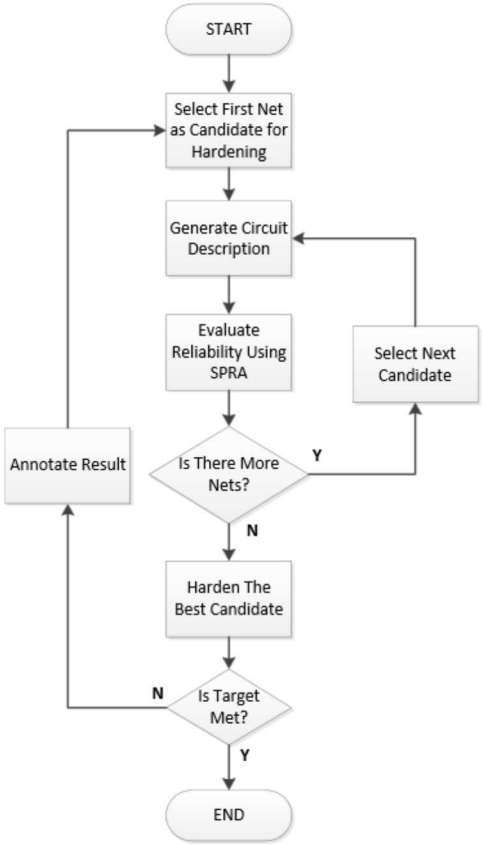


Figure 4.8: Net hardening analysis flow.

of this is given in Fig. 4.9, in which is possible to observe that some nets (the ones in the left portion of the image) are much better candidates for hardening than others. Still regarding Fig. 4.9, the numbers in the horizontal axis represent the identifiers of each net.

Hardening the candidates from Fig. 4.9 would be similar to apply the approach described in [92]. Nevertheless, the interest is to establish a trade-off between the cost of hardening a single net versus the cost of hardening any other net. In order to do so, a new parameter is introduced to express the hardening affinity of a cell, given by Cha_i. Notice that this parameter is slightly different from the one previously shown in Section 4.1. In the previous definition of Cha_i, no value is higher than 1. In this new definition values can be higher than 1. Later, this parameter will be able to provide a comparison basis between nets.

The parameter Cha_i of each type of cell can be defined by the user manually or generated by the analysis itself (based on a library description). This generic parameter still can be used to express any type of hardening trade-off: area, delay, power or combinations of the previous.

In Tab. 4.6, some of the values that were used in the experiments are shown. These are extracted from the same 90nm standard cell library provided by Synopsys [109]. Area was the only parameter taken into account to calculate the hardening affinity Cha_i shown in Tab. 4.6 . Then, for each cell, the given cell actual area was divided by the area of the smallest inverter (INVX0) in the library.

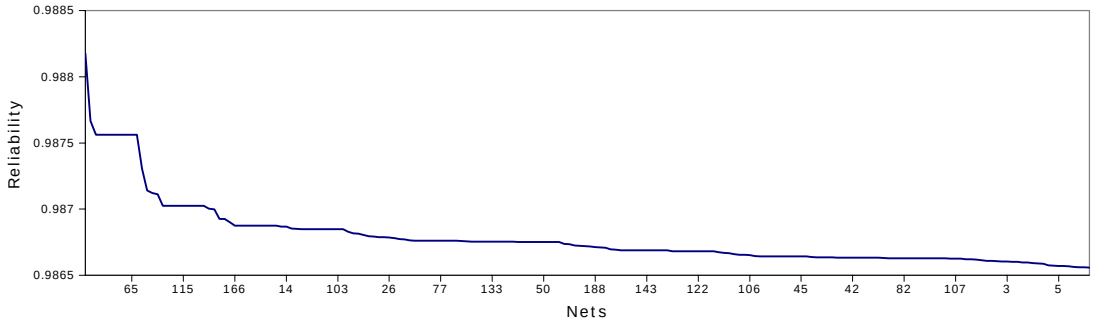


Figure 4.9: Reliability versus chosen net to be hardened.

Table 4.6: Hardening affinity (Cha_i) values for some commonly used standard cells.

Cell	Area (μm^2)	Cha_i
INVX0	5.5296	1
NAND2X0	5.5296	1
NOR2X0	5.5296	1
INVX1	6.4512	1.166
AND2X1	7.3728	1.333
OR4X1	10.1376	1.833
XOR3X1	22.1184	4
INVX16	25.8048	4.666

In order to define the affinity of the nets, another parameter is introduced: Nha_i, which is defined as the sum of all the Cha_j values, given that j is a cell connected to a net i, as shown in the following equation:

$$Nha_i = \sum Cha_j \tag{4.7}$$

The Nha_i values of the nets are then used in a cost function, which provides the trade-off between reliability and hardening cost. The higher the value of Nha_i, the worst candidate the net is. Before using the cost function, all Nha_i values are normalized, i.e., divided by the highest Nha_i value found. Doing this sets all Nha_i values in the [0, 1] interval (since the reliability gain is also in the same interval). The cost function is then expressed as follows:

$$C_i = (Rg_i)^{w1} = (Nha_i)^{w2} \tag{4.8}$$

where w1 and w2 are weights to be applied. By using those weights, it is possible to choose if reliability should be more important than hardening costs (or vice-versa), and by which amount. Finally, the net with the higher value of C_i is selected for hardening. If the target reliability improvement has not been reached then another round is performed to choose the next net for hardening. Such iterative analysis flow is depicted in Fig. 4.8.

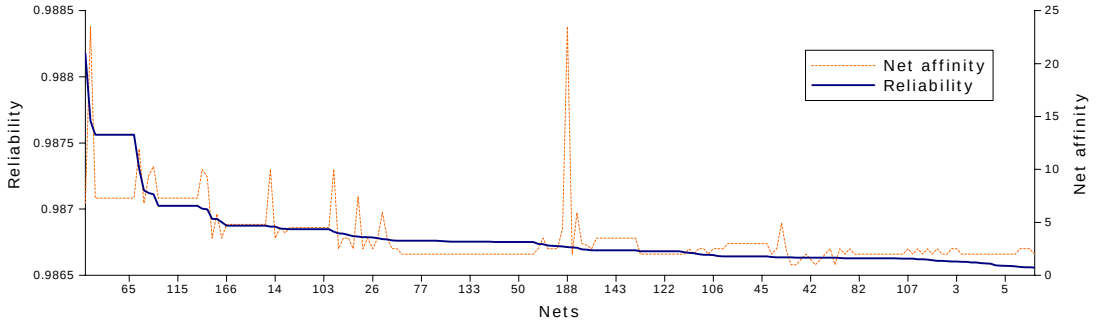


Figure 4.10: Reliability versus chosen net to be hardened.

The need to use some sort of relation or cost function is highlighted in Fig. 4.10, in which it is evidenced that hardening costs and reliability improvements have non-similar trends (the Nha_i values used in the image are purposely not normalized to draw attention to the high values). For instance, it is possible to identify a terrible candidate for hardening in the middle portion of the image (around net 188). Such candidate has a remarkably high affinity and a less than average increase in reliability. On the other hand, it is possible to identify good candidates for hardening in the left portion of the image (in the plateau right before net 65). Such candidates have acceptable hardening costs and above average increases in reliability.

4.2.2 Experimental Results

The methodology previously described was applied to a set of benchmark circuits, most of them from the ISCAS'85 set. Each cell of each circuit was set with a q_i value of 0.999. The weight parameters were set as w1 = 1 and w2 = 1 for all experiments. All gates in the library were characterized as previously discussed, i.e., based solely on an area criterion.

In the first experiment, the hardening goal was set to achieve a relative increase of at least 10% in the reliability of the circuits. The results are shown in Tab. 4.7, in which the figures in bold highlight scenarios where the methodology was more effective (when compared to not using the hardening affinity). Not using the hardening affinity is the equivalent of setting w2 = 0, i.e., the cells' area is not taken into account.

Regarding the improvements in reliability, it must be emphasized that they are relative. For instance, a circuit with a reliability of 99%, after being hardened by 10%, will reach 99.1%. In other words, a relative increase of 10% in reliability is actually a decrease of 10% in the unreliability (in the given example, unreliability goes from 1% to 0.9%).

The results presented in Tab. 4.7 show that, when the hardening affinity is not used, there is an average increase of 24.45% in circuit area. On the other hand, when applying the methodology described in this section, there is a much smaller increase in area, 9.95% on average. A remarkable result is obtained for the circuit 74283, in which the area increase is diminished from 32.2% to 10.8%. Such circuit is quite small (it contains only 40 cells) and, even so, the methodology was able to find candidates that would increase the reliability effectively.

The results concerning a relative increase of 20% in the reliability are presented in Tab. 4.8. An average area increase of 39.1% is obtained when the methodology is not

Table 4.7: Results for relatively increasing the reliability by (at least) 10%.

Circuit	Area (μm^2)	No hardening affinity			With hardening affinity		
		Hardened cells	Area (μm^2)	Area increase	Hardened cells	Area (μm^2)	Area increase
c17	33.1	3	66.3	100%	1	44.2	33.3%
74283	306.5	6	405.2	32.2%	3	339.7	10.8%
c432	1134.4	4	1209.6	6.6%	4	1209.6	6.6%
c499	2155.1	26	2579.6	19.6%	15	2407.1	11.6%
c1355	3194.7	43	3872.1	21.2%	24	3460.1	8.3%
c1908	5273.7	48	6186.7	17.3%	35	5660.8	7.3%
c3540	10855.2	61	11688.3	7.6%	30	11240.4	3.5%
c2670	8018.0	38	8602.4	7.2%	28	8419.9	5.0%
c5315	15293.6	85	16583.8	8.4%	43	15794.9	3.2%

used. However, when applying the methodology there is a much smaller increase in area, 17.4% on average. A remarkable result is obtained for the circuit c1355, in which the area increase is diminished from 45.3% to only 16.6%.

It is noteworthy that the reliability improvement percentages shown in tables 4.7 and 4.8 are not large ones. Nevertheless, it must be emphasized that they are adequate for a scenario in which there is a (very) thin hardening budget. Additionally, there is a certain point where selective hardening is no longer feasible, and a global redundancy solution such as TMR becomes more interesting. This is clearly represented in Fig. 4.11, which plots the area increase versus reliability improvement trend for the circuit 74283. For instance, it is possible to improve the reliability relatively by 30% by paying a 25% increase in area. Yet, when the reliability improvement target is larger than 70%, the required circuit area more than doubles.

In Tab. 4.9, it is shown, for both experiments, both the execution time and number of hardened nets. The execution times were measured in a QuadCore CPU, running at 2.40GHz. The number of hardened nets more than doubles (on average) from the first experiment to the second. It is also possible to notice that the execution time is linearly proportional to the number of hardened nets.

Some of the execution time values shown in Tab. 4.9 can be considered high. Because of that, different ways to perform the analysis have been researched. The profile of the C_i function was studied and heuristics for reducing the execution time have been proposed. Section 4.3 details those heuristics.

4.2.3 Mixing Global TMR and Selective Hardening: a Methodology for Mitigating Single and Multiple Faults

One particular and recent issue with SEEs is the occurrence of multiple faults induced by a single particle strike. Such faults, although less common than single faults, might overcome current fault tolerance techniques such as TMR. Thus, in this section, a circuit-level hardening methodology is proposed. This hardening methodology is able to properly mitigate single faults and it is also able to partially mitigate multiple faults. The former

Table 4.8: Results for relatively increasing the reliability by (at least) 20%.

Circuit	Area (μm^2)	No hardening affinity			With hardening affinity		
		Hardened cells	Area (μm^2)	Area increase	Hardened cells	Area (μm^2)	Area increase
c17	33.1	3	66.3	100%	1	44.2	33.3%
74283	306.5	12	490.0	59.8%	5	361.8	18.0%
c432	1134.4	15	1416.8	24.8%	16	1347.9	18.8%
c499	2155.1	52	3003.9	39.3%	25	2575.0	19.5%
c1355	3194.7	102	4644.4	45.3%	48	3725.6	16.6%
c1908	5273.7	105	6879.7	30.4%	71	6058.9	14.9%
c3540	10855.2	147	12875.3	18.6%	101	12182.3	12.2%
c2670	8018.0	86	9328.6	16.3%	72	9088.9	13.3%
c5315	15293.6	175	17924.0	17.1%	126	16893.7	10.4%

is handled by global TMR, while the latter is handled by selectively hardening gates that are critical to the circuit reliability.

The selective hardening approach applied here is based on the net hardening concept presented in this section. Some issues arise when combining both techniques, which requires a proper modelling of the problem, to be discussed further in the text.

When it comes to circuit hardening, TMR has been frequently used as a generic fault tolerance technique. The concept is to simply place three copies of the same circuit or system operating in parallel and to vote the outputs to eliminate possible discrepancies. As long as only 1-out-of-3 copies is faulty, the final output will remain correct. This is the same approach adopted in the proposed methodology to mitigate single faults.

4.2.3.1 Scenario

As previously mentioned, single faults are still more common than multiple faults. Considering a scenario where hardening against both is required, our approach was to first apply global TMR to the whole circuit in order to tackle all single faults. After that, it was necessary to choose a hardening strategy for multiple faults.

Higher order NMR (N-Modular Redundancy) could be used as well, but the costs associated with this strategy could be overwhelming: using 4MR is not exactly practical because majority cannot be decided in the case of two faulty modules. On the other hand, 5MR is practical, but it imposes a tremendous effort to be applied, since both area and power are roughly multiplied by five. The solution applied in this work is to use selective hardening and perform local TMR in the copies of the global TMR, as represented in Fig. 4.12.

Since all the primary outputs of the circuit are already connected to voters, the hardening methodology will always be able to cope with one faulty module. Thus, when performing local hardening, it is done in only 2-out-of-3 modules. By doing so, some of the hardening budget is saved. Whenever a gate is selected for further hardening by local TMR, 3 copies of it are already present in the circuit, one in each module. After local TMR hardening, 2-out-of-3 modules will be locally tripled, raising the number of copies

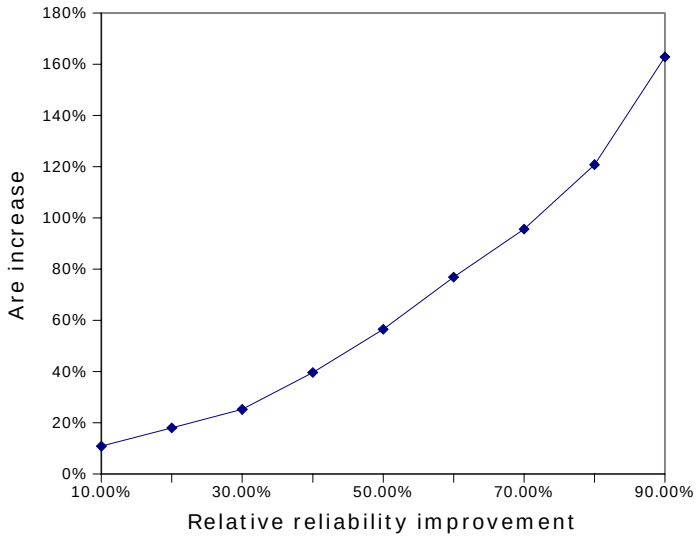


Figure 4.11: Area increase versus reliability improvement for the 74283 circuit.

of that same gate to 7 (3 in the first module, 3 in the second module and 1 in the third module).

Also, for simplicity reasons, the analysis here depicted does not take into account the area cost of voters, neither the local nor the global ones. This is not an issue since the technique being applied locally does not need to be TMR. Other techniques can be used as well without changing the way critical gates are chosen.

4.2.3.2 Modelling

Proper modelling of the circuit is required when global TMR is applied. First, let us assume a non hardened version of the c17 circuit, as presented in the left hand side of Fig. 4.13. If SPR analysis is performed as previously described and all gates are ranked accordingly, a certain order in which the gates (or nets) should be selected for hardening is obtained. Nevertheless, if the same analysis is performed for the same circuit that was already TMR'd, this order might not be the same (see the right hand side of Fig. 4.13).

This further complicates matters for simulation based analysis, since it suggests that performing analysis using a single copy of the circuit does not lead to optimal results. Therefore, the full TMR'd version of the circuit should be simulated, which increases the number of fault sites by 3 (assuming fault-free voters). The same hardening ranking analysis was done for the circuit 74283 and it is graphically depicted in Fig. 4.14. Both lines drawn in Fig. 4.14 were obtained from evaluating the hardening cost function, before and after global TMR. In every occasion that the black dotted line is under the red solid line, more than optimal (thus, incorrect) decisions are made. When the opposite happens, suboptimal decisions are made.

It is important to notice that the analysis takes into account the logical masking capabilities of the logic gates. And these capabilities do not change, either if the circuit is tripled or not. What actually happens (and explains the order changes in figures 4.13 and 4.14) is that the voters placed by global TMR may actually mask the same faults (or pattern of faults) that a given gate already masks. In relative terms, it is then possible to

Table 4.9: Comparison between the execution time and number of nets hardened in both scenarios: relative increases of 10% and 20% in circuit reliability.

Circuit	A 10% increase in reliability		A 20% increase in reliability	
	Hardened nets	Execution time (s)	Hardened nets	Execution time (s)
c17	1	0.2	1	0.15
74283	2	0.6	3	0.9
c432	1	11.2	7	80.1
c499	3	8.4	5	9.8
c1355	12	106.3	24	224.9
c1908	18	729.0	41	1840.8
c3540	13	2170.4	61	11764.3
c2670	17	805.0	39	2116.8
c5315	18	4767.7	56	17116.5

say that gates 'lose' some masking capability since they start to share that same capability with the global voters. It is even more precise to state that some gates become less important from a masking point of view.

In the particular analysis done in Fig. 4.14, 8 out of 40 cells are evaluated out of order, which represents an error rate of 20%. In other words, one out of five times a cell is picked for hardening, an erroneous decision is being made. In a scenario where the hardening budget is tight, this 20% error margin might be representative. This margin might be even more impacting if consecutive erroneous decisions are taken.

4.2.3.3 Results

The experiment's goal was to harden the circuits as much as possible, but always limiting the increase in area. In other words, the idea is to improve the circuit reliability as much as possible within an area-based hardening budget. The comparison basis is a hypothetical 4MR version of the circuit, i.e., selective hardening is performed until the circuit reaches the same size as a 4MR hardened version (roughly a 300% increase in area). The assumed hypothesis is that 4 times the circuit area is a good commitment between area increase and gains in reliability.

Most of the circuits chosen are benchmarks from the ISCAS'85 set. The details regarding such circuits are given in Tab. 4.10. The area values were obtained by summing the area of all gates as if the circuit had been synthesized using a 90nm library [109]. No placement utilization factor or routing is taken into account.

Results are summarized in Tab. 4.11, in which the values in the columns entitled 'area increase' are proportional to the original circuit's area; while the columns entitled 'SPR' contain the output of the algorithm, i.e., the probability that all outputs of the given circuit will be correct at the same time.

Some of the hardened circuit versions are actually bigger than a 4MR version (c17with 333% and 74283 with 303%), but that is due to the small size of the circuits. Although not shown in Tab. 4.11, the execution time of the analysis is not long. For instance, the analysis of the circuit c499 takes only □ 80 seconds in a modern computer. Even the

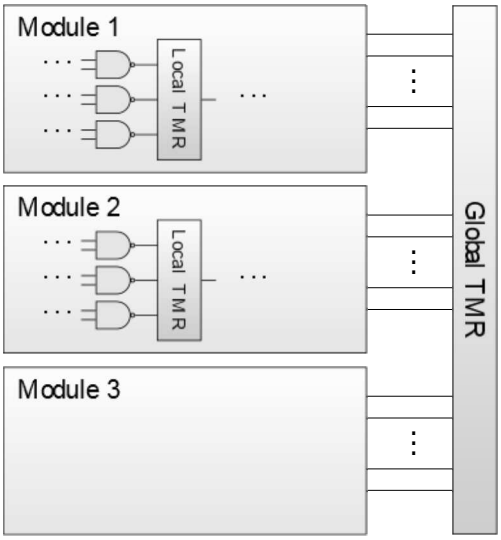


Figure 4.12: Local and global TMR schemes.

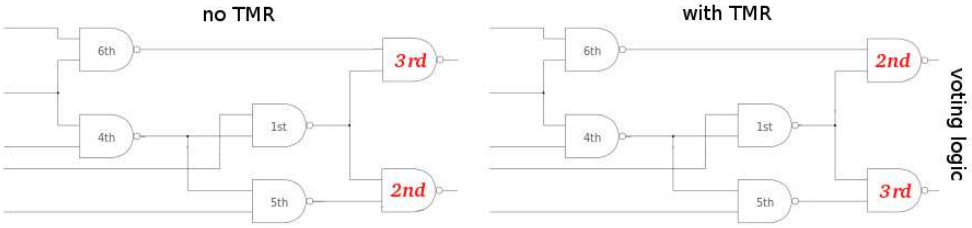


Figure 4.13: Order in which selective hardening should be applied to the circuit c17.

biggest of the circuits, c3540, can be analyzed in a few hours. In Section 4.3, heuristics that are able to reduce the execution time even further were proposed. These heuristics exploit circuit regularity.

In order to better understand the decreases in circuit susceptibility to SETs, Fig. 4.15 contains a plot of both SPR columns from Tab. 4.11, as well as a plot of the circuit reliability before any hardening technique is applied. The SPR reliability results are inverted, i.e., the results plotted are given by 1 □ R. This value is referred as the circuit susceptibility to SETs. The c1355 circuit presents a remarkable result: a reduction of 79.9% in the susceptibility with an relative increase of 33% in area.

Two main conclusions can be taken from the analysis of Fig. 4.15. The first is that hardening against single faults by using global TMR will provide a substantial increase in the circuit reliability. The vertical axis of Fig. 4.15 is plotted in a logarithmic scale, thus for some circuits we see a reduction in the SET susceptibility that is higher than 2 orders of magnitude. The second conclusion is that further reductions are possible, but the costs become large very quickly while the reliability gains increase at a lower pace. For instance, if we take circuit c2670, an decrease in susceptibility in the order of 21.7% is seen. That decrease comes with an increase of 33% in area.

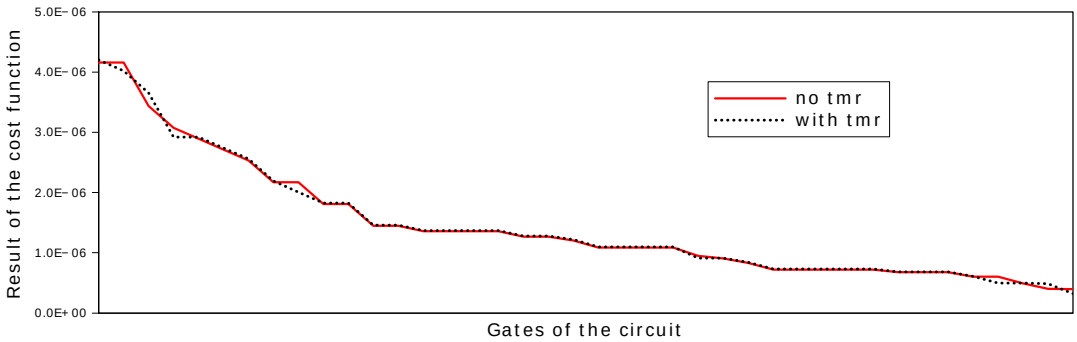


Figure 4.14: Graphical analysis of the differences between selective hardening in simple and tripled versions of the 74283 circuit.

Table 4.10: Characteristics of the case-studied circuits.

Circuit	Area (μm^2)	Number of gates	Reliability
c17	33.1	6	0.999438
74283	306.5	40	0.996152
c432	1134.4	160	0.986534
c499	2155.1	202	0.986389
c1355	3194.7	546	0.978095
c1908	5273.7	880	0.968332
c2670	8018.0	1269	0.935985
c3540	10855.1	1669	0.937365

4.3 Profiling of the Hardening Cost Function

As seen in the previous section, the cost function was used together with a constant improvement target T for the reliability of a given circuit (e.g., improve the reliability relatively by T = 10%). The designer/ user is responsible for choosing the value of T given the project constraints. What is going to be discussed in this section are ways to eliminate such constant target, i.e., automatically determine through the use of heuristics when selective hardening is no longer desired or feasible.

In Fig. 4.8, it is shown how the achieved reliability value is compared against a user-given reliability target T. If it is lower than the target, the algorithm starts again and all gates still not hardened are considered as candidates. Otherwise, if the target is met, the algorithm ends and outputs the ordered list of gates to be hardened. In this section it is assumed that the same flow is still used, only the target is no longer given by the user.

All the other characteristics of the analysis flow remain the same: SPR is is still used together with the cost values shown in Tab. 4.6. Classical TMR is still considered as the technique being used for hardening. Voter cost is neglected while for the other gates only the increases in area are taken into account.

The methodology described in Section 4.2 was applied to several ISCAS'85 benchmark circuits with a reliability target T = 100%. The profile of the cost function was then obtained for circuits of different sizes and topologies. Figures 4.16 and 4.17 illustrate the

Table 4.11: Results for decreasing the circuit susceptibility to multiple faults.

Circuit	TMR		TMR and Selective Hardening			
	Area (μm^2)	SPR	Additionally hardened cells	Area (μm^2)	Area increase	SPR
c17	99.5	0.999999	2	143.7	333%	0.999999
74283	919.5	0.999990	14	1236.5	303%	0.999997
c432	3403.4	0.999908	45	4545.3	300%	0.999962
c499	6465.5	0.999982	58	8622.9	300%	0.999990
c1355	9584.1	0.999954	145	12791.3	300%	0.999990
c1908	15821.2	0.999785	237	21107.5	300%	0.999903
c2670	24054.1	0.999387	334	32075.7	300%	0.999520
c3540	32565.5	0.998880	436	43447.8	300%	0.999403

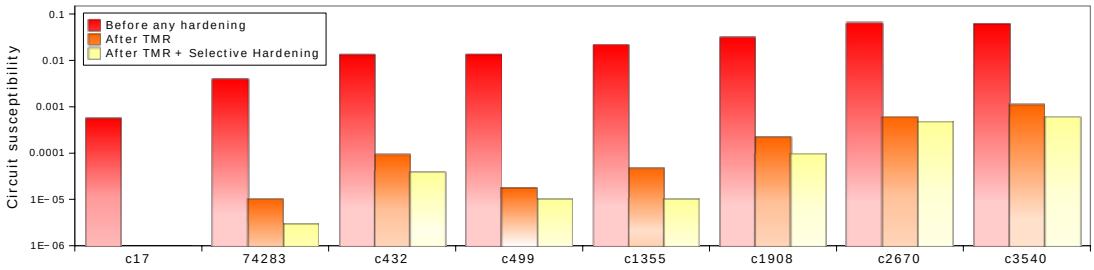


Figure 4.15: Susceptibility comparison between the unhardened and two hardened versions of the same circuits.

cost function profile for the circuits c432 (a channel interrupt controller) and c499 (32-bit single-error-correcting circuit). These circuits were chosen particularly because they present two very contrastive profiles that are of interest.

Concerning figures 4.16 and 4.17, it must be highlighted that the higher the value of the cost function, the best candidate for hardening the net is. The nets in the x axis are ordered according to the result of the cost function.

The illustrations in both figures were obtained using the parameters $q_i = 0{:}999$ and $q_{ij} = 1$. Other combination of values cause slight changes in the plots, i.e., the profile of the function remains the same. In other words, the profile of the function is highly related to the logic masking capabilities and the affinity of each gate. The closer a gate is to the y axis, the better candidate for hardening it is.

The illustration in Fig. 4.16 presents a profile that contains a fast drop in the function, observed in the very first gates. Circuits that have some degree of regularity (e.g., adders and multipliers) have a profile with some similarities with the one in Fig. 4.17, where a 'step-like' pattern is observed. Each 'step' or plateau represents a set of gates that has a similar functionality in the circuit, therefore they can be hardened in any given order. Taking into account both profiles that were presented, two heuristics have been defined in order to decide when selective hardening starts to impose an impractical cost. Those heuristics are explained in details in the next sections.

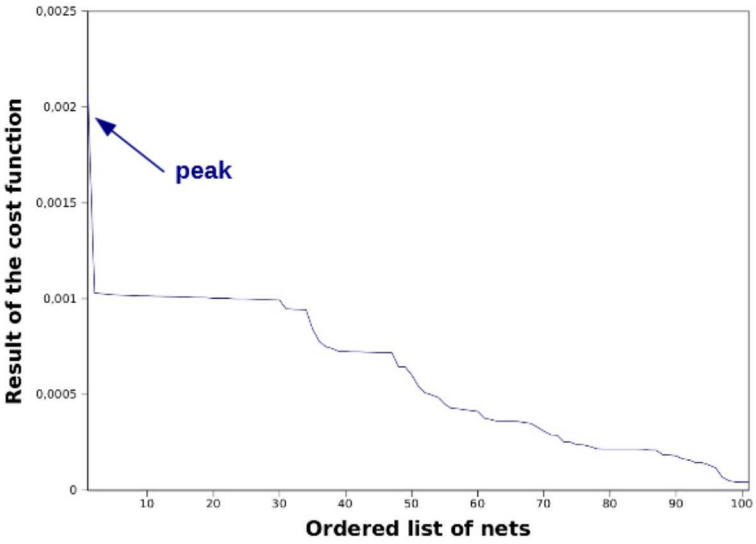


Figure 4.16: Cost function profile for the circuit c432.

4.3.1 Sum of Elements Heuristic

This heuristic was defined to create a stop point when the sum of the C_i terms from the elements that were already hardened reaches a threshold. Let C_0 be the value of the cost function for the best hardening candidate. Then the target becomes to find a value j such that:

$$\sum_{i=2}^{j} C_i \geq K \cdot C_0 \tag{4.9}$$

K is an empirically chosen constant. In other words, the threshold is defined as K times the value of the cost function for the first hardened gate. This heuristic can be interpreted as an integral that sums the area under a curve. For the sake of comparison, in the results that follow, the parameter K was empirically set as $K = 10$.

4.3.2 Percent Wise Heuristic

This heuristic was defined to create a stop point at the first C_i value that is lower than X % of the first term (C_0). This heuristic can be interpreted as an horizontal threshold value. When the function crosses that threshold it is no longer feasible to perform selective hardening for the remaining gates.

For the sake of comparison, in the results that follow, the parameter X was empirically set as $X = 50\%$. In other words, any gate that improves the circuit reliability with a C_i value that is less than half of C_0 should not be hardened, i.e., hardening is only applied to those cells that are at least half as effective as the first candidate.

4.3.3 Comparing the Heuristics

Both heuristics were applied to the circuit c1355 (which is also a 32-bit single-error-correcting circuit). Figure 4.18 contains the plot of the cost function for all elements of

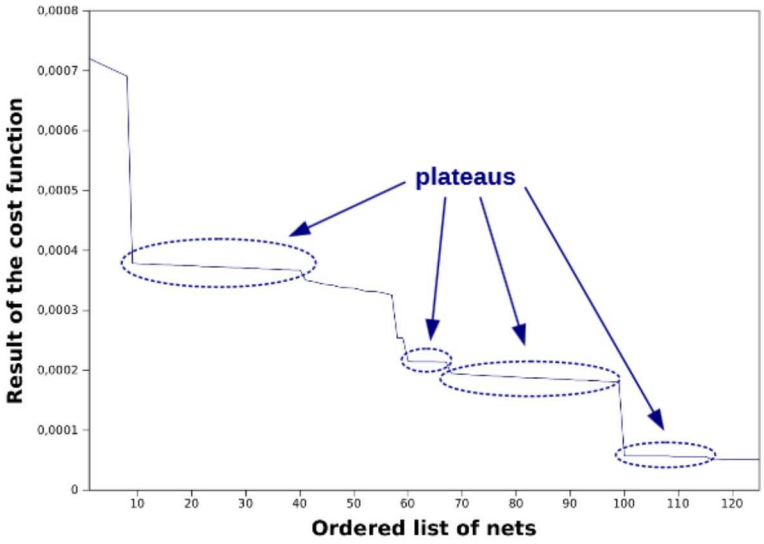


Figure 4.17: Cost function profile for the circuit c499.

the target circuit. The dashed vertical lines represent the points where the heuristics decided that selective hardening was no longer feasible.

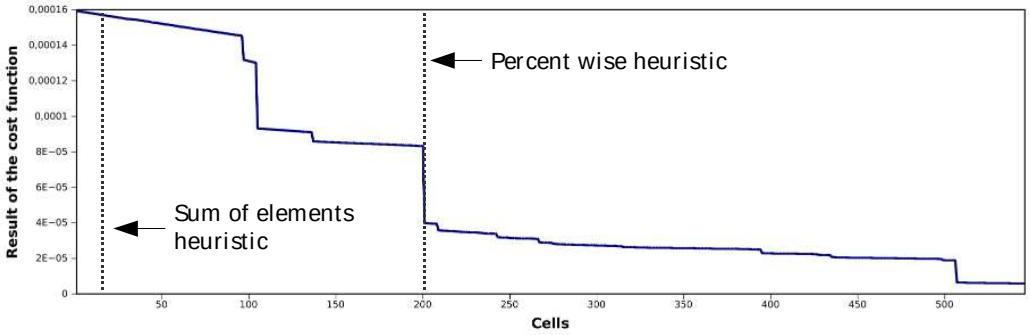


Figure 4.18: Both heuristics applied to the circuit c1355.

Deciding which parameter value is more appropriate for each circuit is a complex task. For instance, for the circuit c1355, the first heuristic would select 11 gates for hardening, while the second heuristic would select 201 gates. Hardening 201 out of 546 gates (around 36%) might be a hard assignment, since most of the times the area budget will not allow for such hardening (the total circuit area would become 76% larger).

Nevertheless, selecting 11 out of 546 gates (around 2%) might be a better and more suitable choice. Along the same lines, applying the percent wise heuristic to the circuit c432 would result in only 2 gates being selected for hardening, which could left some of the hardening budget unused.

In the next section we present the results for other circuits and we also extend the discussion regarding which heuristic (and associated parameter) is more appropriate for which scenario.

4.3.4 Experimental Results

The methodology described in Section 4.1.2 was applied to several ISCAS benchmark circuits. Each gate from each circuit was set using $q_i = 0{:}9999$. The results are presented in tables 4.12 and 4.13. The former table contains the results for the first heuristic defined in Section 4.3.1 (with $K = 10$) while the latter contains the results for the second heuristic defined in Section 4.3.2 (with $X = 50\%$).

Table 4.12: Results for the sum of elements heuristic, $K = 10$.

Circuit	Number of gates	Original area	Hardened gates	Hardened area	Area increase
c17	6	33.1776	6	99.5328	200%
74283	40	306.5096	20	547.9688	78.7%
c432	160	1134.4672	33	1541.4208	35.8%
c499	202	2155.1680	12	2414.1504	12.0%
c1355	546	3194.7328	11	3316.3840	3.8%
c1908	880	5273.7488	13	5417.5184	2.7%
c2670	1269	8018.0632	19	8233.7176	2.6%
c3540	1669	10855.1824	25	11177.7424	2.9%
c5315	2307	15293.5992	20	15518.4696	1.4%

Table 4.13: Results for the percent wise heuristic, $X = 50\%$.

Circuit	Number of gates	Original area	Hardened gates	Hardened area	Area increase
c17	6	33.1776	5	88.4736	166.6%
74283	40	306.5096	9	406.0424	32.5%
c432	160	1134.4672	2	1187.5264	4.6%
c499	202	2155.1680	41	2854.6752	32.4%
c1355	546	3194.7328	201	5647.1232	76.7%
c1908	880	5273.7488	119	6611.912	25.3%
c2670	1269	8018.0632	10	8128.6552	1.4%
c3540	1669	10855.1824	8	10963.9312	1.2%
c5315	2307	15293.5992	15	15459.4872	1.1%

In tables 4.12 and 4.13, the meaning of each column is as follows: the column denoted "Original area" contains the sum of the area from each gate in each circuit (therefore placement utilization rate and routing overhead are not considered). The column denoted "Hardened gates" contains the amount of gates that are selected for hardening. Then, the column denoted "Hardened area" contains the circuit area of from the hardened version of the circuit, while the column denoted "Area increase" contains that same value but percent wise. All the area values are given in $\square m^2$.

An analysis of the area increase values in Tab. 4.12 reveals that the sum of elements heuristic is not prone for small circuits, causing a large overhead for the circuits 74283 and c432. For the smallest of the circuits (c17) the heuristic decides that all gates should be hardened, which is unacceptable when the goal is selective hardening. Nevertheless, this can be avoided by using a smaller value for the parameter K (e.g., K = 1 elects 2 cells while K = 2 elects 4 cells for hardening). This is not the case for the area increase values in Tab. 4.13. There is no value for the parameter X that will be a good fit for all circuits or even for a group of circuits. Therefore, it is quite harder to apply the percent wise heuristic.

4.3.5 Comparison with Related Works

A straightforward comparison with other methodologies is not simple since the hardening goals are usually different. If comparing a methodology is hard, it is even harder to compare the heuristics proposed on top of a methodology.

A simple solution adopted by related works is to define a limit or target for hardening. In [93] a simple limit L is defined as the maximum number of gates to be hardened. In both [85] and [115], a hardening limit in terms of area increase is applied. As shown in Section 4.1 and in [86], a hardening target was defined as a relative improvement in the reliability of the circuit. Nevertheless, none of the mentioned works perform an evaluation of how hard it is to reach a hardening limit or target. This is the reason why the profile of the cost function was studied.

4.3.6 Optimizations

Generating the plots of the cost function requires a long computation time. The issue is that every time a net is selected as the best candidate, the order must be re-evaluated since shifts in the selection order are possible and often seen. In order to tackle this issue, possible ways to optimize the computation were studied. These approaches were published in [116–118]

If we take SPR alone, it has a linear complexity (O(n)) with respect to the number of gates, which is a very positive property of the algorithm. Nevertheless, when it is applied as described in Section 4.2, the execution time also becomes proportional to the number of nets in the circuit ($O(n^2)$, roughly assuming the number of gates and nets is the same, given by n). And since there is a need to re-evaluate all nets once a net is selected, its complexity then becomes bounded by $O(n^3)$.

In order to reduce the execution time, two approaches are proposed: the first approach is an optimization based on the analysis of the regularity of the circuit and it is described in the paragraphs that follow. The second approach is to limit the scope of the analysis to the first elements of the cost function, which are the ones that are in fact interesting for selective hardening. This scope limitation is given by the heuristic described in Section 4.3.1.

Both analysis depicted in figures 4.16 and 4.17 contain some plateaus, i.e., some areas in which the cost function has a linear decreasing trend. This apparent linearity happens because all the nets that are part of the same plateau are equivalent. By equivalent it is meant that those nets contribute equally to the overall circuit reliability as well as presenting the same hardening cost. As a matter of fact, the order in which they are selected for hardening is not relevant. Thus, we work under the fair assumption that those nets

have a similar purpose in the circuit and therefore represent a certain degree of regularity of the given circuit.

From this assumption, once one of these plateaus has been detected, there is no need to proceed with the analysis of all nets in it. It is possible to save some execution time by estimating the profile of the plateau as a whole. Given the number of nets in a plateau is known, as well as the value of the cost function for the first element outside (after) the plateau, it is possible to plot the corresponding profile. After the first round of analysis is done, all nets have been submitted to SPR analysis, one at a time. The results from the first round can be used to find the plateaus.

This optimization was applied to some of the ISCAS'85 circuits. The results are given in Tab. 4.14, in which it is possible to see that some circuits require a long execution time, even with this optimization. In particular, the analysis of the circuit c1908 has an execution time of more than 9 hours. As predicted by Fig. 4.17, the analysis of the circuit c499 is much faster when the optimization is applied (several plateaus).

Table 4.14: Execution time for determining the cost function profile with a target of 100%.

Circuit	Exec. time (s)	Exec. time with optimization (s)
c17	0.26	0.23
74283	4.78	4.48
c432	1058.71	611.76
c499	263.75	27.33
c1355	5907.91	709.14
c1908	56621.25	33898.49

It can be seen in Tab. 4.14 that the optimization has achieved reductions in the computation time of up to 89:6% for the case studied circuits.

Since the first optimization that was proposed is still not able to cope with large circuits, it was combined with a second one. This heuristic was defined to create a stop point when the sum of the elements already evaluated reaches a threshold, as explained in Section 4.3.1.

Using a K value of 10, the same analysis as before was performed to the largest circuits in the ISCAS'85 set. The results are given in Tab. 4.15. Notice that the column entitled 'Exec. time (h)' is given in hours and, when a circuit required more than a day to be analyzed, it was assumed that its analysis is completely unfeasible and the execution of the algorithm was canceled. The values in the column showing the execution time after optimizations are given in seconds.

The second proposed approach has obtained even more substantial results, reducing the computation time from the order of days to minutes for some of the circuits. This can be clearly seen in Tab. 4.15. Unfortunately, the amount of execution time that is saved changes from circuit to circuit since it is related to the architecture of each circuit. Being so, future research efforts should try to find ways to minimize this analysis by different techniques. For instance, the multiple SPR executions could be done in parallel.

Table 4.15: Execution times for determining the partial cost function profile.

Circuit	Exec. time (h)	Exec. time with optimization (s)
c1355	1.64	15.06
c1908	> 19	477.49
c3540	> 24	2109.85
c2670	> 24	523.56
c5315	> 24	2555.74

4.4 Single Event Transient Mitigation Through Pulse Quenching: Effectiveness at Circuit Level

It has been acknowledged that multiple transistors can collect charge from a single ion hit in a process referred as charge sharing [18]. As a matter of fact, these multiples SETs might occur between nodes that are not electrically related, potentially increasing the circuit error rate. Many studies have been performed with the sole goal of characterizing the charge sharing profile [18, 112, 114]. Factors such as particle's energy, angle of incidence and the type of the device that was struck (i.e., NMOS or PMOS) should be considered.

Furthermore, when the affected nodes are indeed electrically related, a secondary mechanism may take place, in which the induced transient pulses might be reduced or quenched. Thus, the term PQ. The work of Ahlbin et al. [27] has described PQ thoroughly and shows how it can reduce the sensitive area of the circuit.

However, the analysis in [27] has been made at gate-level. No actual circuit-level analysis was performed. Thus, the goal of this chapter is to extend such analysis to larger circuits and to evaluate if the mechanism still plays an important role in error rate reduction at circuit-level. Also, a secondary analysis is performed using the layout technique described by Atkinson et al. in [119], which intentionally promotes PQ by introducing additional circuit area.

The following section discusses the foundations of charge sharing and quenching mechanisms. A detailed description of the circuit-level analysis is given in Section 4.4.2 while some results are presented in Section 4.4.3.

4.4.1 Background: Single Event Transients, Charge Sharing and Pulse Quenching

When a particle strikes a microelectronic device, the most sensitive regions are usually reverse-biased p/ n junctions. The high field present in a reverse-biased junction depletion region can very efficiently collect the particle-induced charge through drift processes, leading to a transient current at the junction contact [18]. SETs are usually characterized by the width of such generated transient current.

While the size of the ion track generated by an incident ion on a silicon surface remains relatively constant, the distance between adjacent devices has been significantly reduced with technology scaling. In fact, multiple transients due to a single ion hit (i.e., due to charge sharing) have been measured for currently in use technology nodes such as 90nm [113] and 65nm [120].

Charge sharing is a big concern because it has the potential of making hardening

techniques ineffective, thus many works have aimed at reducing charge sharing. For instance, Black et al. [121] made use of guard contacts to reduce charge sharing in PMOS devices. Other works try to explore/ promote the charge sharing mechanism to reduce error rates. For instance, Entrena et. al [122] identified pairs of cells that, if struck at the same time, would produce transients that would be masked, i.e., would cancel each other. In other words, such work promotes charge sharing between cells that, when struck, will have their transients logically masked.

Analogously, PQ can be used to promote charge sharing and reduce error rates. Nevertheless, it has a different behavior since it is not linked to logical masking. Due to a dynamic interplay of charges, SETs can be masked "electrically" if two adjacent transistors have similar time constants for (delayed) charge sharing and actual signal propagation. The concurrency of these two effects may cause shorter than expected transients, thus partially masking the transient, as shown in Fig. 4.19. The effect is prominent in inverter-like structures (actual inverters or larger cells that have an inverting stage). Details concerning the delayed charge collection mechanism are explained in [27].

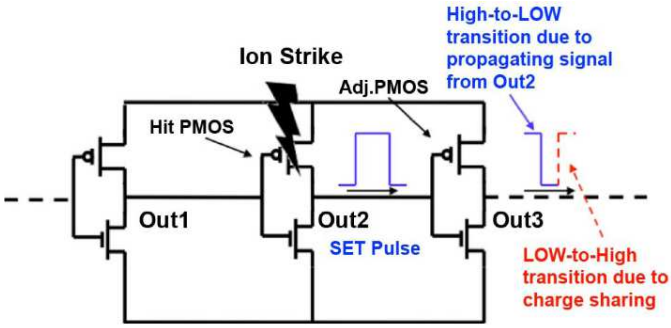


Figure 4.19: A schematic of a chain of three inverters illustrating the change in SET pulsewidth as it propagates [27].

4.4.2 Methodology and Error Rate Analysis

Since the PQ effect is prominent in inverter-like structures, our analysis begins by identifying such structures. First, it should be mentioned that several gates already have an internal inverter stage (ORs and ANDs, for example). Such gates already have the potential of quenching pulses, i.e., they benefit from intra-cell PQ.

Nevertheless, some cells that do not have that potential can benefit from inter-cell PQ, which can be achieved by a rearrangement of the circuit layout. Some pair of cells might be brought together during placement to make them effectively quench pulses. In this work, the interest is focused on those pairs of cells that promote inter-cell PQ. The first cell in the pair is termed the primary struck cell while the other one is termed secondary cell. Those cells must match the following criteria to be considered a feasible pair in the analysis:

- The primary struck cell must have (at least) one 'exploitable' output. The drain region of the PMOS transistor connected to that output must be near the cell's boundary. Thus, charge sharing with a neighboring cell can be exploited.

- The secondary cell must have one input that is connected to an inverter-like structure.

- The secondary cell's input node (i.e., the drain region to which the input is connected) must also be near the cell's boundary region.

Considering the required criteria, an analysis of the gates of a 90nm standard cell library [109] was made. The goal of the analysis is to identify which cells are candidates for being primary struck and/ or secondary cells. Figure 4.20 depicts a layout containing two cells from the referred library: a NOR2 on the right-hand side and an inverter on the left-hand side.

The NOR2 cell in Fig. 4.20 is a good candidate for being a primary struck cell since it matches all the requirements previously mentioned. The portion of the image highlighted in yellow corresponds to the sensitive drain area that has the potential of sharing charge with a neighboring cell. Bringing the inverter close to the NOR2 cell thus effectively promotes PQ (it is assumed that both cells are connected through routing in a higher layer metal, which is not represented in the image).

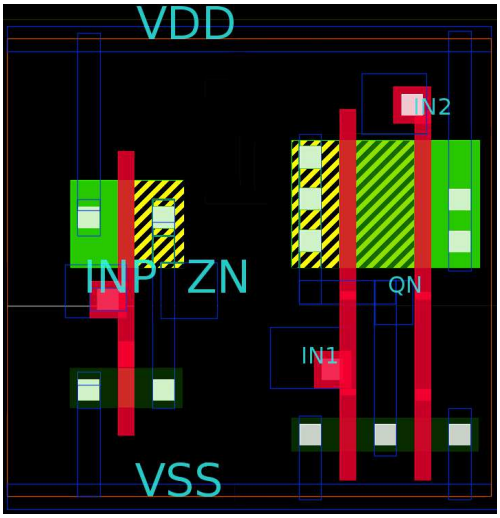


Figure 4.20: Layout of an inverter and a NOR2 cell from a 90nm ASIC library.

After a full analysis of the library, only 16 cells that can act as the primary struck cell were identified. Only two cells can act as the secondary cell (inverters and buffers). Obviously, since all those cells implement different logic functions and have different sizes (i.e., driving strengths), pulses are not quenched with the same efficiency. Being so, average quenching efficiency factors were defined for all concerned cells. Some of these factors are given in Tab. 4.16.

The factors defined in Tab. 4.16 are intentionally overestimated. For instance, let us take the NOR2X1 cell, which has 2 PMOS transistors (as depicted in Fig. 4.20). It is considered that the drain region associated to the leftmost transistor can always share charge with a neighboring cell, while the region in the center of the image can share charge 50% of the time. Thus, the reduction in sensitive area is of 50%. Yet, that value does not take into account the input patterns of the cell. Analogously, cell AND3X1 has 4 PMOS transistors, from which 2 are considered able to share charge. Yet, not all drain regions are

Table 4.16: Average reduction in sensitive area due to pulse quenching.

Cell	Factor	PMOS transistors
INVX0	100%	1
NOR2X1	50%	2
AND3X1	30%	4
XNOR2X1	16.6%	7

Table 4.17: Number of pairs of cells that can be used for promoting PQ.

Circuit	Number of gates	Inter-cell pairs	Unpaired candidates
c432	160	3	42
c499	202	0	98
c1355	546	0	130
c1908	880	214	289
c2670	1269	309	694
c3540	1669	279	1041
c5315	2307	461	1365
c6288	2416	31	257

equally sized in the layout of that cell. Once again, we assumed all nodes to be equally important, which is also a source of overestimation. For a detailed discussion regarding the reductions in sensitive area, the reader is referred to [119].

It was also assumed that a cell can be flipped whenever necessary to meet the criteria previously defined. Cell flipping can and usually will add some additional wiring due to less optimal routing.

The circuits from the ISCAS85 benchmark suite [110] were chosen as case studies. After an analysis of each circuit's topology, two possible scenarios were identified and are shown in Tab. 4.17. An inter-cell pair is a pair similar to the one depicted in Fig. 4.20. The unpaired candidates are cells that fit the profile of a primary strike cell but are not electrically connected to any inverter-like structure, thus PQ cannot be promoted by pairing (at least not directly).

Nevertheless, a hardening by design technique proposed in [119] can be used to make unpaired candidate cells more robust to SEEs. This technique comes with a considerable cost in area, while bringing together pairs of cells that are already supposed to be connected is of minimal cost, if any, since many of those pairs will be already side by side after placement.

From Tab. 4.17 it is already possible to conclude that many circuits will not present a significant reduction in error rate. This and other factors are explored in the next section when results are presented.

4.4.3 Results

In order to evaluate how the reduction in sensitive area translates into reduction in error rate, the SPR method was used. The original purpose of the method is to calculate circuit reliability by taking into account logical masking. One positive aspect of using such method is that it is not simulation-based (i.e., it is analytical), thus all possible input scenarios can be taken into account in a linear execution time.

A modified version of SPR was used for the analysis here reported. First, as previously explained, each gate is characterized by a q value in SPR modelling. This value determines how reliable each cell is, in the [0, 1] range (where 1 means the cell does not produce faults and zero means it always produces faults). For each gate g in the circuit, one SPR run is performed in which that gate (and that gate only) is set with q = 0. All the others are set with q = 1.

Each run would then produce a reliability result R(g), between 0 and 1. Such result can be interpreted as the circuit error rate due to an error in gate g. This result is averaged by taking into account all possible input scenarios, which justifies the need to use SPR instead of fault simulation. Such effort is only possible due to the performance obtained by using SPR's analytical approach.

If all R(g) values are summed, for all gates, and divided by the number N of gates in the circuit, the actual (averaged) circuit error rate is obtained. Such analysis was performed for the circuit c432 and is shown in Fig. 4.21 (the black solid line shows the R(g) value per gate analysis while the dashed line shows the average value for the whole circuit).

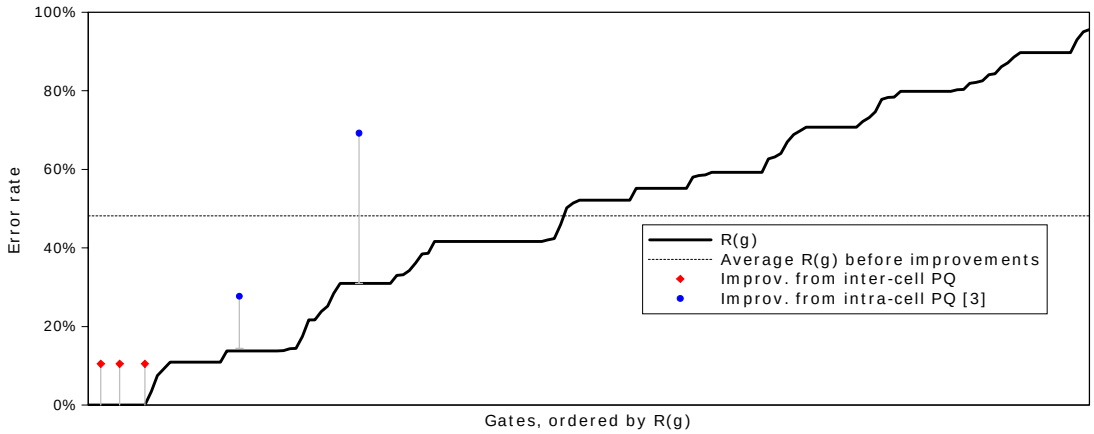


Figure 4.21: Circuit error rates for the circuit c432.

Now, let us suppose a hardening technique that promotes PQ is to be applied to a gate g. Pulse quenching can be useless in a scenario where a gate already performs a great deal of logical masking. The improvement I (g) obtained in the error rate (due to errors in that gate) is limited by the logical masking as follows:

$$I(g) \leq 1 - R(g) \tag{4.10}$$

At the circuit level, improving the sensitive area of one gate will do no good for the errors due to other gates. Thus, the whole improvement I_c at the circuit level becomes limited by:

$$I_c(g) \square (1 \square R(g))=N \quad (4.11)$$

According to Tab. 4.17, only 3 gates from the c432 circuit might benefit from inter-cell PQ, which are shown in Fig. 4.21 as red diamond shaped points. Those improvements are, nevertheless, very small, and represent an average increase of 0.19% in circuit resilience (i.e., a reduction of 0.19% in the circuit error rate).

Since the obtained improvements are very low, the intra-cell PQ technique described in [119] was also applied whenever possible. Such technique consists in adding extra inverters to the cell layout, which are connected to a gate's output. The logic function implemented in the layout is not changed, quite the opposite, it is enhanced by a secondary node that has the same role as another node.

Not all cells can benefit from that technique (the cell must have an internal inverter stage) and not all cells benefit from it in the same way. The best case scenario is the OR2 gate, which becomes symmetrical after layout modifications (one inverter on each layout extremity), as shown in Fig. 4.22. By doing so, the sensitive area of the OR2 gate becomes zero for 3-out-of-4 input patterns. The reader is referred to [119] for more details concerning the technique.

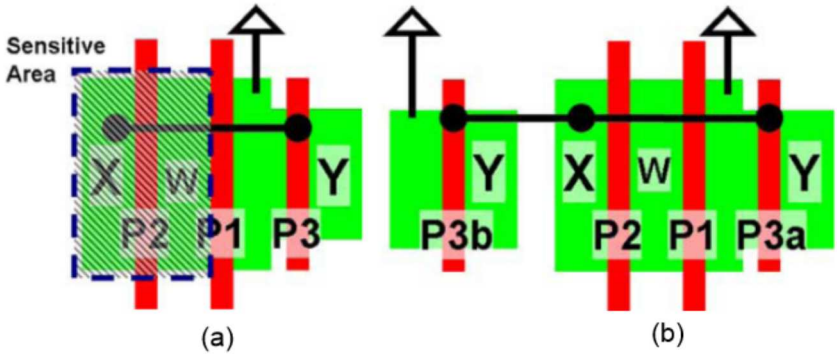


Figure 4.22: Layout of the PMOS transistor of an OR2 gate: (a) original layout and its sensitive area (b) modified layout with no sensitive area [119].

In the particular case of the circuit c432, only two gates can benefit from that technique. They are represented in Fig. 4.21 by blue circles. Once again, even when considering both techniques at the same time, the reduction in the error rate is quite small: 0.453%. The results for all the other studied circuits are given in Tab. 4.18.

Results concerning the area increase figures due to applying the layout technique proposed in [119] are shown in Fig. 4.23. The circuit c5315 has the largest area increase among the studied circuits, with an increase equivalent to 307 OR2 cells. It must be highlighted that the same circuit has only 2300 cells, thus such increase is significant. And, at the same time, the reduction in the error rate is of only 7.8%.

The results given in Tab. 4.18 clearly state that PQ cannot reduce circuit-level error rates by significant amounts. The simplifications described in Section 4.4.2 tend to overestimate the potential of PQ (e.g., is was assumed cell flipping is possible when needed), and even so the results are not expressive. There are several reasons for that:

- Logical masking plays an important role in circuit resilience. It can be meaningless to apply hardening gates to those gates that are not capable of producing errors that

Table 4.18: Error rate improvements due to inter-cell PQ and also due to inter-cell and intra-cell PQ combined.

Circuit	Inter-cell PQ	Combined with intra-cell PQ [119]
c432	0.196%	0.453%
c499	0.000%	9.932%
c1355	0.000%	1.580%
c1908	3.731%	4.199%
c2670	3.691%	7.150%
c3540	2.356%	6.211%
c5315	2.245%	7.841%
c6288	0.941%	5.368%

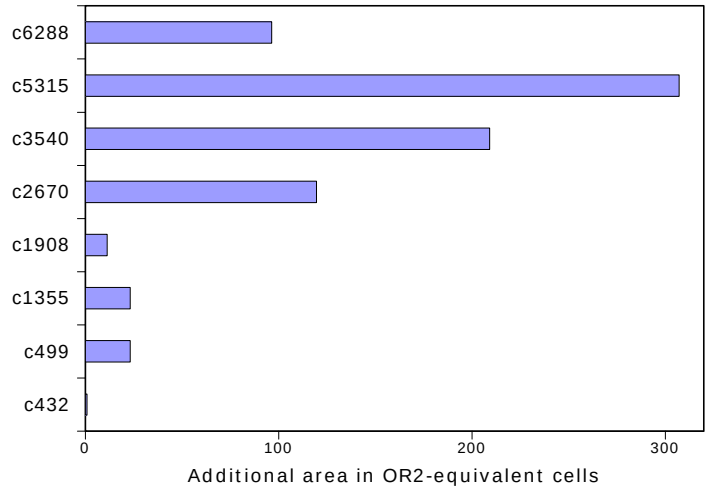


Figure 4.23: Area increase due to the layout technique presented in [119].

propagate to the circuit output(s).

- Not all circuit topologies are prone for inter-cell PQ, i.e., many circuit have a low number of suitable pairs, as highlighted in Tab. 4.17.
- Even for the gates that are paired, the error rate reduction is not 100%. This is highlighted in Fig. 4.21, specially for the paired gates (drawn as red diamonds).

Since the improvements from pairing come with almost zero cost, those are suitable for mostly all circuits and scenarios. Nevertheless, if the technique described in [119] is also to be applied, is must be reasoned if the increase in area is worth the obtained reduction in the error rate. In other words, a trade-off is created and should be properly evaluated.

A detailed analysis of how small or how large the improvements shown in Fig. 4.21 are has been made. The analysis is showed in Tab. 4.19 and the data show that nearly 44% of the improvements give marginal gains (smaller or equal to 10%). Another 22% of the

improvements are low (smaller or equal to 20%). Only 34% of the improvements can be considered of average quality or better. This also contributes for PQ's lack of effectivity at circuit level.

Table 4.19: Improvements classified into marginal, low, average, good or exceptional ones.

Circuit	Marginal	Low	Average	Good	Exceptional
c432	0	4	0	1	0
c499	0	26	0	0	32
c1355	8	50	0	0	0
c1908	128	57	5	24	28
c2670	317	104	57	90	40
c3540	407	226	81	22	66
c5315	610	253	157	125	84
c6288	0	0	0	60	212

Conclusion

The present thesis has dealt with two main concerns related to circuit reliability: analysis and improvement. It is mandatory to highlight, once more, that both are equally important and equally necessary. As technology advances into smaller and smaller dimensions, there is more and more evidence that reliability is going to be an issue. In a way, technology and design methodologies have evolved at a fast pace in the last years. Circuits with millions of gates are produced on a daily basis with surprisingly low power requirements and impressive performances. Nevertheless, the advances towards reliable circuits have not evolved at the same pace. This thesis, its techniques and methods, and the related publications, contribute so that reliable circuits are foreseeable and feasible.

When it comes to reliability analysis methods, it is clear that the literature has been expanded by several authors for more than decades now. Simulation has established itself as the prevalent method in use even with its limitations. Other solutions like PTM and SPR-MP have merits too, but are still not practical. On the other hand, the methods presented in this thesis can be easily adopted in a traditional design flow. Both SPR+ as well as SNaP can obtain reliability figures in a few seconds, that is even when considering a relatively complex circuit.

If we take SPR+ alone, it can be considered a simplified version of SPR-MP. Yet, it is practical. More than that, an important contribution that comes with the method is the analysis used to rank the fanout nodes. Such analysis can be used in different contexts. For instance, it could be used to drive other methods (such as SNaP) or it could be used to drive a synthesis algorithm. One simple way of doing it is by avoiding fanouts that are sources of large discrepancies.

The most important contribution of this thesis is, by far, the method termed SNaP. First, the method can evaluate sequential logic and that is mostly due to its hybrid approach. That being said, the fact that SNaP does use simulation is not a limitation. What happens with other simulation approaches is that they rely on fault injection, a process that is inherently time consuming. SNaP does not perform in that same way, it only uses simulation to propagate signals as in a true-value simulation.

Still concerning SNaP, there is a number of possible improvements and tweaks. One of these improvements is detailed in Section 3.2.3 and it is referred as the pessimistic version of SNaP. Not only this improvement is still practical and can be used in a real design flow, it provides a reliability figure that is meaningful even if it is not accurate. Simply by the way it is obtained, that reliability figure is always an underestimate of the actual circuit reliability. Let us assume that a given circuit has to satisfy a certain reliability target. Also assume that SNaP obtains a reliability figure that is higher than that target. That circuit is very likely to satisfy that target if the number of evaluated input samples is high enough. Unfortunately, not all methods can be used like that. As a matter for future works, the emulation capability of SNaP can be further explored to

obtain the reliability of even bigger circuits.

This thesis has also given a fair deal of attention to techniques that improve circuit reliability. The idea of using a cost function to decide which gates to harden is the heart of the techniques here proposed and it distinguishes itself from other approaches found in the literature. The results clearly state how savings in cost can be obtained. Other works use different metrics which makes the results very difficult to compare. Nevertheless, qualitatively speaking, there seems to be a good agreement between the results found here and in other works.

Following the same line of research, the cost function has been modified to account for multiple faults through a locality bias. Most of the techniques available in the literature do not deal with multiple faults and/ or charge sharing issues. This fact highlights the importance of such type of technique, especially if we consider that the occurrence of multiple faults is probably going to keep increasing for the next technologies of ICs.

Last but not least, this thesis has looked into the PQ effect. There is very little research concerning this effect, and even less if we consider the effect at circuit level like it was done in this thesis. Interestingly, in some scenarios the effect can be used to improve circuit reliability with almost zero cost. This alone is reason enough to look into the effect. The results obtained in this thesis can be extended and be used to promote PQ either at the synthesis level or at the standard cell library level.

Most of the topics covered in this thesis were published in the appropriate forums and those publications were cited along the text. A comprehensive list is also given in Appendix D. Future avenues of research were briefly discussed and can certainly be extended. Our belief is that those publications have helped the evolution of this field and have helped to solve some of its challenges.

Appendix A

Verilog RTL Code for mini□p

This code is for the version termed mini□p_v2.

```
1  module minicpu(clk, rst_n, data, opcode, result, zero);
2  input clk;
3  input rst_n;
4  input [7:0] data;
5  input [1:0] opcode;
6  output reg [7:0] result;
7  output reg zero;
8
9  reg [7:0] next_result;
10 reg next_zero;
11 reg [7:0] regA;
12 reg [7:0] next_A;
13 reg [7:0] regB;
14 reg [7:0] next_B;
15
16 always @(posedge clk or negedge rst_n) begin
17     if (rst_n == 1'b0) begin
18         zero <= 1;
19         result <= 0;
20         regA <= 0;
21         regB <= 0;
22     end
23     else begin
24         result <= next_result;
25         zero <= next_zero;
26         regA <= next_A;
27         regB <= next_B;
28     end
29 end
30
31 always @(□) begin
32     next_result = result;
33     next_zero = zero;
34     next_A = regA;
35     next_B = regB;
36
37     if (opcode == 2'b00) begin // load A
```

```
38          next_A = data;
39          next_result = data;
40      end
41      else if (opcode == 2'b01) begin // load B
42          next_B = data;
43          next_result = data;
44      end
45      else if (opcode == 2'b10) begin // add
46          next_result = regA + regB;
47          next_zero = (next_result == 0);
48      end
49      else if (opcode == 2'b11) begin // subtract
50          next_result = regA □ regB;
51          next_zero = (next_result == 0);
52      end
53  end
54
55
56  endmodule
```

Appendix B

An Experiment with Neutrons

The experiment here reported was conducted in May 2013. It consisted in placing FPGAs under a neutron beam for reliability evaluations. The experiment was conducted in the ISIS [123] grounds, more precisely in the VESUVIO facility.

Two identical A3PE3000 FPGAs from Microsemi's ProASIC3 family [106] were used. They were programmed with the circuit depicted in figures B.1 and B.2. Such circuit has two different clock domains. Since the rate at which SETs become errors depends on the circuit frequency, the vast majority of the circuit was programmed to work at a clock frequency of 133MHz.

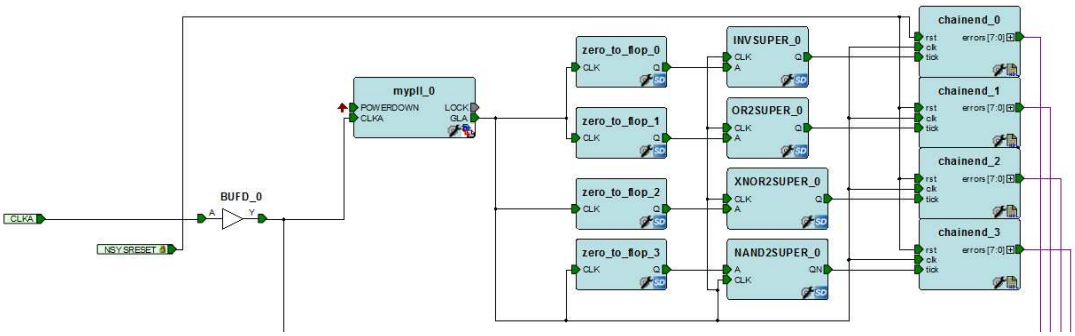


Figure B.1: Schematic of the circuit showing the modules from the fast clock domain.

Figure B.1 shows the Phase-locked Loop (PLL) responsible for generating the clock signal to the fast clock domain. It also shows four zero_to_flop modules. Each module contains a single flip-flop that is constantly fed with 0. This flip-flop, as well as many other gates in the design, is optimized out by the synthesis tool. Therefore, all gates of the fast clock domain received a special synthesis attribute (alspreserve = 1) to make sure that they are not removed by the synthesis.

Connected to each zero_to_flop module, comes a chain of gates. The image shows four chains of gates, which are made of inverters, ORs, XNORs and NAND gates. Each chain has 13200 copies of the same gate. Flip-flops are also present in the chains to keep the design's clock frequency high. Each chain is balanced in a way that the output of the last stage is always supposed to be zero. If not, then an error occurred.

Each chain is connected to a chainend module, which is responsible to accumulate the errors generated in the chains. Each chainend module has a register of 8 bits and an adder to accumulate results. The register array is protected against faults by using TMR. The

chainend module is the one that interfaces with the slow clock domain, which is shown in Fig. B.2.

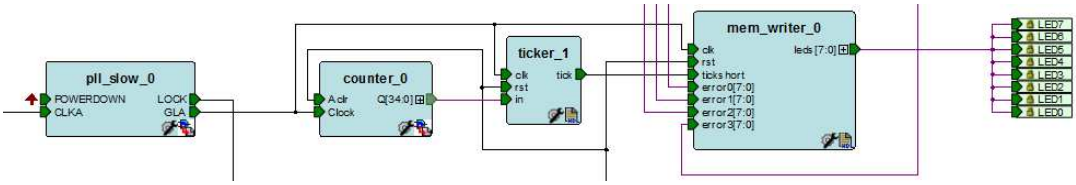


Figure B.2: Schematic of the circuit showing the modules from the slow clock domain.

Figure B.2 shows another PLL, which is configured to generate a clock signal with a frequency of 0.75MHz. Nevertheless, this frequency is still relatively high and a 35-bit counter is used to generate a signal with an even slower frequency. This signal drives the mem_writer module which is responsible for driving the LEDs of the board.

The LEDs are controlled by an FSM that shows the output of the chains (total number of errors) and cycles through the four chains. The FSM was encoded using a safe FSM so, even if an SET or SEU occurred at such low frequency, the FSM would not go to an illegal state. In other words, any error shown at the LEDs must come from the chains and not from anywhere else. Also, since the FPGAs used are flash-based, there is no need to be concerned with SEUs in the configuration logic as in traditional SRAM-based FPGAs.

Figures B.3 and B.4 show the setup of the boards inside the chamber and the beam source. The webcam used for monitoring the results is also shown in Fig. B.3.

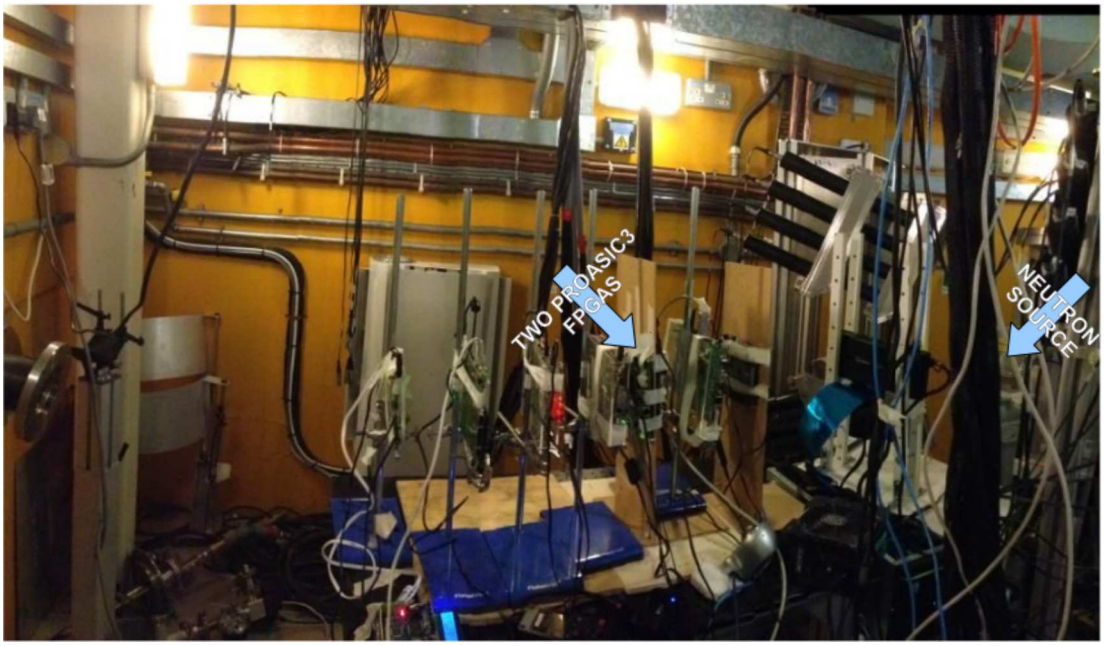


Figure B.3: Full experiment setup showing the FPGA boards, the neutron source and the webcam.

Figure B.5 shows the laser targetting system used. Since the neutron beam suffers from scattering, properly targetting the FPGA is required.

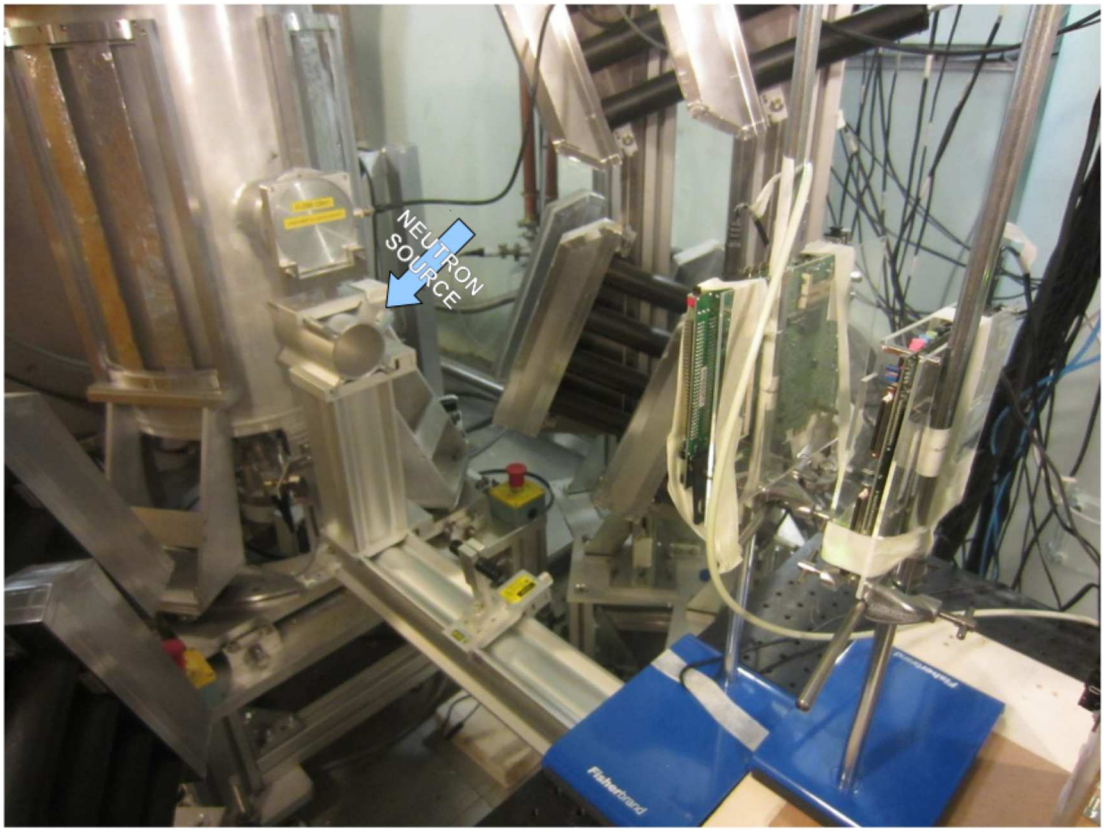


Figure B.4: A closer look at the neutron source.

Figure B.6 shows one screenshot of the webcam monitoring system used from outside the chamber. The image has a blue-ish background because the chamber turns blue lights on when the beam is on.

After 5 days of experiments, the boards were irradiated with an average flux of 37640 neutrons per squared centimeter per second. The first board was placed 99cm from the source while the second one was placed 102cm from the source. The beam was turned on/ off 52 times during the five days for multiple reasons.

The goal of the experiment was to register enough errors such that one chain could be considered less reliable than the others. This could be considered enough evidence for concluding that SETs caused that disparity (since the number of flip-flops in each chain is the same). Nevertheless, only a handful of errors were registered during the whole experiment time. The expected number of errors was supposed to be from the order of hundreds of errors, given the results found in [124]. Below are listed some of the reasons that might have contributed for this:

- The FPGA used in [124] is from the same family but the device is different.
- The design used in [124] is mainly composed of flip-flops while the designed reported in here is mainly combinational.

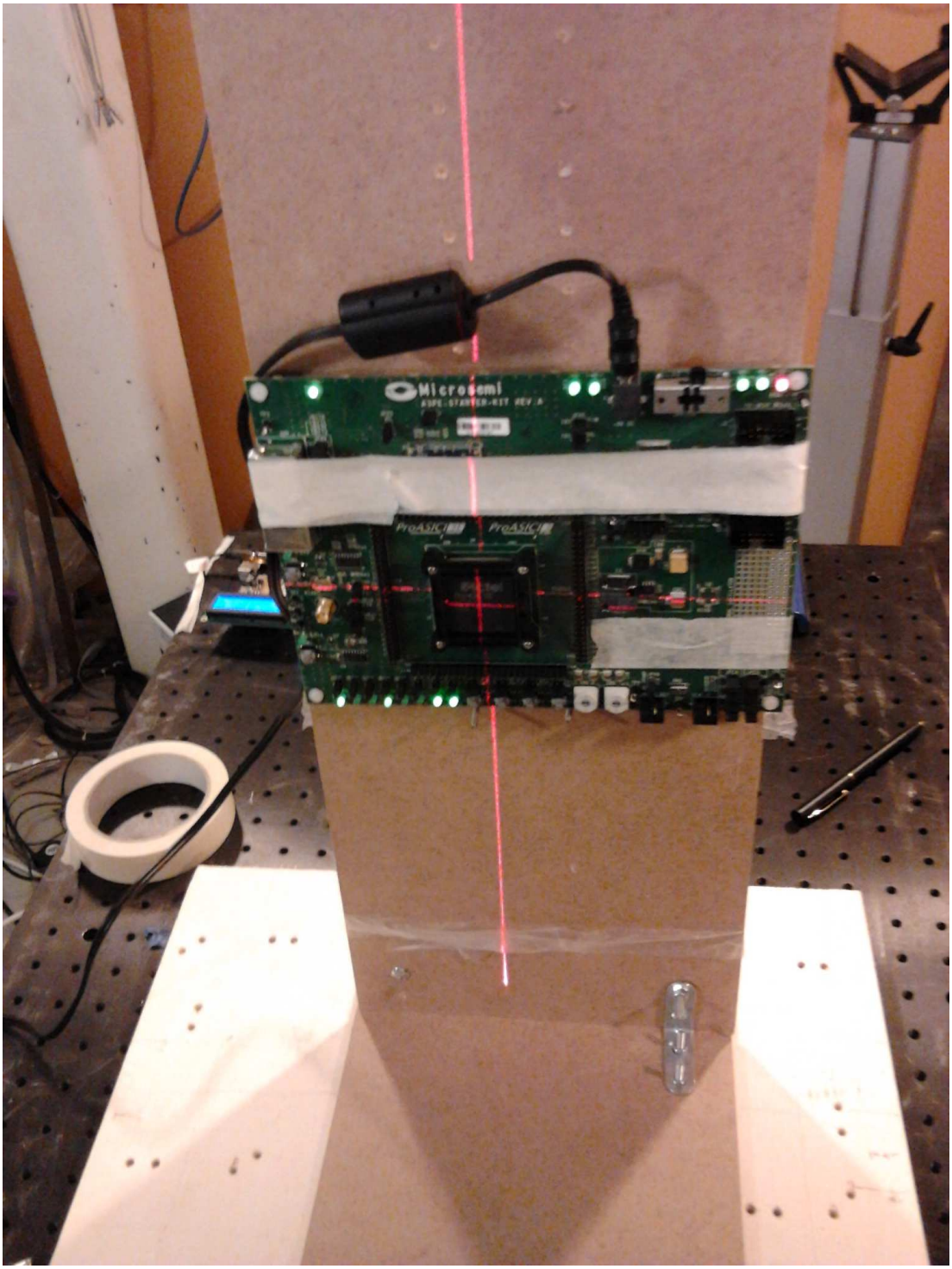


Figure B.5: Laser targetting system.

Figure B.6: Webcam monitoring of the experiment.

Appendix C

Example of an Instrumented Circuit Description

This code is SNaP's modified version of the c17 circuit. Each NAND2 gate is modelled using FSMs as described in Section 3.2.

```
1  // generated by SNaP
2  // Tue Jul 2 2013
3  // 12:00:52
4
5  module c17(clk, n_rst, N1, N2, N3, N6, N7, N22, N23, acc_N22, acc_N23,
       ready);
6  localparam ACC_WIDTH = 10;
7  input clk;
8  input n_rst;
9  input N1;
10 input N2;
11 input N3;
12 input N6;
13 input N7;
14 output N22;
15 output N23;
16 output [ACC_WIDTH-1:0] acc_N22;
17 output [ACC_WIDTH-1:0] acc_N23;
18 output ready;
19
20 wire tick;
21
22 wire N10;
23 wire N11;
24 wire N16;
25 wire N19;
26 // these are the regular wires
27
28 wire [ACC_WIDTH-1:0] acc_N10, acc_N11, acc_N16, acc_N19;
29 // acc for the regular wires
30 wire go_N10, go_N11, go_N16, go_N19;
31 // go (ready) signals for the regular wires
32
```

```
33 wire [ACC_WIDTH-1:0] acc_N1;
34 wire [ACC_WIDTH-1:0] acc_N2;
35 wire [ACC_WIDTH-1:0] acc_N3;
36 wire [ACC_WIDTH-1:0] acc_N6;
37 wire [ACC_WIDTH-1:0] acc_N7;
38 // special wires representing the acc signals from the inputs
39
40 assign acc_N1 = 0;
41 assign acc_N2 = 0;
42 assign acc_N3 = 0;
43 assign acc_N6 = 0;
44 assign acc_N7 = 0;
45
46 wire go_N1, go_N2, go_N3, go_N6, go_N7;
47 //special wires representing the go signals from the inputs
48
49 assign go_N1 = 1'b1;
50 assign go_N2 = 1'b1;
51 assign go_N3 = 1'b1;
52 assign go_N6 = 1'b1;
53 assign go_N7 = 1'b1;
54
55 wire go_N22, go_N23;
56 // special wires representing the go signals for the outputs
57
58 nand2 #(.GF(256), .OFSW(ACC_WIDTH) ) NAND2_1(.clk(clk), .n_rst(tick),
      .in1(N1), .in2(N3), .go1(go_N1), .go2(go_N3), .acc1(acc_N1), .acc2(
      acc_N3), .out(N10), .acc(acc_N10), .go(go_N10) );
59 nand2 #(.GF(256), .OFSW(ACC_WIDTH) ) NAND2_2(.clk(clk), .n_rst(tick),
      .in1(N3), .in2(N6), .go1(go_N3), .go2(go_N6), .acc1(acc_N3), .acc2(
      acc_N6), .out(N11), .acc(acc_N11), .go(go_N11) );
60 nand2 #(.GF(256), .OFSW(ACC_WIDTH) ) NAND2_3(.clk(clk), .n_rst(tick),
      .in1(N2), .in2(N11), .go1(go_N2), .go2(go_N11), .acc1(acc_N2), .
      acc2(acc_N11), .out(N16), .acc(acc_N16), .go(go_N16) );
61 nand2 #(.GF(256), .OFSW(ACC_WIDTH) ) NAND2_4(.clk(clk), .n_rst(tick),
      .in1(N11), .in2(N7), .go1(go_N11), .go2(go_N7), .acc1(acc_N11), .
      acc2(acc_N7), .out(N19), .acc(acc_N19), .go(go_N19) );
62 nand2 #(.GF(256), .OFSW(ACC_WIDTH) ) NAND2_5(.clk(clk), .n_rst(tick),
      .in1(N10), .in2(N16), .go1(go_N10), .go2(go_N16), .acc1(acc_N10), .
      acc2(acc_N16), .out(N22), .acc(acc_N22), .go(go_N22) );
63 nand2 #(.GF(256), .OFSW(ACC_WIDTH) ) NAND2_6(.clk(clk), .n_rst(tick),
      .in1(N16), .in2(N19), .go1(go_N16), .go2(go_N19), .acc1(acc_N16), .
      acc2(acc_N19), .out(N23), .acc(acc_N23), .go(go_N23) );
64 assign ready = go_N22 && go_N23;
65
66 assign tick = (ready == 1'b0) && (n_rst == 1'b1);
67 endmodule
```

Appendix D

List of Publications

- Selective Hardening Methodology for Combinational Logic, in Test Workshop (LATW), 13th Latin American, 2012.

- Selective Hardening Methodology Concerning Multiple Faults, in Nuclear and Space Radiation Effects Conference (NSREC), IEEE, 2012.

- Towards the Mitigation of Multiple Faults Induced by Single Event Effects: Combining Global TMR and Selective Hardening, in Radiation and Its Effects on Components and Systems (RADECS), 13th European Conference on, 2012.

- Single-Event-Induced Charge Sharing Effects in TMR with Different Levels of Granularity, in Radiation and Its Effects on Components and Systems (RADECS), 13th European Conference on, 2012.

- Exploring the Feasibility of Selective Hardening for Combinational Logic, in European Symposium on the Reliability of Electron Devices, Failure Physics and Analysis (ESREF), 2012.

- Automatic Selective Hardening Against Soft Errors: A Cost-based and Regularity-aware Approach, in Electronics, Circuits and Systems (ICECS), 19th IEEE International Conference on, 2012.

- Selective hardening methodology targeted at single and multiple faults, Journées Nationales du Réseau Doctoral en Micro-nanoélectronique (JNRDM), 2012.

- Exploring the Feasibility of Selective Hardening for Combinational Logic, Microelectronics Reliability, vol. 52, no. 9-10, pp. 1843 - 1847, 2012.

- Reliability Estimation Methods: Trade-offs Between Complexity and Accuracy, in South Symposium on Microelectronics (SIM), 2012.

- Selective Hardening Against Multiple Faults Employing a Net-based Reliability Analysis, in Northeast Workshop on Circuits and Systems (NEWCAS). International IEEE, 2013.

- ESTIMATION DE LA FIABILITE D'UN CIRCUIT LOGIQUE, Patent FR 13 52 279, 2013.

- SNaP: a Novel Hybrid Method for Circuit Reliability Assessment Under Multiple Faults, in European Symposium on the Reliability of Electron Devices, Failure Physics and Analysis(ESREF), 2013.

- SNaP: a Novel Hybrid Method for Circuit Reliability Assessment Under Multiple Faults, Microelectronics Reliability, 2013.

- Circuit-level Hardening Against Multiple Faults: Combining Global TMR and Selective Hardening, Journées Nationales du Réseau Doctoral en Micro-nanoélectronique (JNRDM),

2013.

- Reliability Assessment of Combinational Logic Using First-Order-Only Fanout Reconvergence Analysis, in Midwest Symposium on Circuits and Systems (MWSCAS), 2013.

- A defect-tolerant area-efficient multiplexer for basic blocks in SRAM-based FPGAs,, in European Symposium on the Reliability of Electron Devices, Failure Physics and Analysis(ESREF), 2013.

- A defect-tolerant area-efficient multiplexer for basic blocks in SRAM-based FPGAs,, Microelectronics Reliability, 2013.

Bibliography

[1] International Technology Roadmap for Semiconductors. (2011, Dec) ITRS Report 2011, Executive Summary. [Available] http://www.itrs.net/Links/2012ITRS/.

[2] A. Saha and N. Manna, Digital Principles and Logic Design, ser. Infinity Science Series. JONES AND BARTLETT P, 2009. [Online]. Available: http://books.google.com/books?id=-mLKFXxtZk4C

[3] D. Gajski, Principles of Digital Design, ser. Prentice Hall International Editions. Prentice-Hall International, 1997. [Online]. Available: http://books.google.com/books?id=xCz6PAAACAAJ

[4] M. Hutton, R. Yuan, J. Schleicher, G. Baeckler, S. Cheung, K. K. Chua, and H. K. Phoo, "A Methodology for FPGA to Structured-ASIC Synthesis and Verification," in Proceedings of the conference on Design, automation and test in Europe (DATE), 2006, pp. 64–69. [Online]. Available: http://portal.acm.org/citation.cfm?id=1131355.1131369

[5] A. Avizienis, J.-C. Laprie, B. Randell, and Vytautas. (2000) Fundamental Concepts of Dependability. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.1400

[6] J. Azambuja, S. Pagliarini, L. Rosa, and F. Kastensmidt, "Exploring the Limitations of Software-based Techniques in SEE Fault Coverage," Journal of Electronic Testing Theory and Applications, pp. 1–10, 2011. [Online]. Available: http://dx.doi.org/10.1007/s10836-011-5218-7

[7] R. Baumann, "Soft Errors in Advanced Semiconductor Devices-part I: the three Radiation Sources," Device and Materials Reliability, IEEE Transactions on, vol. 1, no. 1, pp. 17–22, mar 2001.

[8] T. J. O'Gorman, J. M. Ross, A. H. Taber, J. F. Ziegler, H. P. Muhlfeld, C. J. Montrose, H. W. Curtis, and J. L. Walsh, "Field Testing for Cosmic Ray Soft Errors in Semiconductor Memories," IBM Journal of Research and Development, vol. 40, no. 1, pp. 41–50, jan. 1996.

[9] R. D. Eldred, "Test Routines Based on Symbolic Logical Statements," J. ACM, vol. 6, pp. 33–37, January 1959. [Online]. Available: http://doi.acm.org/10.1145/320954.320957

[10] J. M. Galey, R. E. Norby, and J. P. Roth, "Techniques for the diagnosis of switching circuit failures," in Switching Circuit Theory and Logical Design (SWCT). Proceedings of the Second Annual Symposium on, oct. 1961, pp. 152–160.

[11] M. Bushnell and V. Agrawal, "Fault Modeling," in Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits. Springer US, 2002, vol. 17, pp. 57–80. [Online]. Available: http://dx.doi.org/10.1007/0-306-47040-3_4

[12] C. Stapper, F. Armstrong, and K. Saji, "Integrated Circuit Yield Statistics," Proceedings of the IEEE, vol. 71, no. 4, pp. 453–470, 1983.

[13] I. Koren and Z. Koren, "Defect Tolerance in VLSI Circuits: Techniques and Yield Analysis," Proceedings of the IEEE, vol. 86, no. 9, pp. 1819–1838, 1998.

[14] B. Benware, C. Schuermyer, M. Sharma, and T. Herrmann, "Determining a Failure Root Cause Distribution From a Population of Layout-Aware Scan Diagnosis Results," Design Test of Computers, IEEE, vol. 29, no. 1, pp. 8–18, 2012.

[15] F. Ferguson, M. Taylor, and T. Larrabee, "Testing for Parametric Faults in Static CMOS Circuits," in Test Conference. Proceedings of the International, 1990, pp. 436–443.

[16] F. Sexton, "Destructive Single-event Effects in Semiconductor Devices and ICs," Nuclear Science, IEEE Transactions on, vol. 50, no. 3, pp. 603–621, 2003.

[17] J. Schwank, V. Ferlet-Cavrois, M. R. Shaneyfelt, P. Paillet, and P. Dodd, "Radiation Effects in SOI Technologies," Nuclear Science, IEEE Transactions on, vol. 50, no. 3, pp. 522–538, 2003.

[18] P. E. Dodd and L. W. Massengill, "Basic Mechanisms and Modeling of Single-event Upset in Digital Microelectronics," Nuclear Science, IEEE Transactions on, vol. 50, no. 3, pp. 583–602, Jun. 2003. [Online]. Available: http://dx.doi.org/10.1109/TNS.2003.813129

[19] N. Seifert, "Radiation-induced Soft Errors: A Chip-level Modeling Perspective," Found. Trends Electron. Des. Autom., vol. 4, pp. 99–221, Feb. 2010. [Online]. Available: http://dx.doi.org/10.1561/1000000018

[20] X. Li, S. V. Adve, P. Bose, and J. A. Rivers, "Online Estimation of Architectural Vulnerability Factor for Soft Errors," in Symposium on Computer Architecture, Proceedings of the 35th Annual International (ISCA). IEEE Computer Society, 2008, pp. 341–352. [Online]. Available: http://dx.doi.org/10.1109/ISCA.2008.9

[21] T. Karnik and P. Hazucha, "Characterization of Soft Errors Caused by Single Event Upsets in CMOS Processes," Dependable and Secure Computing, IEEE Transactions on, vol. 1, no. 2, pp. 128 – 143, april-june 2004.

[22] H. Nguyen and Y. Yagil, "A Systematic Approach to SER Estimation and Solutions," in Reliability Physics Symposium (IRPS). Proceedings of the 41st Annual IEEE International, april 2003, pp. 60 – 70.

[23] P. Shivakumar, M. Kistler, S. Keckler, D. Burger, and L. Alvisi, "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic," in Dependable Systems and Networks (DSN). Proceedings of the International Conference on, 2002, pp. 389 – 398.

[24] A. Dixit and A. Wood, "The impact of new technology on soft error rates," in Reliability Physics Symposium (IRPS), 2011 IEEE International, 2011, pp. 5B.4.1–5B.4.7.

[25] M. Casey, A. Duncan, B. Bhuva, W. Robinson, and L. Massengill, "Simulation Study on the Effect of Multiple Node Charge Collection on Error Cross-Section in CMOS Sequential Logic," Nuclear Science, IEEE Transactions on, vol. 55, no. 6, pp. 3136–3140, 2008.

[26] S. Pagliarini, F. Kastensmidt, L. Entrena, A. Lindoso, and E. Millan, "Analyzing the Impact of Single-Event-Induced Charge Sharing in Complex Circuits," Nuclear Science, IEEE Transactions on, vol. 58, no. 6, pp. 2768 –2775, Dec. 2011.

[27] J. Ahlbin, L. Massengill, B. Bhuva, B. Narasimham, M. Gadlage, and P. Eaton, "Single-Event Transient Pulse Quenching in Advanced CMOS Logic Circuits," Nuclear Science, IEEE Transactions on, vol. 56, no. 6, pp. 3050 –3056, dec. 2009.

[28] N. Seifert, X. Zhu, and L. Massengill, "Impact of Scaling on Soft-error Rates in Commercial Microprocessors," Nuclear Science, IEEE Transactions on, vol. 49, no. 6, pp. 3100 – 3106, dec 2002.

[29] P. Liden, P. Dahlgren, R. Johansson, and J. Karlsson, "On Latching Probability of Particle Induced Transients in Combinational Networks," in Fault-Tolerant Computing (FTCS). Digest of Papers from the Twenty-Fourth International Symposium on, jun 1994, pp. 340 –349.

[30] M. Baze and S. Buchner, "Attenuation of Single Event Induced Pulses in CMOS Combinational Logic," Nuclear Science, IEEE Transactions on, vol. 44, no. 6, pp. 2217 –2223, dec 1997.

[31] R. Ramanarayanan, V. Degalahal, R. Krishnan, J. Kim, V. Narayanan, Y. Xie, M. Irwin, and K. Unlu, "Modeling Soft Errors at the Device and Logic Levels for Combinational Circuits," Dependable and Secure Computing, IEEE Transactions on, vol. 6, no. 3, pp. 202 –216, july 2009.

[32] N. George and J. Lach, "Characterization of Logical Masking and Error Propagation in Combinational Circuits and Effects on System Vulnerability," in Dependable Systems Networks (DSN), IEEE/IFIP 41st International Conference on, june 2011, pp. 323 –334.

[33] F. Wang, Y. Xie, R. Rajaraman, and B. Vaidyanathan, "Soft Error Rate Analysis for Combinational Logic Using An Accurate Electrical Masking Model," in VLSI Design, 2007 (VLSID). 20th International Conference on, jan. 2007, pp. 165 –170.

[34] R. Rao, K. Chopra, D. Blaauw, and D. Sylvester, "An Efficient Static Algorithm for Computing the Soft Error Rates of Combinational Circuits," in Design, Automation and Test in Europe (DATE). Proceedings of the, vol. 1, march 2006, pp. 1 –6.

[35] S. Krishnaswamy, I. L. Markov, and J. P. Hayes, "On the Role of Timing Masking in Reliable Logic Circuit Design," in Design Automation Conference (DAC), Proceedings of the 45th annual. ACM, 2008, pp. 924–929. [Online]. Available: http://doi.acm.org/10.1145/1391469.1391703

[36] E. Czeck and D. Siewiorek, "Effects of Transient Gate-level Faults on Program Behavior," in Fault-Tolerant Computing, 1990. FTCS-20. Digest of Papers., 20th International Symposium, jun 1990, pp. 236 –243.

[37] S. Kim and A. Somani, "Soft Error Sensitivity Characterization for Microprocessor Dependability Enhancement Strategy," in Dependable Systems and Networks (DSN). Proceedings of the International Conference on, 2002, pp. 416 – 425.

[38] X. Li, S. Adve, P. Bose, and J. Rivers, "SoftArch: an Architecture-level Tool for Modeling and Analyzing Soft Errors," in Dependable Systems and Networks (DSN). Proceedings of the International Conference on, july 2005, pp. 496 – 505.

[39] K. R. Walcott, G. Humphreys, and S. Gurumurthi, "Dynamic Prediction of Architectural Vulnerability from Microarchitectural State," SIGARCH Comput. Archit. News, vol. 35, pp. 516–527, June 2007. [Online]. Available: http://doi.acm.org/10.1145/1273440.1250726

[40] D. T. Franco, "Fiabilité du Signal des Circuits Logiques Combinatoires sous Fautes Simultanées Multiples," Ph.D. dissertation, École Nationale Supérieure des Télécommunications, 2009.

[41] W. Kuo, "Reliability Enhancement Through Optimal Burn-In," Reliability, IEEE Transactions on, vol. R-33, no. 2, pp. 145–156, 1984.

[42] X. Li, J. Qin, and J. Bernstein, "Compact Modeling of MOSFET Wearout Mechanisms for Circuit-Reliability Simulation," Device and Materials Reliability, IEEE Transactions on, vol. 8, no. 1, pp. 98–121, 2008.

[43] M.-C. Hsueh, T. Tsai, and R. Iyer, "Fault Injection Techniques and Tools," Computer, vol. 30, no. 4, pp. 75 –82, apr 1997.

[44] J. Clark and D. Pradhan, "Fault Injection: a Method for Validating Computer-system Dependability," Computer, vol. 28, no. 6, pp. 47 –56, jun 1995.

[45] IEEE, "Standard for the Verilog Hardware Description Language," IEEE, 1995.

[46] ——, "Standard VHDL Language Reference Manual," IEEE, 1987.

[47] E. Jenn, J. Arlat, M. Rimen, J. Ohlsson, and J. Karlsson, "Fault Injection into VHDL Models: the MEFISTO Tool," in Fault-Tolerant Computing (FTCS). Digest of Papers from the Twenty-Fourth International Symposium on, jun 1994, pp. 66 –75.

[48] L. Massengill, A. Baranski, D. Van Nort, J. Meng, and B. Bhuva, "Analysis of Single-event Effects in Combinational Logic-simulation of the AM2901 Bitslice Processor," Nuclear Science, IEEE Transactions on, vol. 47, no. 6, pp. 2609 –2615, dec 2000.

[49] S. Rezgui, G. Swift, R. Velazco, and F. Farmanesh, "Validation of an SEU Simulation Technique for a Complex Processor: PowerPC7400," Nuclear Science, IEEE Transactions on, vol. 49, no. 6, pp. 3156 – 3162, dec 2002.

[50] M. Valderas, P. Peronnard, C. Lopez Ongil, R. Ecoffet, F. Bezerra, and R. Velazco, "Two Complementary Approaches for Studying the Effects of SEUs on Digital Processors," Nuclear Science, IEEE Transactions on, vol. 54, no. 4, pp. 924 –928, aug. 2007.

[51] J. Arlat, M. Aguera, L. Amat, Y. Crouzet, J.-C. Fabre, J.-C. Laprie, E. Martins, and D. Powell, "Fault Injection for Dependability Validation: a Methodology and Some Applications," Software Engineering, IEEE Transactions on, vol. 16, no. 2, pp. 166 – 182, feb 1990.

[52] J. Karlsson, P. Liden, P. Dahlgren, R. Johansson, and U. Gunneflo, "Using Heavy-ion Radiation to Validate Fault-handling Mechanisms," Micro, IEEE, vol. 14, no. 1, pp. 8 –23, feb 1994.

[53] S.-A. Hwang, J.-H. Hong, and C.-W. Wu, "Sequential Circuit Fault Simulation Using Logic Emulation," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 17, no. 8, pp. 724–736, 1998.

[54] P. Civera, L. Macchiarulo, M. Rebaudengo, M. Reorda, and M. Violante, "Exploiting Circuit Emulation for Fast Hardness Evaluation," Nuclear Science, IEEE Transactions on, vol. 48, no. 6, pp. 2210–2216, 2001.

[55] M. Aguirre, V. Baena, J. Tombs, and M. Violante, "A New Approach to Estimate the Effect of Single Event Transients in Complex Circuits," Nuclear Science, IEEE Transactions on, vol. 54, no. 4, pp. 1018–1024, 2007.

[56] C. Lopez-Ongil, M. Garcia-Valderas, M. Portela-Garcia, and L. Entrena, "Autonomous Fault Emulation: A New FPGA-Based Acceleration System for Hardness Evaluation," Nuclear Science, IEEE Transactions on, vol. 54, no. 1, pp. 252–261, 2007.

[57] L. Entrena, M. Garcia-Valderas, R. Fernandez-Cardenal, A. Lindoso, M. Portela, and C. Lopez-Ongil, "Soft Error Sensitivity Evaluation of Microprocessors by Multilevel Emulation-Based Fault Injection," Computers, IEEE Transactions on, vol. 61, no. 3, pp. 313–322, 2012.

[58] P. Sedcole, B. Blodget, T. Becker, J. Anderson, and P. Lysaght, "Modular Dynamic Reconfiguration in Virtex FPGAs," Computers and Digital Techniques, IEE Proceedings -, vol. 153, no. 3, pp. 157–164, 2006.

[59] L. Antoni, R. Leveugle, and B. Feher, "Using Run-time Reconfiguration for Fault Injection in Hardware Prototypes," in Defect and Fault Tolerance in VLSI Systems, 2002. DFT 2002. Proceedings. 17th IEEE International Symposium on, 2002, pp. 245–253.

[60] L. Naviner, J.-F. Naviner, G. Goncalves Dos Santos Junior, E. Crespo Marques, and N. Maciel Paiva Junior, "FIFA: A Fault-injection-fault-analysis-based Tool for Reliability Assessment at RTL Level," Microelectronics Reliability, vol. 51, no. 9-11, Jul. 2011.

[61] M. de Vasconcelos, D. Franco, L. de B.Naviner, and J.-F. Naviner, "Reliability Analysis of Combinational Circuits Based on a Probabilistic Binomial Model," in Circuits and Systems and TAISA Conference (NEWCAS-TAISA). Joint 6th International IEEE Northeast Workshop on, 2008, pp. 310–313.

[62] J. Partridge, E. C. Hall, and L. D. Hanley, "The Application of Failure Analysis in Procuring and Screening of Integrated Circuits," in Physics of Failure in Electronics, 1965. Fourth Annual Symposium on the, 1965, pp. 95–139.

[63] J. Autran, P. Roche, J. Borel, C. Sudre, K. Castellani-Coulie, D. Munteanu, T. Parrassin, G. Gasiot, and J.-P. Schoellkopf, "Altitude SEE Test European Platform (ASTEP) and First Results in CMOS 130 nm SRAM," Nuclear Science, IEEE Transactions on, vol. 54, no. 4, pp. 1002–1009, 2007.

[64] Z. Torok, S. Platt, and C. X. Xiao, "SEE-inducing Effects of Cosmic Rays at the High-Altitude Research Station Jungfraujoch Compared to Accelerated Test Data," in Radiation and Its Effects on Components and Systems, 2007. RADECS 2007. 9th European Conference on, 2007, pp. 1–6.

[65] L. Artola, R. Velazco, G. Hubert, S. Duzellier, T. Nuns, B. Guerard, P. Peronnard, W. Mansour, F. Pancher, and F. Bezerra, "In Flight SEU/ MCU Sensitivity of Commercial Nanometric SRAMs: Operational Estimations," Nuclear Science, IEEE Transactions on, vol. 58, no. 6, pp. 2644–2651, 2011.

[66] S. Buchner, M. Baze, D. Brown, D. McMorrow, and J. Melinger, "Comparison of Error Rates in Combinational and Sequential Logic," Nuclear Science, IEEE Transactions on, vol. 44, no. 6, pp. 2209–2216, 1997.

[67] A. Chugg, J. Ward, J. McIntosh, N. Flynn, P. Duncan, T. Barber, and C. Poivey, "Improved Fine-scale Laser Mapping of Component SEE Sensitivity," in Radiation and Its Effects on Components and Systems (RADECS), 12th European Conference on, 2011, pp. 442–448.

[68] ——, "Improved Fine-Scale Laser Mapping of Component SEE Sensitivity," Nuclear Science, IEEE Transactions on, vol. 59, no. 4, pp. 1007–1014, 2012.

[69] E. Cannon, M. Cabanas-Holmen, J. Wert, T. Amort, R. Brees, J. Koehn, B. Meaker, and E. Normand, "Heavy Ion, High-Energy, and Low-Energy Proton SEE Sensitivity of 90-nm RHBD SRAMs," Nuclear Science, IEEE Transactions on, vol. 57, no. 6, pp. 3493–3499, 2010.

[70] P. Rech, J.-M. Galliere, P. Girard, A. Griffoni, J. Boch, F. Wrobel, F. Saigne, and L. Dilillo, "Neutron-induced Multiple Bit Upsets on Dynamically-stressed Commercial SRAM Arrays," in Radiation and Its Effects on Components and Systems (RADECS), 12th European Conference on, 2011, pp. 274–280.

[71] J. Schwank, M. Shaneyfelt, J. Baggio, P. Dodd, J. Felix, V. Ferlet-Cavrois, P. Paillet, D. Lambert, F. Sexton, G. L. Hash, and E. Blackmore, "Effects of Particle Energy on Proton-induced Single-event Latchup," Nuclear Science, IEEE Transactions on, vol. 52, no. 6, pp. 2622–2629, 2005.

[72] B. Sierawski, M. Mendenhall, R. Reed, M. Clemens, R. Weller, R. Schrimpf, E. Blackmore, M. Trinczek, B. Hitti, J. Pellish, R. Baumann, S.-J. Wen, R. Wong, and N. Tam, "Muon-Induced Single Event Upsets in Deep-Submicron Technology," Nuclear Science, IEEE Transactions on, vol. 57, no. 6, pp. 3273–3278, 2010.

[73] G. Gasiot, D. Giot, and P. Roche, "Alpha-Induced Multiple Cell Upsets in Standard and Radiation Hardened SRAMs Manufactured in a 65 nm CMOS Technology," Nuclear Science, IEEE Transactions on, vol. 53, no. 6, pp. 3479–3486, 2006.

[74] C. Gelderloos, R. J. Peterson, M. Nelson, and J. Ziegler, "Pion-induced Soft Upsets in 16 mbit DRAM Chips," Nuclear Science, IEEE Transactions on, vol. 44, no. 6, pp. 2237–2242, 1997.

[75] K. N. Patel, I. L. Markov, and J. P. Hayes, "Evaluating Circuit Reliability Under Probabilistic Gate-Level Fault Models," in In International Workshop on Logic Synthesis (IWLS), 2003, pp. 59–64.

[76] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Accurate Reliability Evaluation and Enhancement via Probabilistic Transfer Matrices," in Proc. Design Automation and Test in Europe (DATE, 2005, pp. 282–287.

[77] T. Larrabee, "Test Pattern Generation Using Boolean Satisfiability," IEEE Transactions on Computer-Aided Design, pp. 4–15, 1992.

[78] D. T. Franco, M. C. Vasconcelos, L. Naviner, and J.-F. Naviner, "Signal Probability for Reliability Evaluation of Logic Circuits," Microelectronics Reliability, vol. 48, no. 8-9, pp. 1586 – 1591, 2008.

[79] D. Franco, M. Vasconcelos, L. Naviner, and J.-F. Naviner, "Reliability of Logic Circuits Under Multiple Simultaneous Faults," in Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on, 2008, pp. 265–268.

[80] B. Krishnamurthy and I. Tollis, "Improved Techniques for Estimating Signal Probabilities," Computers, IEEE Transactions on, vol. 38, no. 7, pp. 1041–1045, 1989.

[81] J. Han, H. Chen, E. Boykin, and J. A. B. Fortes, "Reliability evaluation of logic circuits using probabilistic gate models," Microelectronics Reliability, vol. 51, pp. 468–476, 2011.

[82] J. T. Flaquer, J. Daveau, L. Naviner, and P. Roche, "Fast reliability analysis of combinatorial logic circuits using conditional probabilities," Microelectronics Reliability, vol. 50, no. 9 - 11, pp. 1215 – 1218, 2010, ¡ce:title¿21st European Symposium on the Reliability of Electron Devices, Failure Physics and Analysis¡/ ce:title¿. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0026271410003318

[83] A. Abdollahi, "Probabilistic decision diagrams for exact probabilistic analysis," in Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design, ser. ICCAD '07. Piscataway, NJ, USA: IEEE Press, 2007, pp. 266–272. [Online]. Available: http://dl.acm.org/citation.cfm?id=1326073.1326128

[84] J. von Neumann, "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components," Automata Studies, vol. 34, pp. 43–99, 1956. [Online]. Available: http://www.cs.caltech.edu/courses/cs191/paperscs191/VonNeumann56.pdf

[85] E. C. Marques, L. A. de Barros Naviner, and J.-F. Naviner, "An Efficient Tool for Reliability Improvement Based on TMR," Microelectronics Reliability, vol. 50, no. 9-11, pp. 1247 – 1250, 2010.

[86] S. N. Pagliarini, L. A. d. B. Naviner, and J.-F. Naviner, "Selective Hardening Methodology for Combinational Logic," in Test Workshop (LATW), 13th Latin American, Apr. 2012.

[87] S. N. Pagliarini, L. A. de B. Naviner, and J.-F. Naviner, "Selective Hardening Methodology Concerning Multiple Faults," in Nuclear and Space Radiation Effects Conference (NSREC), IEEE, Jul. 2012.

[88] S. N. Pagliarini, L. A. B. Naviner, and J.-F. Naviner, "Towards the Mitigation of Multiple Faults Induced by Single Event Effects: Combining Global TMR and Selective Hardening," in Radiation and Its Effects on Components and Systems (RADECS), 13th European Conference on, 2012.

[89] F. Almeida, F. L. Kastensmidt, S. N. Pagliarini, L. Entrena, A. Lindoso, E. S. Millan, E. Chielli, L. A. de Barros Naviner, and J.-F. Naviner, "Single-Event-Induced Charge Sharing Effects in TMR with Different Levels of Granularity," in Radiation and Its Effects on Components and Systems (RADECS), 13th European Conference on, 2012.

[90] S. N. Pagliarini, L. A. de B. Naviner, and J.-F. Naviner, "Selective Hardening Against Multiple Faults Employing a Net-based Reliability Analysis," in Northeast Workshop on Circuits and Systems (NEWCAS). International IEEE, Jun. 2013.

[91] L. Entrena, A. Lindoso, M. Valderas, M. Portela, and C. Ongil, "Analysis of SET Effects in a PIC Microprocessor for Selective Hardening," Nuclear Science, IEEE Transactions on, vol. 58, no. 3, pp. 1078 –1085, Jun. 2011.

[92] L. A. de B. Naviner, J.-F. Naviner, T. Ban, and G. S. Gutemberg, "Reliability Analysis Based on Significance," in Argentine School of Micro-Nanoelectronics Technology and Applications (EAMTA), aug. 2011, pp. 1 –7.

[93] I. Polian, S. Reddy, and B. Becker, "Scalable Calculation of Logical Masking Effects for Selective Hardening Against Soft Errors," in Symposium on VLSI (ISVLSI). IEEE Computer Society Annual, april 2008, pp. 257 –262.

[94] Q. Zhou and K. Mohanram, "Cost-effective Radiation Hardening Technique for Combinational Logic," in Computer Aided Design (ICCAD). IEEE/ACM International Conference on, nov. 2004, pp. 100 – 106.

[95] D. Limbrick, D. Black, K. Dick, N. Atkinson, N. Gaspard, J. Black, W. Robinson, and A. Witulski, "Impact of Logic Synthesis on Soft Error Vulnerability Using a 90-nm Bulk CMOS Digital Cell Library," in Southeastcon, 2011 Proceedings of IEEE, 2011, pp. 430–434.

[96] F. de Lima Kastensmidt, G. Neuberger, R. Hentschke, L. Carro, and R. Reis, "Designing Fault-tolerant Techniques for SRAM-based FPGAs," Design Test of Computers, IEEE, vol. 21, no. 6, pp. 552–562, 2004.

[97] R. Velazco, D. Bessot, S. Duzellier, R. Ecoffet, and R. Koga, "Two cmos memory cells suitable for the design of seu-tolerant vlsi circuits," Nuclear Science, IEEE Transactions on, vol. 41, no. 6, pp. 2229–2234, 1994.

[98] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron cmos technology," Nuclear Science, IEEE Transactions on, vol. 43, no. 6, pp. 2874–2878, 1996.

[99] M. Ghahroodi, M. Zwolinski, and E. Ozer, "Radiation hardening by design: A novel gate level approach," in Adaptive Hardware and Systems (AHS), 2011 NASA/ESA Conference on, 2011, pp. 74–79.

[100] K. Parker and E. McCluskey, "Probabilistic Treatment of General Combinational Networks," Computers, IEEE Transactions on, vol. C-24, no. 6, pp. 668–670, 1975.

[101] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, "Estimate of Signal Probability in Combinational Logic Networks," in European Test Conference, 1989., Proceedings of the 1st, 1989, pp. 132–138.

[102] S. N. Pagliarini, L. A. de B. Naviner, and J.-F. Naviner, "ESTIMATION DE LA FIABILITE D'UN CIRCUIT LOGIQUE," Patent FR 13 52 279, Mar., 2013.

[103] S. Pagliarini, L. de B. Naviner, and J.-F. Naviner, "SNaP: a Novel Hybrid Method for Circuit Reliability Assessment Under Multiple Faults," in European Symposium on the Reliability of Electron Devices, Failure Physics and Analysis, 2013.

[104] S. Pagliarini, A. B. Dhia, L. de B. Naviner, and J.-F. Naviner, "SNaP: a Novel Hybrid Method for Circuit Reliability Assessment Under Multiple Faults," Microelectronics Reliability, 2013.

[105] "Synplify Pro Data Sheet," Synopsys, Inc., Mountain View, California. [Online]. Available: http://www.synopsys.com/Tools/Implementation/FPGAImplementation/

[106] "ProASIC3E Flash Family FPGAs Datasheet," Microsemi Corporation, Aliso Viejo, California. [Online]. Available: http://www.actel.com/documents/PA3E_DS.pdf

[107] I. Polian and J. Hayes, "Selective Hardening: Toward Cost-Effective Error Tolerance," Design Test of Computers, IEEE, vol. 28, no. 3, pp. 54 –63, may-june 2011.

[108] C. Zoellin, H.-J. Wunderlich, I. Polian, and B. Becker, "Selective Hardening in Early Design Steps," in Test Symposium (ETS), 13th European, may 2008, pp. 185 –190.

[109] Synopsys Armenia Educational Department, "SAED 90nm Generic Library." [Online]. Available: http://www.synopsys.com/Community/UniversityProgram

[110] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," in Proceedings of the International Symposium on Circuits and Systems, Jun. 1985, pp. 663–698.

[111] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI Module Placement Based on Rectangle-packing by the Sequence-pair," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 15, no. 12, pp. 1518–1524, 1996.

[112] O. A. Amusan, A. F. Witulski, L. W. Massengill, B. L. Bhuva, P. R. Fleming, M. L. Alles, A. L. Sternberg, J. D. Black, and R. D. Schrimpf, "Charge Collection and Charge Sharing in a 130 nm CMOS Technology," Nuclear Science, IEEE Transactions on, vol. 53, no. 6, pp. 3253 – 3258, Dec. 2006.

[113] O. Amusan, M. Casey, B. Bhuva, D. McMorrow, M. Gadlage, J. Melinger, and L. Massengill, "Laser Verification of Charge Sharing in a 90 nm Bulk CMOS Process," Nuclear Science, IEEE Transactions on, vol. 56, no. 6, pp. 3065 –3070, Dec. 2009.

[114] N. M. Atkinson, "Single-event Characterization of a 90-nm bulk CMOS digital cell library," Master's thesis, Vanderbilt University, Nashville, Tennessee, 2010.

[115] T. Ban and L. Naviner, "Progressive Module Redundancy for Fault-tolerant Designs in Nanoelectronics," Microelectronics Reliability, vol. 51, no. 9-11, pp. 1489 – 1492, 2011.

[116] S. Pagliarini, G. dos Santos, L. de B. Naviner, and J.-F. Naviner, "Exploring the Feasibility of Selective Hardening for Combinational Logic," in European Symposium on the Reliability of Electron Devices, Failure Physics and Analysis, 2012.

[117] ——, "Exploring the Feasibility of Selective Hardening for Combinational Logic," Microelectronics Reliability, vol. 52, no. 9-10, pp. 1843 – 1847, 2012. [Online]. Available: http:/ / www.sciencedirect.com/ science/ article/ pii/ S0026271412002399

[118] S. Pagliarini, A. Ben Dhia, L. de B Naviner, and J.-F. Naviner, "Automatic Selective Hardening Against Soft Errors: A Cost-based and Regularity-aware Approach," in Electronics, Circuits and Systems (ICECS), 19th IEEE International Conference on, 2012, pp. 753–756.

[119] N. Atkinson, A. Witulski, W. Holman, J. Ahlbin, B. Bhuva, and L. Massengill, "Layout Technique for Single-Event Transient Mitigation via Pulse Quenching," Nuclear Science, IEEE Transactions on, vol. 58, no. 3, pp. 885 –890, june 2011.

[120] R. Harada, Y. Mitsuyama, M. Hashimoto, and T. Onoye, "Neutron Induced Single Event Multiple Transients With Voltage Scaling and Body Biasing," in Reliability Physics Symposium (IRPS), 2011 IEEE International, april 2011, pp. 3C.4.1 –3C.4.5.

[121] J. Black, A. Sternberg, M. Alles, A. Witulski, B. Bhuva, L. Massengill, J. Benedetto, M. Baze, J. Wert, and M. Hubert, "HBD Layout Isolation Techniques for Multiple Node Charge Collection Mitigation," Nuclear Science, IEEE Transactions on, vol. 52, no. 6, pp. 2536 – 2541, dec. 2005.

[122] L. Entrena, A. Lindoso, E. S. Millan, S. Pagliarini, F. Almeida, and F. Kastensmidt, "Constrained Placement Methodology for Reducing SER Under Single-Event-Induced Charge Sharing Effects," Nuclear Science, IEEE Transactions on, vol. 59, no. 4, pp. 811 –817, aug. 2012.

[123] ISIS. (2013, Jul.) A world centre for neutrons and muons. [Online]. Available: http:/ / www.isis.stfc.ac.uk/

[124] L. A. Tambara, F. L. Kastensmidt, M. S. Lubaszewski, T. R. Balen, P. Rech, and C. Frost, "Neutron-induced Single Event Upset in Mixed-Signal Flash Based

FPGA," in Radiation and Its Effects on Components and Systems (RADECS), 13th European Conference on, 2012.